Progress in IS

Egon Berghout · Rob Fijneman ·
Lennard Hendriks · Mona de Boer ·
Bert-Jan Butijn *Editors*

Advanced Digital Auditing

Theory and Practice of Auditing Complex Information Systems and Technologies

OPEN ACCESS

Springer

Progress in IS

"PROGRESS in IS" encompasses the various areas of Information Systems in theory and practice, presenting cutting-edge advances in the field. It is aimed especially at researchers, doctoral students, and advanced practitioners. The series features both research monographs that make substantial contributions to our state of knowledge and handbooks and other edited volumes, in which a team of experts is organized by one or more leading authorities to write individual chapters on various aspects of the topic. "PROGRESS in IS" is edited by a global team of leading IS experts. The editorial board expressly welcomes new members to this group. Individual volumes in this series are supported by a minimum of two members of the editorial board, and a code of conduct mandatory for all members of the board ensures the quality and cutting-edge nature of the titles published under this series.

Egon Berghout • Rob Fijneman •
Lennard Hendriks • Mona de Boer •
Bert-Jan Butijn
Editors

Advanced Digital Auditing

Theory and Practice of Auditing Complex Information Systems and Technologies

Springer

Editors
Egon Berghout
Rotterdam, The Netherlands

Rob Fijneman
TIAS School for Business and Society
Tilburg, The Netherlands

Lennard Hendriks
Ernst and Young
Maarssen, The Netherlands

Mona de Boer
PricewaterhouseCoopers
Amsterdam, The Netherlands

Bert-Jan Butijn
Erasmus University Rotterdam
Rotterdam, The Netherlands

This work was supported by Prof. Dr. J. R. ter Horst, Prof. Dr. H. R. Commandeur.

ISSN 2196-8705 ISSN 2196-8713 (electronic)
Progress in IS
ISBN 978-3-031-11088-7 ISBN 978-3-031-11089-4 (eBook)
https://doi.org/10.1007/978-3-031-11089-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Not withholding the long tradition of IT auditing expertise in almost all technological areas, we experience the intricacy of today's world including new technologies, such as machine-learning, blockchain technology, massive data processing, and artificial intelligence, as increasingly complex and problematic to control. Contemporary information systems more and more interact with external stakeholders and regularly make complex decisions, often with far-reaching consequences and without adjuvant human supervision. Furthermore, digital systems are increasingly a part of larger ecosystems making it more difficult to precisely identify ownership and responsibilities. Controlling advanced digital systems becomes increasingly challenging. Going back to a sort of black box approach does not seem to be the solution, transparency is asked for by all stakeholders and society at large.

This increasing system complexity affects the IT auditing discipline in several ways. First, the amount of work and the variety thereof is tremendously increasing. However, similar to the financial industry in the 80s, risks often originate from the interaction of parts of the ecosystems and will be difficult to identify. Auditors traditionally counter complexity by breaking up their work into manageable parts. However, society increasingly requires comprehensive audits, including accumulating effects of intrusions, system or cloud unavailability, and data leakages.

In this book, we explore auditing frameworks for advanced digital technologies. We intend to provide insights and methodologies that can be directly implemented in practice. The editors wish to thank the 16 writers, who took the time to share with us their thought-leading insights. We also wish to thank the Erasmus School of

Auditing & Assurance, TIAS School for Business and Society, and Springer Nature for making this book available as open source.

We sincerely hope that you enjoy reading this book.

Rotterdam, The Netherlands — Egon Berghout
Tilburg, The Netherlands — Rob Fijneman
Maarssen, The Netherlands — Lennard Hendriks
Amsterdam, The Netherlands — Mona de Boer
Rotterdam, The Netherlands — Bert-Jan Butijn

Contents

Editors and Contributors

About the Editors

Egon Berghout is leading the consultancy practice of the Information Management Institute, which supports senior management in the governance of digital endeavors. He is also a part-time academic director of the IT Auditing & Advisory program at Erasmus University and a part-time full professor of Information Systems at the University of Groningen. He is a co-founder and past-president of the Benelux AIS Chapter, an initiator of the European CIONET research paper of the year election, and a senior editor of Information System Management journal. Egon authored several academic books, including the popular GQM-approach for software development and quality management. He also serves as a digital and financial non-executive board member in several organizations.

Rob Fijneman Since 2004, Rob Fijneman is a professor of IT auditing at Tilburg University and the TIAS School for Business and Society. He leads the Executive Master of Science in IT auditing at TIAS. Rob is a seasoned IT audit professional working since 1986 in the international IT assurance and advisory practice of KPMG. He served multiple global companies both as an IT audit partner and as an advisor to the Board regarding Technology programs, IT Governance, and controls and compliance. Currently, he works for international audit clients out of Switzerland.

Lennard Hendriks is a Partner at EY and an advisor on IT risk management and internal control to a large number of multinationals with listings in among others the Netherlands, the United States, and Germany. Lennard manages (large) programs and performs Quality Assurance assignments as well as sundry IT Audit activities on a regular basis. As a lecturer, he is affiliated to the post graduate IT Auditing & Advisory program at the University of Rotterdam, and he is a regular speaker at the Erasmus School for Accounting & Assurance.

Mona de Boer is a Partner of Data & Technology at PwC Netherlands, where she leads the Responsible AI & Digital Ethics practice. Mona is a Chair of the Algorithm Assurance expert group of the Professional Association for IT Auditors in the Netherlands (NOREA) and affiliated with the University of Amsterdam as a lecturer and Ph.D. researcher.

Bert-Jan Butijn has conducted his Ph.D. research at the Jheronimus Academy of Data Science in the field of blockchain technology. He is currently employed as a Postdoctoral researcher at the Erasmus School of Accountancy and Assurance. IT has always been of great interest to him.

Contributors

Alexander Boer provides Trusted Analytics services at KPMG. He is an expert in Artificial Intelligence, law, and risk management, and holds a Ph.D. degree in Artificial Intelligence and Law from the University of Amsterdam. At that university, he worked as an Artificial Intelligence researcher and lecturer for two decades, applying Artificial Intelligence technologies to practical and theoretical problems in the field of law.

Léon de Beer is a senior manager at the Trusted Analytics team of KPMG The Netherlands. He has more than a decade of experience with questions around the reliability and security of data-intensive and data-driven processes, both as an auditor and advisor.

Rewin J. M. Doekhi studied at Leiden University and ESAA and is currently working as an IT-auditor for the KLM Royal Dutch Airlines. Prior to this, he worked as a SAP Finance & Controlling functional consultant on global roll outs for various multinationals. At KLM, he worked on a blockchain Proof of Concept and its integration with SAP. Because of his background in IT, he closely follows digital innovations such as Artificial Intelligence, augmented reality, and blockchains.

Jalal Bani Hashemi started his career in 2010 as a professional trainee at ABN AMRO Group Audit. Over the years as an IT Auditor, he performed various audits on IT infrastructure, platforms, and services. Currently, he is the IT Audit Manager responsible for the technical IT Audit coverage of IT infrastructure and platform services within ABN AMRO.

Tjitske Jager started her audit career at Deloitte after which she moved to a senior auditor position in Internal Audit at Achmea (Insurance). Her team focusses on examining of and advising on the quality of organizational structures, risk

management, and internal control. Internal Audit at Achmea not only identifies problems, it also provides constructive and improved solutions. She recently completed the IT & Advisory program at the Erasmus School of Accounting & Assurance to focus more on the IT component in her work area.

Mieke Jans is an Associate Professor in Business Informatics at Hasselt University and Maastricht University.

Manal Laghmouch is a Ph.D. candidate at Hasselt University (Belgium) within the research group Business Informatics. She obtained a Master's degree in Applied Economics: Business Informatics at Hasselt University. She teaches courses on Process Mining at Hasselt University and at the Erasmus School on Accounting & Assurance in Rotterdam (The Netherlands). Her research interests include Audit Analytics, Process Mining, and Business Process Management.

Pieter Oosterwijk is a researcher and auditor in the Netherlands Court of Audit. He has a background in statistics and data science and specializes in quantitative analysis for policy research.

Miranda Pirkovski is a director at the Netherlands Court of Audit. She is a certified accountant and IT auditor with 12 years of experience in operational and IT risk management & audit in the banking industry.

Jacques Putters started his career as a mainframe/MVS system programmer at KLM. He has been working at Group Audit ABN AMRO as a technical IT Auditor since 2004. He has extensive experience in auditing IT infrastructure, such as OpenVMS, Tandem, HP-Unix, AIX, Solaris, Linux, Windows and z/OS and subsystems such as IMS and DB2.

Frank van Praat is a Director at KPMG The Netherlands and is overall responsible for KPMG's Trusted Analytics service. Having a masters in digital human intelligence and a background in IT auditing, he specializes in how to achieve trust in (advanced) data analytics and complex algorithms. On top of his work for KPMG, Frank also teaches about the topic of AI Governance in various universities and on different masterclasses.

Eric Westhoek is a co-founder of the IT audit firm 3angles audit risk & compliance and affiliated with the Erasmus School of Accounting & Assurance (Postmaster IT Auditing & Advisory) and at Tilburg University as a teacher, counselor, and examiner. He is a seasoned consultant and IT auditor at the interface of organization and ICT and enjoys working with directors, regulators, and professionals to optimize the strategy of an organization and the role of the automated information systems to get and stay "in control."

Ayhan Yavuz started his career at ABN AMRO in 1995 as a management trainee. He subsequently worked in several positions within Group Audit, covering business lines, control functions, and IT. Since January 2021, he has been working for the Platforms & Technology Unit within Innovation & Technology as the Senior Manager Control Posture Transformation.

Berrie Zielman is a senior auditor in the Netherlands Court of Audit, and is interested in the application of data science and statistics in policy research.

Abbreviations

AI	Artificial Intelligence
BCT	Blockchain Technology
BPM	Business Process Management
BPMN	Business Process Model Notation
CNN	Convolutional Neural Network
CNN	Recurrent Neural Network
COBIT	Control Objectives for Information and related Technology
COSO	Committee of Sponsoring Organizations of the Treadway Commission
DApp	Decentralized Application
DLT	Distributed Ledger Technology
DPIA	Data Protection Impact Assessment
ELC	Entity Level Controls
ERP	Enterprise Resource Planning
GAAP	Generally Accepted Accounting Principles
GDPR	General Data Protection Regulation
IaaS	Infrastructure as a Service
ICFR	Internal Control over Financial Reporting
ICS	Intercompany settlement
IS	Information System
IT	Information Technology
ITGC	IT general controls
ITGC	Information Technology General Controls
ITIL	Information Technology Infrastructure Library
KYC	Know Your Customer
ML	Machine Learning
NER	Named Entity Recognition
NIST	National Institute of Standards and Technologies
NLP	Natural Language Processing

P2P	Peer-to-Peer
PaaS	Platform as a Service
PIA	Project Impact Assessment
PoC	Proof of Concept
POS	Part-of-Speech tagging
PoS	Proof-of-Stake
PoW	Proof-of-Work
SaaS	Software as a Service
SHA-256	Secure Hashing Algorithm 256
SLA	Service Level Agreement
SOx	Sarbanes–Oxley Act
SRL	Semantic Role Labeling
TLC	Transaction Level Controls
XAI	Explainable AI

Auditing Advanced Information Systems and Technologies in a Modern Digital World

Egon Berghout, Rob Fijneman, Lennard Hendriks, Mona de Boer, and Bert-Jan Butijn

1 Introduction

Complex technology has been around ever since the start of computers. Maxwell Newman's first programmable computer Colossus in 1943 cracking World War II cryptography was a truly complex system at that time (Haigh & Ceruzzi, 2021). In 1965 Gordon Moore posited that the number of transistors on microchips doubles every 2 years, implying that the technical developments underlying our increasingly complex systems continue to develop at an impressive pace (Valacich & Schneider, 2022). We may therefore expect that the complexity of information systems will also continue to increase for the years ahead.

Due to the continuous development of the underlying technology, information systems take over increasingly complex tasks. An example of a currently cutting-edge task concerns autonomously driving cars. The computing power required for

E. Berghout (✉)
Erasmus University Rotterdam, Rotterdam, The Netherlands
e-mail: berghout@ese.eur.nl

R. Fijneman
TIAS School for Business and Society, Tilburg, The Netherlands
e-mail: r.g.a.fijneman@tilburguniversity.edu

L. Hendriks
Ernst and Young, Maarssen, The Netherlands
e-mail: lennard.hendriks@nl.ey.com

M. de Boer
PricewaterhouseCoopers, Amsterdam, The Netherlands
e-mail: mona.de.boer@pwc.com

B.-J. Butijn
Erasmus University Rotterdam, Rotterdam, The Netherlands
e-mail: butijn@ese.eur.nl

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_1

image processing and interpretation is at the edge of today's capabilities. Autonomous-driving systems also affect our business and private lives and could possibly even run you over. Besides this increasing complexity of information systems themselves, there is also their accumulating interaction with people and other information systems.

The complexity of information systems also caused the emergence of the IT audit discipline in the late 1980s. IT auditors initially focused on the quality of financial reporting systems, however, also quickly deployed their knowledge in many other business domains. In this book we will explore the complexity of information systems and how we should develop the IT auditing discipline in order to control this complexity and maintain a trustworthy society.

2 Assurance Continuum

How do we know whether digital applications and solutions are sufficiently secure, are the answers generated by algorithms, for example, honest and fair, are we sufficiently resilient to cyberattacks and do we spend our money on the right digital solutions? These questions are extremely relevant for managers and supervisors of organizations as they must be able to account for their choices. Traditionally, the management report is a form of accountability for policy, which is fairly static in nature in the annual cycle. The Board report could explicitly discuss the digital agenda, and it has recently been explored whether an (external) IT audit "statement"[1] can also be added. Accountability for the quality of digital applications is taking on new dimensions now that developments are moving at lightning speed and everyone is linked to everyone. Certainties must be found on the digital highway.

These issues also play a role in our society. The protection of privacy is under considerable pressure, the numerous digital solutions build up a continuous personal profile. There are also painful examples of the use of algorithms in the public domain (Netherlands Court of Audit, 2021) that seriously harmed a number of citizens. According to the 2021 report on algorithms from the Court of Audit, responsible development of more complex automated applications requires thorough consideration and improved quality control. The social significance of aspects of digital integrity such as, honesty, fairness and security is increasing.

In the eighties of the last century, linked to the introduction of the Computer Crime Act (WCC I), an explicit link was created for the first time with accountability for automated data processing. Since 2019, the Computer Crime Act (WCC) III has been in effect, taking into account many developments in the field of cybersecurity and privacy. As the final element in the chain of control and accountability from WCC I, according to, for instance, the Dutch Civil Code 2, Article 393 paragraph 4, auditors must express their view on the reliability and continuity of the automated

[1] www.norea.nl

data processing insofar as it is relevant for the financial reporting. Many other countries have similar regulations and the European Union accepted the EU Cybersecurity Act and is developing a comprehensive European cybersecurity certification framework. More than 40 years later, we are dealing with complex legislation in the field of information systems and we use digital solutions that affect our administrative processes, but also almost all primary business functions. Consequently, the associated business and societal risk accumulate enormously.

Summarizing, there is an increasing need for quality control along the many new developments. How should accountability be organized, what role do directors and supervisory boards play in this, and how can IT auditing add value and balance risk? As indicated, these questions play a role not only at the individual organizational level, but also at the societal level. For example, how can the government restore or regain the trust of their citizens by explicitly accounting for the use of its digital solutions?

3 Technology Developments

Contemporary businesses maintain a complex mix of technology solutions, partly older (legacy) systems and new online (often front office) oriented solutions. Ensuring integrity of data, ensuring continuity, and being able to make the right investments versus costs for maintenance of older solutions remains challenging for almost all organizations. The following trends seem pertinent (WilroffReitsma, 2021; KPMG, 2020):

- *Flexible working* is becoming the norm. Over the past year, the cloud workplace has grown in popularity—more than predicted. Initially, employees had to start working from home because of COVID. However, soon appreciated this flexible way of working. The cloud-based workplace emerged quickly.
- *Distributed cloud* offers new opportunities. Cloud technology seems more economical for most organizations. Distributed cloud solutions may speed up data transfer, resolve compliance issues and further reduce cost. Storing data within specific geographic boundaries (often required by law or due to compliance) is an important reason for choosing the distributed cloud, where cloud solutions are offered in the proximity of the client.
- Business use of *Artificial Intelligence* (AI) is increasing, for example, chatbots and navigation apps. This technology will become increasingly prominent in businesses, also because computing power and software are becoming cheaper and more widely available. For example, AI will increasingly be used to analyse patterns from all kinds of data.
- *Internet of behaviours*. Today, large amounts of data are generated by many business processes and providing new insights, which plays an increasingly important role in strategic decision-making. Data-driven methods will increasingly be used to change human behaviour. Based on data analyses, suggestions or

autonomous actions can be developed that contribute to issues such as human safety and health. An example is the smartwatch that monitors blood pressure and oxygen levels and provides health tips based on that.
- Maturity of *5G mobile internet* in practice. In many European countries 5G mobile internet is now operational allowing many new applications, especially in the field of the Internet of Things and also autonomous vehicles.

Advancing technology and advancing software engineering practices, together with the increasing installed base of systems, allow the realization of increasingly complex systems. Indicators of system complexity are:

- The number of data entities in the system.
- The number of relations between the above data entities (the relationship as separate entity).
- The diversity of entities (and of relationships).
- The velocity with which entities are added to the system.
- The agility with which those systems can be adapted to new requirements.
- The context of the system, including the number of stakeholders and the systems' impact on these stakeholders.

Unfortunately, the only way to make complex information systems controllable is to add functionality and, therefore, making these systems again more difficult. IT auditors traditionally assessed the quality of financial information systems in order to protect external stakeholders from incorrect financial information; however, they will increasingly also assess non-financial information systems that particularly impact many stakeholders and probably also through a financial risk and return perspective. In analogy with the car industry, the liability of manufacturers remains important; however, this will not be sufficient to control the quality of high impact complex information systems. Comparable to the road readiness certification of cars, third party assurance granted by independent IT auditors remains an important tool to control the quality of complex digital systems.

4 Management Responsibilities

Managing and supervising digital solutions remains an extremely complex and often less desirable management topic. The complexity of technology is a deterrent, the mixture of legacy systems and new digital solutions reduces its transparency. Many stakeholders manage part of the technology chain and the quality requirements are accordingly complex. The introduction of new technology often includes major organizational changes and these changes subsequently introduce "winners" and "losers" in the new situations. Both of these groups will often unite and introduce additional complex political processes on top of the technological complexity. Furthermore, digital innovations always partly depend on external systems and consultants. These external stakeholders again accrue proprietary interests in the

complex transition of the organization. This makes digital innovations often extremely difficult to manage, especially in more traditional organizations with vested interests.

Digital innovations, therefore, require extensive management attention, particularly from senior management and supervisory boards. Both value creation and risk management can be aligned with the COSO framework (Everson et al., 2017). Subsequently, one may opt for (parts of) the international COBIT (Control Objectives for Information Technology) framework (ISACA, 2018). In doing so, management makes explicit which management standards are applicable in and around the digital solutions and can determine their design as well as their operational functioning.

In view of all digital changes, new knowledge of the emerging technologies is constantly required. Organizing this in conjunction with an eye for the quality of the solutions and sometimes also the inherent limitations is what makes governance work well. Governance processes and structures require continuous evaluation and adjustments.

The suppliers of the digital solutions also play an important role. They provide increasingly better and safer, often cloud-based, solutions. Some suppliers tend to primarily focus on functional innovation than on cyber security and control. Buyers of digital solutions also insufficiently ask providers to develop "secure by design" systems. While designing the system, sufficient controls can, and should, be built in.

Is the tide turning, in other words are the new digital solutions becoming so complex that no one can determine the correctness of the content? It is not possible to opt for such a "black box" approach from the perspective of management responsibility. Management always remains responsible to balance risk and controls, and this book provides ample frameworks and techniques to do so.

5 Outline of This Book

This book encompasses a total of nine chapters. In the chapter hereafter (chapter "Auditing Complexity"), we will discuss the fundamentals and principles of auditing. Another topic the chapter touches upon is the effects of increasing technological complexity on the IT auditing discipline. The third chapter provides an introduction to several complex information systems like blockchain technology, artificial intelligence and cloud computing. This chapter provides the background for the chapters following thereafter that each present a framework to audit a complex technology. Chapter "The Intercompany Settlement Blockchain: Benefits, Risks, and Internal IT Controls" presents a framework to audit blockchain technology. The framework is based on a case study of a blockchain system implemented at the Royal Dutch KLM. An extensive description of the case, and an analysis of risks and controls of the blockchain system is presented. Following, in chapter "Understanding Algorithms" an extensive analysis of three case studies of algorithms is discussed that are used in practice by Dutch ministries. The analysis has resulted

in a framework to audit algorithms in general, supplemented with considerations for algorithms that employ artificial intelligence. Building on the framework in chapter "Understanding Algorithms", chapter "Keeping Control on Deep Learning Image Recognition Algorithms" presents a framework specifically for image recognition. The framework is developed for the specific case wherein a large insurer has developed an algorithm to recognize damage to greenhouses. The seventh chapter introduces the concept of algorithm assurance, to give some background on the relevance and importance of algorithm assurance, and to prepare the auditor for the basic skills needed to organize and execute an algorithm audit. Evermore organizations are working in the cloud increasing the need for best practices and guidance on how to audit cloud-based services. Chapter "Demystifying Public Cloud Auditing for IT Auditors" discusses these best practices and provides guidance to IT auditors. Fortunately, advanced techniques are now also available to IT auditors to aid them in their work. One of these techniques is processes mining employed to lay bare processes within an IS. Chapter "Process Mining for Detailed Process Analysis" provides an elaborate background on process mining along with several examples of how to use it.

References

Everson, M. E. A., Chesley, D. L., Martens, F. J., Bagin, M., Katz, H., Sylvius, K. T., Perraglia, S. J., Zelnik, K. C., & Grimshaw, M. (2017). *Enterprise risk management: Integrating with strategy and performance*. Committee of Sponsoring Organizations of the Treadway Commission. Retrieved from https://www.coso.org/pages/erm-framework-purchase.aspx

Haigh, T., & Ceruzzi, P. E. (2021). *A new history of modern computing*. MIT Press.

ISACA. (2018). *COBIT 2019 framework: Governance and management objectives*. ISACA.

KPMG. (2020). *Harvey nash/KPMG CIO survey 2020: Everything changed. Or did it?* Retrieved from https://home.kpmg/xx/en/home/insights/2020/09/harvey-nash-kpmg-cio-survey-2020-everything-changed-or-did-it.html

Netherlands Court of Audit. (2021, January). *Understanding algorithms*. Retrieved from https://english.rekenkamer.nl/publications/reports/2021/01/26/understanding-algorithms#:~:text=The%20Court%20of%20Audit%20found,use%20and%20operation%20of%20algorithms

Valacich, J., & Schneider, C. (2022). *Information system today managing the digital world* (5th ed.). Prentice Hall.

WilroffReitsma. (2021, December 21). *ICT trends 2021: Dit zijn de 10 belangrijkste*. Retrieved March 25, 2022, from https://wilroffreitsma.nl/nieuws/ict-trends-2021/

Auditing Complexity

Egon Berghout and Rob Fijneman

1 Introduction

Modern technology introduces new questions. For instance, how do we know that answers generated by algorithms are fair, and whether complex systems are sufficiently resilient to cyberattacks? These questions are extremely relevant for managers and supervisors of organizations as they must be able to account for their choices. Boards of Managements and other stakeholders have various alternatives to have these questions answered, including asking IT auditors. IT auditors provide consultancy; however, they will often be invited to provide assurance regarding high risk and high impact related issues. Assurance concerns trusted advice. However, auditing differs from consultancy and is primarily focused on investigating whether generally accepted IT auditing standards apply to the auditing object, where consultancy seems more focused on making recommendations. Consultancy could also be primarily based on prior experience in other engagements. Furthermore, IT auditors will also include societal relevance in their assessments and include consequences for other stakeholders.

IT auditing concerns the independent assessment of the quality of information technology, being, infrastructure, applications, processes, data, and governance. Quality includes many characteristics and is not only about integrity, availability, and security, but also includes fairness. In addition, the effectiveness and efficiency may also be assessed. This makes IT auditing an important instrument to identify and

E. Berghout (✉)
Erasmus University Rotterdam, Rotterdam, The Netherlands
e-mail: berghout@ese.eur.nl

R. Fijneman
TIAS School for Business and Society, Tilburg, The Netherlands
e-mail: r.g.a.fijneman@tilburguniversity.edu

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_2

control IT-related risks, when developing and applying digital solutions. IT auditing concerns the following entities:

1. Object of auditing—investigating whether the object is suitable for investigation.
2. Auditing criteria and methodology—investigating which criteria are suitable for this particular object and which methodology should be applied.
3. Client—investigating whether the person granting the audit is authorized to do so.
4. Auditor—investigating whether the persons performing the audit are capable to do so.

Auditing standards are controlled by standardization bodies, being the Auditing Standards Board (ASB), the International Auditing and Assurance Standards Board (IAASB), which is supervised by the International Federation of Accountants (IFAC) and the US-based Public Company Accounting Oversight Board (PCAOB). The above four entities are further discussed in the following sections.

2 Object of Auditing

IT auditing targets information systems, which typically include components, such as hardware, communications, software, data, procedures, and Staff. Together, these components fulfill certain functionality and this functionality is able to fulfill increasingly complex tasks. Due to the continuous developing technology, we may also expect increasing use and complexity for the decade ahead. Audits therefore require careful scoping of the auditing object because inclusion of all components of all information systems will commonly be uneconomical. Risk/impact assessments form the basis of this scoping. What is the likelihood of misstatements and their associated impact? Subsequently, which information systems require investigation and which ones are out of scope?

Auditors apply the following audit risk model:

$$AR = IR \times CR \times DR$$

In the above formula, (AR) represents the acceptable audit risk. (IR) represents the inherent risks associated to the audit object (business and/or technology). (CR) represents the control risk and whether internal procedures should be capable to detect misstatements. For instance, the (CR) in a blockchain should be close to zero. (DR) represents the risk that errors, which are not prevented by internal procedures, are also not detected by the auditor.

Audits require a comprehensive overview of the information function of the organization and associated risk/impact assessment. The information function includes (Romney & Steinbart, 2018):

1. The information systems and associated data (IS).
2. The information technology supporting the information systems (IT).

3. The organization supporting both information systems and information technology (IM).

In modern organizations, there is an abundant number of auditing objects for IT auditors and these audits are definitely not restricted to systems supporting financial reporting. Auditability also requires viable and working control measures. Even the most basic information systems become error prone in notoriously unprofessional environments. Along with the increasing complexity of information systems more and more control measures are required.

3 Auditing Criteria and Methodology

As mentioned before, in modern organizations there is an abundant number of auditing objects and audits are definitely not restricted to financial reporting related systems. The applied audit methodology should address all relevant system quality characteristics. Common risk categories include:

1. Confidentiality—unauthorized disclosure is prevented and external regulatory requirements are met.
2. Processing integrity—data are processed flawlessly, completely, and timely, and only with proper identifications and authorizations.
3. Availability—ensuring that legitimate users are offered continuous access, also in case of contingencies.

In practice, IT auditors may apply an extensive set of methods to assess their object of study (if appropriate), such as ISA3000 for financial controls, ISO2700x for operational system controls, COBIT for IT governance controls, and PRINCE2 for system development controls.

The emergence of self-learning systems challenges the existing auditing methods. As long as the algorithm training phase can be discerned from the operational phase, one should be able to scope auditable components of both the self-learning and the operational system. The additional complexity is primarily caused by the unpredictability of the environment of the system and, therefore, the interaction of the self-learning and operational system and the environment. For instance, the self-driving car systems would probably be auditable if its driving would be restricted to predetermined isolated roads, however, not in any traffic situation and in any weather condition. In such a case, the car requires automated controls that hopefully prevent the car from haphazard behavior, or stop the car in certain conditions. Such automated control systems could again be subject to auditing. Auditing always concerns historic data.

In continuous auditing, technology is used to continuously monitor exemptions and inconsistencies. Subsequently, the recovery of these exemptions is included in the overall systems, comparable to fault tolerant systems.

Audit methodology also includes the audit process, which includes the systematic engagement of clients, their possible acceptance, confirmation through the engagement letter, and establishment of an audit plan. Such an IT audit plan includes the basic understanding of the organization and its information function, risk assessment, defining the control objectives, test plan (building/execution), and evaluation of findings. Subsequently, the audit findings are reported to the client.

Increasing complexity of systems often requires new audit methodology, for instance, criteria for self-learning systems. Common risk categories, being confidentiality, integrity, and availability remain relevant; however, the impact of a particular category might change. For instance, confidentiality in blockchains differs from traditional trading platforms.

4 Clients

Being audited requires genuine commitment of organizations, because they should disclose all necessary information regarding the object of study, facilitate the consultation of employees, and/or site visits. Such commitment is restricted to those who represent the ownership of the organization. Normally, this concerns the board of management, audit committee, or board of supervisors. Furthermore, IT auditors should also include societal relevance and ethical considerations in their considerations and avoid audits that may be harmful to society.

Modern technology introduces additional problems regarding the identification of the client. For instance, who has the role of the client in a blockchain with distributed ownership? Some AI-based algorithms may not be equally beneficial to all relevant stakeholders and there is scarce information about degrees of societal acceptance or acceptable categories of inequality.

5 Auditors

From auditors we expect that they work on basis of a common body of knowledge and conclusions should be indifferent for the person performing the audit. This requires state of art technical, legislation, and organizational knowledge; and also critical reflection of proprietary expertise. Auditors should be objective, integer, competent, and confidential.

Given the complexity of information systems and organizations, being an auditor requires a profound basic education accompanied with lifelong permanent education. This basic education should in our opinion include a relevant master of science degree because the scientific approach to learning and accumulation of knowledge remains prerequisite in such a complex and dynamic domain. Graduates should also be able to further develop the existing IT auditing body of knowledge. Currently, the entry level education differs per country; however, we expect these requirement to

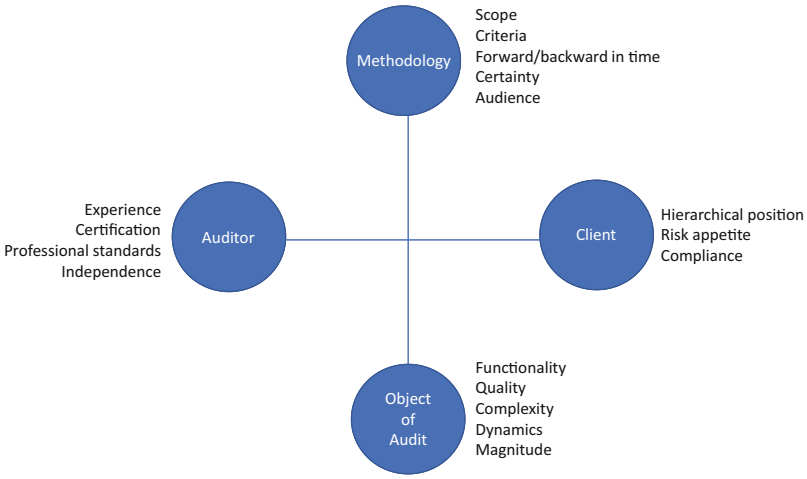


Fig. 1 Primary IT auditing entities

homogenize. Similar to other critical professions, such as the medical profession or architecture, IT auditors need to accomplish a certain supervised working period. Advanced work areas typically require continuous training. To organize professional standards and also control the adherence of auditors to these standards, governments should facilitate IT auditing communities. In the Netherlands, for instance, IT auditors require an accredited parttime IT auditing university degree and minimally 3 years of relevant practice. Becoming an auditor could also be considered an audit itself and should, therefore, be transparent and controllable. This includes a professional association with mandatory ethical and quality control standards and the possibility to dispute professional issues.

Advanced technologies require comprehensive advanced technology knowledge of the IT auditor. In order to obtain this knowledge, the IT auditor should enroll in relevant courses or partner with more experienced auditors.

The four key-entities of IT auditing, which have been described in this chapter, and their key-characteristics have been illustrated in Fig. 1.

6 Conclusions

IT auditors provide assurance regarding the quality of information systems and technology allows these information systems to become increasingly complex. These complex Information systems also increasingly independently interact with their environment. Examples are complex web services, self-driving cars, and unsupervised digital currencies, such as Bitcoin. Managing the quality of these complex information systems requires additional incorporated control measures. Bitcoin being a meritorious example of a system that includes advanced operational

controls for trading integrity. As such, one could say that many system developers are increasingly performing some IT auditing tasks and contemporary information systems often include more functionality to control their quality than they encompass core functionality. Additionally, there is also a growing need for a truly independent IT auditor that balances risk and control measurer and provides assurance regarding their adequacy.

Reference

Romney, M. B., & Steinbart, P. J. (2018). *Accounting information systems* (14th ed.). Pearson.

Introduction to Advanced Information Technology

Bert-Jan Butijn

1 Introduction

Over the years Information Systems (IS) have become increasingly complex and are difficult to grapple. The complexity of recent novel technologies like blockchain (BCT), artificial intelligence (AI) and cloud computing constitutes a genuine challenge to IT-auditors tasked with auditing these IS to provide assurance. Recognizing this challenge this book aims to aid IT-auditors in their audit of such complex IS. This book provides novel insights into these complex IS by demonstrating how control frameworks can be applied to these technologies using several real-life case studies. The chapters that follow hereafter each discuss a different technology.

Each of the aforementioned IS complex, and therefore particularities of the technologies discussed in this book may not be well understood. This chapter discusses the inner-workings, intricacies, and concepts related to these technologies to provide the background necessary to perform an audit using the frameworks presented in the chapters hereafter. In Sect. 2 background is provided about blockchain technology. Section 3 expounds on artificial intelligence, more specifically how it can be perceived and how it is practically used. Similar to the outline of this book, the final technology discussed in the chapter in Sect. 4 is cloud computing. It is strongly recommended to read this chapter before continuing to read the other chapters.

B.-J. Butijn (✉)
Erasmus University Rotterdam, Rotterdam, The Netherlands
e-mail: butijn@ese.eur.nl

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_3

2 Blockchain Technology

The concept of blockchain technology was first published in an anonymous paper by an author called Satoshi Nakamoto in 2008.[1] Blockchain technology incorporates several technologies previously developed for initiatives like Adam Back's Hash Cash (Back, 2002), Digi Cash proposed by David Chaum (1979), and Bit Gold created by Nick Szabo (2005).[2] In 2009, the Bitcoin network was established when the first (genesis) block was mined by Satoshi Nakamoto. Although Bitcoin is often mentioned in the same breath as BCT, there is an important distinction: BCT is the technology that underpins the Bitcoin making it possible to perform transactions without a trusted third party. Bitcoin on the other hand is a cryptocurrency that represents value similar to normal currency that is made possible by the technology.

Since the initial conception of BCT it has gained immense worldwide attention from organizations. BCT has many favorable characteristics and currently many prominent firms like JP Morgan Chase, Maersk, and KLM have started to explore how they can leverage the potential of the technology to their advantage. One of the key features of BCT is that it allows for transactions between parties without requiring a trusted intermediary (e.g., a bank) to safeguard the safety of their transaction. This remarkable feature is made possible by a sophisticated combination of technologies.

BCT is a specific form of distributed ledger technology where the ledger is deployed on a Peer-to-Peer network (P2P). On the P2P network all data about transactions is replicated, shared, and synchronously distributed across multiple peers. Transactions are processed following a strict consensus protocol that is operated by specific nodes to ensure the validity of the transactions requested by other peers in the network, and to synchronize all shared copies of the distributed ledger. During the execution of the consensus protocol, the data of valid transactions, along with other required metadata concerning the network, and the hash of the previous block are bundled by these specific nodes into a block using hashing functions. The essential and key property reflecting BCT architectures is that each block contains the hash of their predecessor, therefore linking all prior transactions to newly appended transactions; the blocks therefore form a chain with the aim of establishing a tamper-proof historical record. This property is depicted in Fig. 1.

As can be noted, BCT is a complex technology that itself encompasses a combination of several other technologies. Let us now further explore how these interrelated technologies interact with one another and constitute to a blockchain system. To exemplify how the technology works we will further discuss the initial BCT underpinning the Bitcoin from the perspective of the trustless transactions it enables. In Sect. 2.1, we first discuss the basic notions of blockchain technology. Then, smart contracts an important concept related to blockchain technology is

[1] Until date nothing is known about the identity or whereabouts of the original author(s).

[2] For further reading about the origins of blockchain technology, we recommend "On the Origins and Variations of Blockchain Technologies" by Sherman et al. (2019).

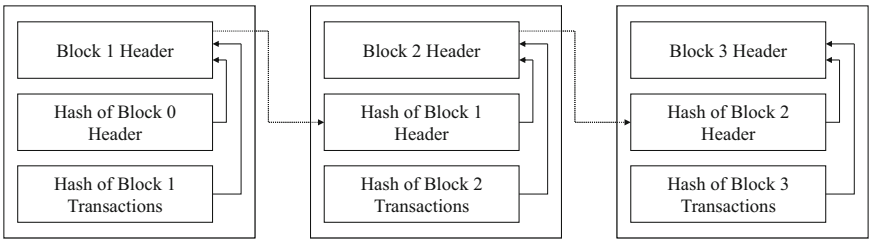


Fig. 1 Graphical depiction of blocks in a blockchain. Note how the combination of the previous block hash and the hash of current transactions form the blockheader

discussed in Sect. 2.2. The last section, Sect. 2.3 presents an overview of a typical blockchain architecture that explains the relation between some over the overarching concepts.

2.1 Basic Notions of Blockchain Technology

Owners of a Bitcoin can commit a transaction to the network by digitally signing a hash of the previous transaction and combining it with the public key of the requested recipient. Within a blockchain network public keys are used as the addresses of agents that make use the blockchains' services. The combination of the hash of the previous transaction and public key of the recipient are added to the end of the coin. Therefore, crypto coins can be considered as a chain of digital signatures. This chain of signatures allows anyone to audit and verify the transaction history of a coin. Albeit that the chain of signatures allows anyone to verify ownership claims, this technique does not prevent current owners to double spend a coin. *Double-spending* refers to the act of spending the same coin twice in two different transactions yet at the same time.

One of the unique features of BCT is that it prevents double-spending by introducing a distributed ledger that is shared among peers. Traditional transaction processors like a bank maintain a centrally kept ledger that records all transactions made and especially when they were made. This centralized ledger allows the transaction processor to verify whether transactions have already taken place.

BCT achieves these objectives in a different manner: (1) Transactions are publicly announced to all peers that are part of the P2P network that thereafter record them on their own copy of the distributed ledger. These peers are oftentimes referred to as nodes in blockchain nomenclature. It is important to note that nodes are physical or virtual machines connected to other nodes via a P2P network. Nodes can have one or many human owners, and someone can own several nodes. (2) Because there is no centralized ledger the nodes in the network need to reach a consensus about the history of the transactions on the ledger, and more specifically how to correctly chronologically order them. In principle this approach effectively

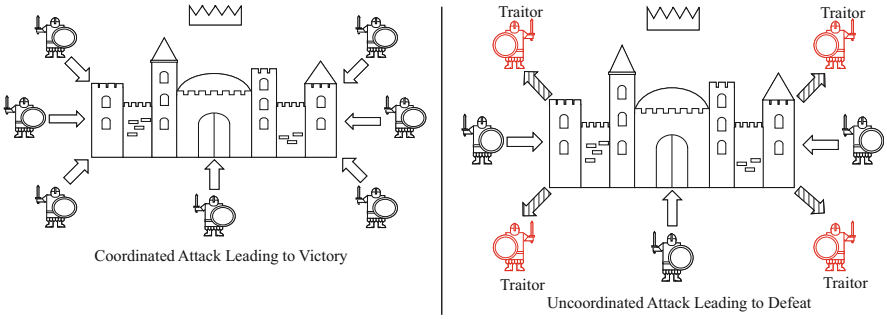


Fig. 2 The Byzantine Generals Problem. When all nodes behave honest and work together the system works otherwise it will fail

prevents double-spending when all nodes behave honestly. However, not all nodes can be trusted as some might be used to act maliciously and propose incorrect versions of the distributed ledger for their own gain. For instance, by introducing non-valid transactions to increase their own balance. Literature on distributed systems refers to this issue as the *Byzantine Generals Problem* (Lamport et al., 2019). Figure 2 illustrates this problem.

The illustration should be regarded as a metaphor for how distributed systems work: Imagine that there are several generals that have laid siege to Byzantium. The generals must collectively decide when to attack the city. Only when all generals launch their attack simultaneously, they can capture the city. However, if they do not the attack fails. Unfortunately, the generals cannot safely communicate with each other because all messages they will send might be intercepted or deceptively sent by the defenders of Byzantium. This raises the question how the generals can successfully organize their attack simultaneously?

When applying this analogy to blockchain Byzantium is the distributed ledger, and the generals are the nodes within the P2P blockchain network. Similar to the generals in the Byzantines Generals Problem, some nodes will try to manipulate the ledger and thus dismantle its integrity. Honest nodes need a method that enables them to identify transactions on the ledger that are fraudulent or incorrect to keep the distributed ledger free from errors.

To overcome this problem, several safeguards are presented in the original Bitcoin paper (Nakamoto, 2008). One of these safeguards is that transactions are processed in batches by several nodes[3] and are then stored in data structures called *blocks*. Note that each block can only contain a specific amount of data called the *blocksize*, meaning that a limited number of transactions can be included in the block. To create a block, the nodes proceed in the following manner: First, a node checks the validity of a requested transaction. Then, the node uses a timestamp server to timestamp a batch of transactions. Thereafter the node uses the Secure

[3] On some blockchain platforms like Ethereum, the number of nodes that process the transactions can amount up to 10,000.

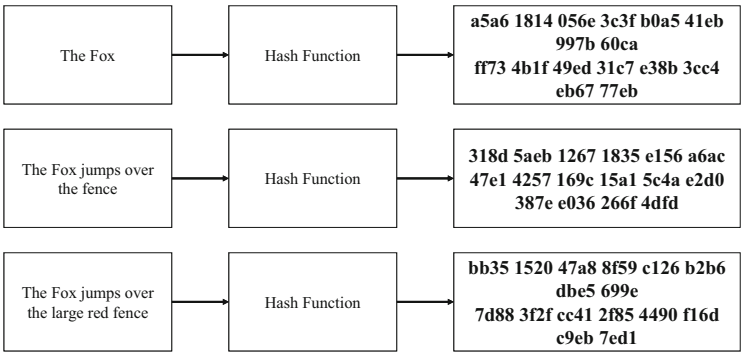


Fig. 3 Example of three texts translated into three unique hash digests. Note how although the length of the text differs the length of the hash is always 64 symbols

Hashing Algorithm 256 (SHA-256) to create a hash of each individual transaction. When given the same input, the SHA-256 algorithm will always return the same output as hash better known as a *digest*. Any small change to the original input however, will render a completely different digest. Figure 3 shows the differences in hashes with two different inputs.

Using the hash, it can effectively be proven that data existed at a certain point in time. More important, any tampering with the hash of a transaction would immediately be recognized as the corrupted hash would not be identical to the one of a correct transaction. Storing the individual hashes of each transaction would require vast amounts of storage space to store the data. Therefore, as a second step nodes bundle the batch of hashes using a *Merkle tree*. An example of a Merkle tree is shown in Fig. 4. In effect this means that the hash of each transaction re-hashed with that of other transactions until only one hash remains.

Another safeguard is proposed in the paper to further guarantee the historical integrity of the distributed ledger. In the hash of a novel block, the hash of the previous block is also included. Effectively this means that the blocks are chained together, and the more blocks are appended to this chain the more difficult it becomes to tamper with the ledger. This solution safeguards the ledger against tampering with the chronology of the transactions by malicious nodes. However, incorrect novel transactions could still be introduced. BCT remedies this problem by demanding that the nodes in the network verify whether (a) any of the newly announced transactions are legit and (b) what the correct version of the distributed ledger is. These activities are integral part of a consensus protocol with the aim of ensuring that the nodes in the P2P network reach a consensus on these aspects. A simple way of reaching consensus would be to allow all nodes to vote. Unfortunately, this would enable malicious nodes to launch a *Sybil attack* by creating an infinite number of duplicates of itself to gather more votes and control the P2P network.

The BCT underpinning the Bitcoin decreases the chance of a sybil attack by employing a Proof-of-Work (PoW) consensus protocol that nodes follow to verify

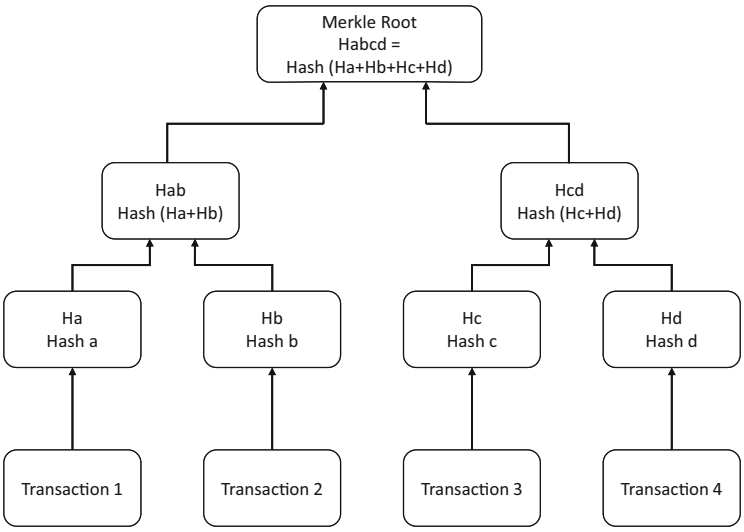


Fig. 4 Graphical depiction of a Merkle tree

transactions. This PoW entails that nodes use their computational power to "vote" on the validity of transactions instead of IP addresses, effectively meaning that the majority of computational power within the network decides. Although it might be easy for someone of ill-intend to amass several IP addresses, obtaining large amount of computational power is likely to be more difficult. Nodes deliver their PoW by solving a computational difficult mathematical puzzle. The first node to solve the puzzle is granted some Bitcoin as a reward. Finding the solution to the puzzle requires finding the right nonce (a random number) that matches the header of the current block, given information of the prior block. The process of finding the right solution to build a block is called *mining*, and nodes that make the effort to solve the puzzle are referred to as *miners*. There is only one miner that can be the first to mine a block. Whenever a node has found the right solution, it propagates the block it constructed to the other nodes. The other nodes then verify the correctness of the block, and if correct append it to their copy of the ledger.

Due to slow propagation of the block among nodes situations might arise where two different miners propagate a block concurrently as they are not aware of the existence of another new block. From that moment on it remains unclear for other miners which of the new blocks is the correct one. In such instances a *fork* in the chain of blocks is created. Figure 5 depicts what a fork looks like from a schematic perspective.

Whenever a fork occurs as a rule, nodes should always trust the longest chain as it represents the branch on which the most computational power has been spend. Nodes that did not propagate the novel block will have to wait until one of the chains becomes longer than the other. Forks are resolved by nodes choosing to adopt the longest chain over the other chain. It is only when the fork is resolved that the

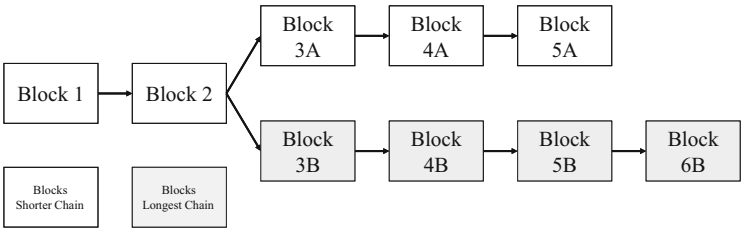


Fig. 5 Graphical example of the longest chain rule. Eventually all nodes will accept the bottom branch as it is the longest of the two

transactions in the new blocks that are part of the longest chain are confirmed. Besides resolving accidental forks, the longest chain rule also protects the integrity of the ledger from malicious users.

2.2 Smart Contracts

Initial versions of BCT only allowed their users to make transactions without a trusted intermediary. The desire and potential to employ the technology for uses other than cryptocurrency led to the creation of the Ethereum platform in 2015 by Vitalik Buterin (Buterin et al., 2016). Besides allowing users of the platform to request transactions using the native cryptocurrency called Ether, the Ethereum platform also supports the storage and execution of smart contracts. Smart contracts are computer programs that are stored on the blockchain and contain transaction logic in the form of code. The interesting prospect that this ability offers is that user can stipulate the conditions that have to hold before the transaction is executed (Zheng et al., 2020). Because a smart contract has its own balance and account, they can even hold funds in escrow until these conditions are met. Users can communicate with the smart contract and prompt it to execute some logic. Because these transactions that prompt the smart contract are also stored on the blockchain, a record is created who prompted the smart contract to perform the transaction. If the logic executed by the smart contract involves performing a transaction, this transaction is also recorded (Zheng et al., 2020). The execution of the smart contract and the transactions potentially resulting from this execution are performed by a large number of nodes in the blockchain network. It is therefore important that the execution of the smart contract code always yields the same output when executed by different nodes. If this were not to be the case, the nodes would never be able to reach a consensus on the validity of the transactions resulting from the execution. On public blockchains like Ethereum, a fee is paid for the execution of a smart contract to diminish the chance of abuse and to reward the executing nodes for their efforts (Xu et al., 2017).

Smart contract as a term has been coined by Nick Szabo already in 1994 (Szabo, 1997). However, the concept gained little traction in practice because there was no

suitable platform to store the smart contracts or to process transactions resulting from execution of the contract itself. With the rise of blockchain an infrastructure has been provided capable of storing and executing smart contracts while also enabling the processing of transactions resulting from the execution of the smart contract. Because smart contracts are deployed and stored on a blockchain, they inherit some important characteristics from the technology:

- Automatic execution: Smart contracts are in essence coded programs stored on a blockchain. By stipulating conditions with code users control under which circumstances a transaction is executed. It is because of this feat that smart contracts enable the automatic execution of transactions.
- Immutable: Once a smart contract is stored on the blockchain, it cannot be changed. Equally important, a deployed smart contract cannot be removed unless specifically instructed to do so.
- Tamper proof: Because a smart contract is immutable once deployed, no one can tamper with the code in order to influence the outcomes of a transaction process. Because transactions resulting from the execution of the smart contract are verified and performed via the blockchain, these are also tamper proof.
- Self-enforcing: All smart contracts have their own balance. Data concerning this balance is stored on the blockchain. This enables smart contracts to hold funds in *escrow* on their own balance until the predefined conditions are met.

The importance of smart contracts for the further development of BCT cannot be understated. By allowing users to stipulate their own transaction logic, the technology can be used for several applications that go well beyond cryptocurrency transactions. Collectively these applications are referred to as Decentralized Applications or DApps for short. Whereas traditional applications are connected to a database to retrieve information, smart contracts and by extension DApps, are connected to a blockchain from which they can obtain information. As can be noted, a blockchain therefore provides the infrastructure for a smart contract. The addition of smart contracts to the blockchain technology stack has significantly influenced the architecture of blockchain platforms. We will now further dive into the architectures of several blockchain architectures.

2.3 An Overview of Blockchain Architectures

Since the advent of Bitcoin, other blockchain platforms have been established like Ethereum that offer services other than cryptocurrency transactions. As a result, nowadays there are several types of blockchain platforms that can be discerned based on two main characteristics: how access to the network is arranged and whom has what permissions. Table 1 depicts the network arrangements.

Public blockchain platforms like Bitcoin and Ethereum allow for anyone to join the network as a miner or a client. Because anyone is allowed to join the network and subsequently verify and request any transactions, these platforms are also considered

Table 1 Spectrum of blockchain network arrangements

		Accessibility	
		Private	Public
Authorization	Permissioned	Participants in the network need to request access to an administrator to join the network. Each participant is assigned a unique set of rights linked to their digital identity.	In a public permissioned network, anyone can join the network. However, the rights on the network are restricted per participant. For instance, anyone can join the network, but not everyone can read or verify transactions.
	Permissionless	Private and permissionless allow only a group of network participants that have been admitted to the network to perform all actions possible on the network. Participants do have an identity but not a unique set of rights.	Public permissionless networks have as a characteristic that anyone can join the network. When a participant joins the network, they are allowed to read, write, and verify transactions. All data about the transactions (e.g., the blocks) is shared among all willing participants.

to be *permissionless*. It is important to note that the public and permissionless nature of a public blockchain is usually encapsulated in the algorithms that the platform uses to process data among things. Such features are therefore not easily changed.

Unfortunately, the fact that anyone can join the network and perform all possible actions might be considered as inconvenient by some organizations as their control over the platform is diminished. Moreover, public blockchains require complete transparency of the transactions history which is sometimes at odds with the privacy concerns of an organization. Combined, these two factors have led to the introduction of permissioned and *private* and *consortium* blockchains. Proponents of such blockchains advocate that more privacy and access control is needed to guarantee that the blockchain can be used for business. Rather than having one network for all participants, and being owned by all participants private/consortium blockchains are owned by a consortium of organizations or even one organization. Contrary to public blockchains, most private and consortium blockchains have tailor-made distributions of the permissions each participant is granted. Therefore, these types of networks can be considered *permissioned*. Projects like Hyperledger Fabric (Androulaki et al., 2018) provide frameworks to build these consortium/private networks. There are also blockchains that combine features of both architectures.

Blockchain networks provide the technical infrastructure on which several services like smart contracts can be run. As said, ultimately the blockchain infrastructure potentially combined with a smart contract allow for the creation of DApps. Figure 6 depicts a full stack architecture of a DApp that most platforms use.

Working from top to bottom, the first layer is the front end that like for any normal application serves as the *presentation layer* for end users. As blockchain services are normally offered via the internet, the front end is usually a website. Similar like any

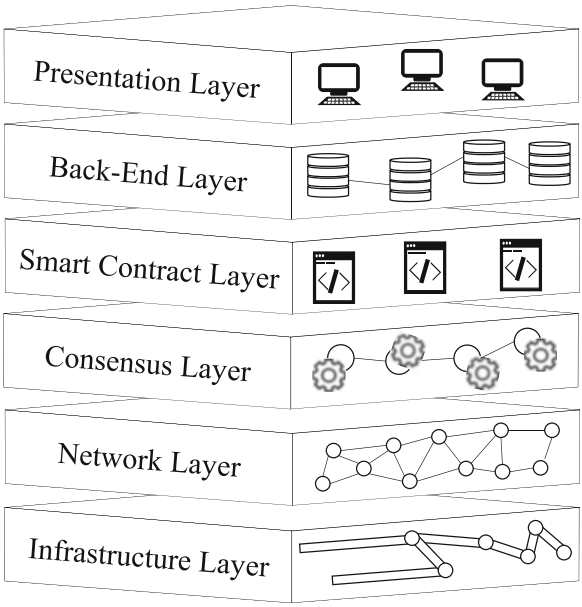


Fig. 6 Full stack architecture of a Decentralized Application (DApp)

normal application, a DApp has a *back-end layer* that processes the programming logic when for instance a user pushes a button. In this case, the back-end usually also sets in motion the actions that a smart contract needs to perform or that needs to be executed on the blockchain. Where a traditional app differs from a DApp is that instead of being connected to a database, a DApp is connected to a blockchain that serves as the point for data storage. Although the back-end is supposed to process the logic within the DApp, it cannot execute any logic used for the blockchain. Executing logic on the blockchain is the purpose of a smart contract that serves as a connector between the users' back-end and the blockchain and forms the *smart contract layer*. This feature is made possible because smart contracts are deployed on the blockchain and users can send transactions to trigger them. Like a normal program a smart contract can be programmed to follow a certain logic when performing transactions. A smart contract could, for instance, store conditions and logic that need to be satisfied before a transaction is executed. Not all blockchain platforms or frameworks cater for smart contracts. As explained in Sect. 2.1, to ensure the validity of transactions and secure the historical record of transactions the nodes in the network need to reach a consensus. The specific set of algorithms deployed to ensure the consensus between the nodes is called the *consensus-layer*. A consensus-layer is the beating heart of the blockchain. Nodes within the blockchain network form a *network-layer* on which the data concerning the blockchain is shared. This data includes the blocks, in other words the data about the transactions but also the code of smart contracts that have been deployed on the blockchain. Communication and distribution of data about the blockchain is shared by the nodes via the *infrastructure-layer*. Nodes are not natural persons but machines or

computers that execute the algorithms required for the blockchain. The standard TCP/IP protocol used for everyday communication on the internet provides the channel for nodes to communicate.

3 Artificial Intelligence

AI is nowadays often the subject of conversation within society. The potential to use AI for a wide variety of processes has led organizations to explore how they could harness its potential. Some examples of processes for which AI is employed are fraud detection, marketing, Siri on your phone. Although AI is often referred to as one technology, the term actually represents a broader concept of intelligence demonstrated by machines. The term AI was coined in 1956 by John McCarthy (1995) that describes it as:

> It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

It is important to note that the field of AI focuses on intelligent machines with a strong emphasis on computer programs. Computer programs encompass a combination of *algorithms* that have been designed to *learn* how to perform a specific task, usually by employing statistics. This notion is important because when evaluating how the AI program performs the task at hand the combination of algorithms needs to be examined. What further can be noted from the definition provided by McCarthy is that the aim of AI is to *mimic human intelligence*. With their intelligence humans are capable of performing several tasks. Researchers and practitioners in the field of AI developed several algorithms over the years that have enabled sophisticated programs to mimic the performance of these tasks. Each of these tasks has over time constituted to specialized subfields of AI.

In the remainder of this section, we will first explore how machines learn to perform tasks in Sect. 3.1. To understand how AI is used in practice, in Sect. 3.4 an overview of all subfields of AI will be provided. Each of these subfields will thereafter be explained, and some important concerns for auditing are discussed.

3.1 How Machines Learn

The effort of letting machines learn in order to perform human-like tasks is collectively called machine learning (ML) (Samuel, 1959). Like humans, machines learn by example. When using ML these examples are provided in the form of a machine-readable data set. Each data set encompasses several observations, or measuring points linked to variables. In turn from each observation several features can be

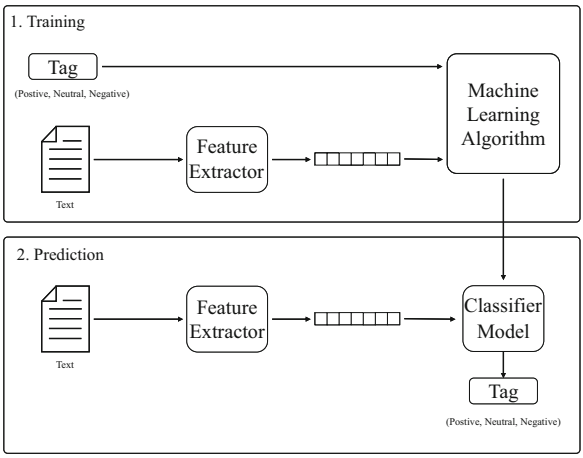


Fig. 7 Separate steps to train a neural network

discerned which are the characteristics or properties of an observation (Bishop, 2006). Relations that the machine has learned are represented as models, that express these relations as parameters, variables, or other mathematical concepts like vectors.

ML algorithms can learn in a descriptive, predictive, or prescriptive manner from a provided data set. These types of learning differ from one another because the aims of the learning process are different. Descriptive learning focuses on extracting relations between features in the data set with the aim of understanding laying bare these relations. For instance, a data set encompassing several customers of a firm can be used to learn how customers are grouped, and on the basis of what characteristics.

Predictive learning is not only aimed at learning relations between features in the data set, but in addition being able to predict what outcome is most likely given a certain input. An example of a task that such an algorithm could learn is to predict the likelihood that a customer will make an insurance claim based on several demographic factors. Similar to descriptive learning, when learning to predict outcomes ML algorithms first examine and learn the relation between features. However, the important difference is that these relations are considered independent variables that serve to predict one or many dependent variables. Getting back to our insurance example, in this case the goal is to predict whether someone will make an insurance claim (dependent variable) based on other independent variables like demographics and so on. The creation of a model to predict outcomes generally takes place in two steps: (1) Training and (2) Prediction (Ashmore et al., 2021). Both steps are depicted in Fig. 7.

An algorithm written with the purpose of training inspects a set of machine-readable observations provided as the input data. These observations serve as examples for the algorithm to determine how the input with certain features is related to certain outcomes. For instance, how demographic factors like postal code, age, and income predict whether or not a customer is likely to make an insurance claim. In some cases, a tag (label) is provided as a target that the algorithm should be able to

predict as the dependent variable. The relations between the features and outcomes are then captured in a model. In the next step, called prediction the "fit" of the model is examined. In other words, given a set of provided examples how well does the model predict the expected outcome. Some ML algorithms further improve the fit of the model by using another set of examples to partially retrain the model after an initial training. Again, statistical methods underpin the predictions made using the model.

Prescriptive learning is another approach to ML learning that combines aspects of descriptive and predictive learning with the addition that the algorithm is able to take an informed action based on the data provided. Self-driving cars for instance are not only capable of detecting objects like other cars around them but also to take appropriate action when needed (e.g., hitting the breaks). An important aspect of prescriptive learning is that the algorithm cannot only understand patterns based on prior examples, but can also make informed decisions which action to perform given the information provided.

Besides discerning the algorithms based on its aim, we can also make another distinction between ML algorithms that is related to the manner in which the algorithm is trained or learns from data. The approaches to learn machines are usually divided into three generic categories, based on the nature of stimuli and feedback that is provided to the learning system (Ayodele, 2010):

- *Supervised learning*: For a supervised learning approach the computer is presented by a human with a dataset containing multiple examples with inputs and correct outputs. The main aim of this approach is to learn the algorithm the relations between the inputs and the outputs.
- *Unsupervised learning*: No desired outcomes are provided to the learning algorithm. The algorithm itself has to determine what relations exist in the data set. Note that the discovering these relations or patterns in the data can be the aim itself, or a means towards an end (e.g., to subsequently predict a relation).
- *Reinforced learning*: When employing reinforced learning, a computer interacts in a dynamic environment. In this environment, it must be able to perform a specific task such as driving a vehicle or vacuum clean your house as a robot. While carrying out the task the algorithm is provided with feedback from the environment through which it learns to maximize efficiency. Using this approach, the algorithm learns by trial and error.

What can be noted when closely examining these different types of learning is that they can be discerned based on how and when the input for training is administered. When using unsupervised learning, the builder of the algorithm does not offer any of his own knowledge to the algorithm. In supervised learning, this knowledge is offered by providing the algorithm with examples of the data and classifying (labeling) each example. For instance, providing a set of messages with coherent classification of the sentiment of the message (e.g., angry, happy, or sad). This also introduces hazards however, because what if the provider of the examples made a misjudgment about what sentiment a message, or even several messages actually have. In other words, what if the provider of the examples has provided the

wrong examples to the algorithm. Obviously, this would greatly reduce the accuracy of the ML algorithm because it learns from incorrect examples. To diminish the possibility of errors when providing examples for supervised learning, it is desirable to maintain a *four eyes principle*, meaning that at least two or more distinct persons independently label each example provided to the algorithm as input. The distinct sets of independently labeled examples are then compared for agreement. The measurement of the agreement between two raters is called *inter-rater reliability* and serves to provide an indication about the reliability of the labeling of the dataset (LeBreton & Senter, 2008). Several tests like Krippendorf's Alpha, and Cohen's Kappa can be used to measure the inter-rater agreement. However, the process of labeling examples is often arduous and time consuming. Therefore, instead of examining all of the examples provided by another person it is common to only assess a sample.

3.2 Deep Learning and Neural Networks

Oftentimes deep learning is discerned as another subset of machine learning. Like "normal" machine learning deep learning can be employed for descriptive, predictive, and prescriptive purposes and can also be taught to learn using a supervised, unsupervised, or reinforced learning approach. What sets deep learning apart from other machine learning approaches is how the relations between features are stored. Neural networks often consist of many hidden layers to extract and store features from data. In essence, neural networks are data structures modeled to resemble the human brain. Figure 8 depicts a schematic version of a neural network.

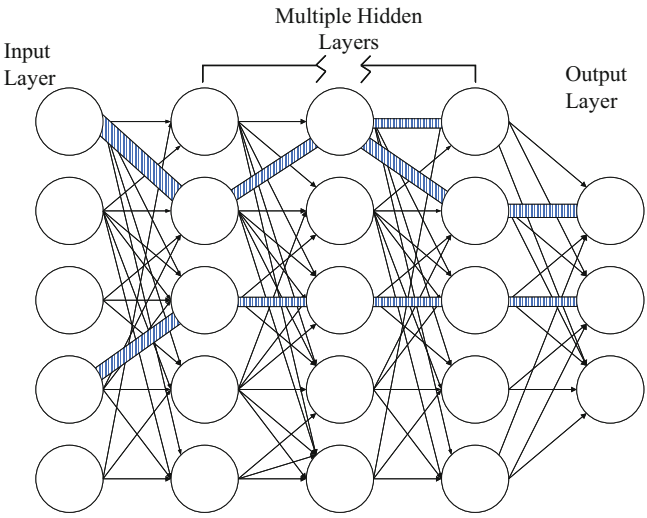


Fig. 8 Schematic depiction of a neural network

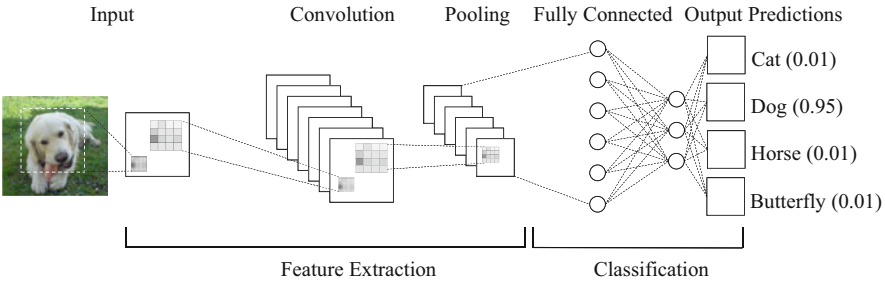


Fig. 9 An architecture for a convolutional neural network

Neural networks are vastly complex multi-layered networks. Similar to a human brain, a neural network encompasses several nodes (similar to a neuron) that are inter-connected which allows data to be passed between them. The neural network always encompasses an input layer and an output layer. In between the input and output layer there are multiple hidden layers. Some neural networks can encompass millions of hidden layers, whereas others only have 20. The hidden layers in a neural network pass on data from the input layer and provide a subsequent outcome to the output layer. Due to the complexity of neural networks it difficult, if not impossible, to understand what happens when data is passed between the nodes. Therefore, neural networks in all of their different shapes and sizes are considered a *black box*, meaning that we know the input and the output of the algorithm but not what happens during the processing of the data. How neural networks are structured strongly depends on the deep learning algorithm used to perform a task. In turn, research (Pouyanfar et al., 2018) has demonstrated that some types of deep learning algorithms are more suitable than others for a specific task. Hence, there is often a strong relation between the task at hand and the type of neural network employed to store the data. Roughly speaking neural networks can be divided into two groups: convolutional neural networks and recurrent neural networks.

Convolutional neural networks are predominantly used for image recognition. Hence, the input to train these neural networks is almost always an image. Figure 9 depicts a typical architecture for convolutional neural networks.

The typical architecture of a recurrent neural network encompasses several layers. However, they perform the two distinct tasks of feature extraction and the classification. In the input layer images are provided to the convolutional neural network model in the form of a matrix. Next, the images are passed on to the convolutional layer that performs the mathematical operations. Each image is then convolved with a separate square matrix that functions as a kernel or filter. The kernel is then slid over each pixel of the image to attain a feature map that contains the information about features of the image such as edges and lines. However, raw feature maps consume vast amounts of memory and are computationally expensive. Therefore, after convolving the image a dedicated pooling layer diminishes the size of the feature map. Several types of formulas like max pooling, average pooling, and sum pooling can be used for this purpose. The last layer or fully connected layer is used to

Table 2 Differences between a CNN and RNN

Convolutional neural network	Recurrent neural network
CNNs are neural networks for deep learning that is predominantly used for image processing.	RNNs are neural networks that are commonly used temporal and sequential data. An important feature of RNNs is that the nodes in the network are sequentially connected allowing for the creation of memory.
CNNs are feed-forward network that require little preprocessing, made possible by multi-layers of nodes.	An RNN can use its internal memory to handle different sequences of input.
Compared to an RNN, a CNN is far more powerful.	RNNs can include and combine far less features compared to a CNN.
A CNN always takes fixed size inputs, and returns fixed size outputs.	An advantage of an RNN is that they can process different sizes of input versus output.

make predictions over the images as they are activated. Although recurrent neural networks are helpful in many aspects, they are not particularly useful to process temporal or sequential data (e.g., a movie).

Recurrent neural networks are better equipped to work with temporal or sequential data. This is largely due to the fact that recurrent neural networks use the input of prior nodes in the network to weigh in their information in order to establish the relation between input and output. Effectively this constitutes an internal memory that is able to distinguish important details such as those related to the input they received. Using its memory, the neural network is able to predict what will come next. This important characteristic of a RNN makes them highly usable for tasks related to speech, video, and text. The key takeaway about RNNs is that when sequence is of the essence, a RNN will learn a far more profound understanding of the sequence as compared to other algorithms.

What sets a RNN apart from a CNN is that the output that has been passed through a prior step is provided as input to the current step. A RNN has therefore two inputs: data concerning the current step and data concerning the recent step(s). This memory build-up is pivotal because the chain of information that is forwarded to each step is what makes that a RNN performs so well on sequential tasks. Contrary to CNNs, the hidden layers of a RNN actively memorizes information about the calculations on the sequential data it has been trained on. Like a CNN the size of a model can vastly increase depending on the task it is trained for. To reduce the complexity and thus size of the model the same parameters are used for each task. The differences between the two types of networks are summarized in Table 2.

3.3 Measuring the Accuracy of Machine Learning Algorithms

When using algorithms to predict or even prescribe certain outcomes, assessing the accuracy of an algorithm is important for auditors. What we mean here by accuracy is how well the model is at doing its task in predicting the right outcome. The accuracy of predictive or prescriptive ML algorithm can be verified by using another distinct set of examples as input and then scoring how many times the algorithm performed the task in line with the right outcome. A dedicated metric to measure the accuracy can then be employed to calculate the accuracy. Because there are several statistical techniques that enable machines to predict, over the years several metrics have been developed to test the efficacy of an ML algorithm. The simplest of these metrics is to measure the precision of an algorithm. *Precision* in this context means how many of all of the observations predicted by the algorithm as positive were actually positive. We can calculate the precision by using the following formula:

$$\text{precision} = \frac{\text{true positives and selected elements}}{\text{Selected elements}}$$

To explain this formula, consider that we have an algorithm that is built to predict whether there is a tree on a picture or a house. In the set of pictures that are provided to the algorithm there are 12 pictures of a tree and 12 with a house, making a total of 24 pictures. The algorithm predicts that in this set of 24 pictures there are 9 pictures that contain a tree. However, in reality of these nine selected pictures there are only four trees on the picture the other five are houses. We call these four correctly predicted pictures with trees *true positives*, while we refer to the total of nine pictures as the *selected elements* as they are predicted by the algorithm.

Another important metric is *recall* also referred to as sensitivity that measures the ratio of correctly identified elements (true positives) among the total of relevant elements in the entire set. Coming back to our example, the relevant elements here are all the pictures with a tree depicted on it (total of 12). We can calculate the recall for this example using the formula:

$$\text{recall} = \frac{\text{true positives and relevant elements}}{\text{relevant elements}}$$

Contrary to the precision metric we use the identified true positives and the total known of relevant elements to calculate. Taking the same example again we would now use the 4 pictures of the tree and divide it by 12. Although at a first glance precision and recall seem appropriate metrics to measure the accuracy of an algorithm, they have some disadvantages. For instance, what if both the precision and recall of an algorithm matter? It is not unreasonable to say that both do and thus to address this problem the $F1$ score was introduced. Using a $F1$ score as a metric is

especially popular because it measures the harmonic mean between precision and recall. The formula to calculate the $F1$ score is as follows:

$$F1 = \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Because we already know how to calculate the precision and recall we can simply plug in these calculations into the formula. Where we multiply precision by recall and dividing it by the product of precision and recall. Please note that despite the fact that the $F1$ score is a commonly used metric there is an ongoing debate on the appropriateness of the metric. In the example here above, we only used the $F1$ score to calculate an algorithms performance on two classes. However, an adjusted version of the $F1$ score can also be used for multi-classification testing.

3.4 Using AI in Practice

As mentioned already the main aim of employing AI is to let computers perform tasks otherwise carried out by humans. Over time, a logical division of these tasks has led to the creation of several subfields within the AI domain. In Fig. 10, these subfields are portrayed.

Some of these subfields overlap and this overlap can be attributed to the fact that because these subfields are organized by task, some or almost all of them are one way or another related. Let us now further explore how each of these tasks is performed by AI algorithms.

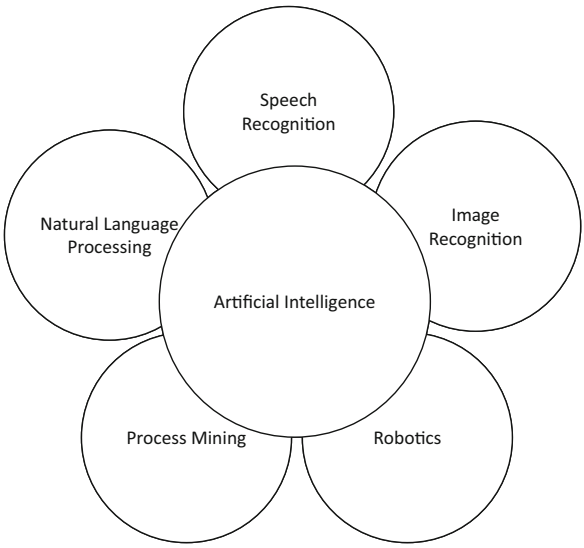


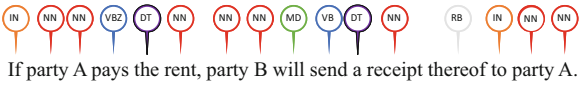
Fig. 10 Representation of subfields in AI. Note that this depiction is not exhaustive and some fields may be missing

3.4.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of AI that focuses on developing approaches to enable machines to understand and generate written natural (human) language. The goal of NLP is to create an IS that can sensibly process text to perform a variety of tasks like spell checking, determine the sentiment of a text, or extract relevant information from a text. To understand how these algorithms are able to perform these tasks, we must first understand that computers are ill equipped to perform tasks on text because they are meant to calculate, not interpret. A human reader when presented with a text has learned to discern paragraphs, sentences, words, and letters. Computers however would consider a text merely as a sequence of characters (mostly letters) and without human guidance do not have the capacity to identify sentences or even words. However, paragraphs, words, and other characters often are employed in NLP as features. For a machine to learn relations in texts, whether that be in a descriptive, predictive, or prescriptive manner, these features first have to be created.

Most NLP algorithms therefore require that a certain piece of text is first split into units that serve as an observation. For example, if we want to create an algorithm that is able to predict what sentiment (e.g., angry, happy, or satisfied) a customer review has we take the whole review as the observation. Instead of multiple sentences, single sentences or even single words could also be the unit of observation. This would make it possible for instance to classify a word as being a verb, noun, or other. However, merely dividing a text into observation units usually does not provide enough features for ML to identify meaningful relations. To remedy this problem most ML algorithms for NLP employ a technique called *tokenization* (Webster & Kit, 1992). Tokenization means that an algorithm is employed to divide the set of characters that the text encompasses into a set of strings (like words) that each contain sequence of tokens. At the most generic level the algorithm can predict for a given sentence to what class it belongs. A common application for this is to determine whether customer can be classified as angry, sad, or happy. However, at a more granular level NLP algorithms are able to classify words.

NLP is predominantly used for natural language understanding by analyzing pieces of text for either syntax or semantic meaning. Syntactic analysis involves creating algorithms that are able to dissect the syntax of a sentence, paragraph, or entire text. A well-known task for instance is *Part-of-Speech tagging* (POS) where an algorithm is tasked with syntactically classifying and predicting whether a word is a noun, verb, or coordinating conjunction. An example of what the output of a POS task looks like is depicted in Fig. 11.



Legenda: **IN** = Preposition. **NN** = Noun. **VBZ** = Verb, 3rd person singular present. **DT** = Determiner. **MD** = Modal. **VB** = Verb, base form. **RB** = Adverb.

Fig. 11 Labels related to words when using POS tagging

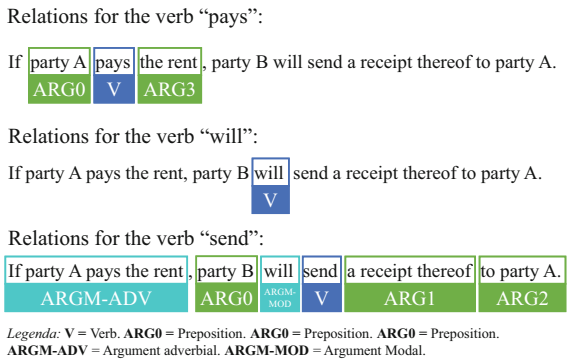


Fig. 12 Labels related to different verbs and words when using SRL

Semantic analysis focuses on understanding meaning within a text. This task goes well beyond merely dissecting a sentence by predicting whether words are nouns, verbs, and so on. As the name suggests, the aim of *semantic role labeling* (SRL) is investigating which parts of for instance a sentence play what role. To discern the different roles a part of the sentence has these are connected to verbs. Like POS that is very much akin to SRL a dedicated annotation schema is needed usually also with integrated BIO (Begin Inside Outside) tagging that indicates where a role starts and ends. The most annotation schema used for SRL is that by Palmer et al. (2005). To explain how SRL works take the following sentence: "If party A pay's the rent, party B will send a receipt thereof to party A. The algorithm would first try to predict all of the verbs in the sentence, and then for each of these verbs predict what the relation is between the verb and other parts of the sentence. In the case of our example, this would yield the result depicted in Fig. 12.

The explanation of the labels is omitted here for brevity's sake but further information can be publicly consulted.[4] Taking this notion a step further, practitioners and scholars have started to design algorithms for information extraction. One important part of information extraction is Named Entity Recognition (NER) where NLP algorithms are used to find people, dates, and places in a text. Information extraction also relies on SRL as a basis but an additional algorithm is used a top of a SRL algorithm to give the labels more contextual meaning.

Besides using NLP algorithms to analyze existing text, they are also employed to generate new text. This task is called natural-language generation (NLG) and it serves to produce natural language as output (Reiter & Dale, 1997). The general idea behind NLG is that instead of letting a human author a text, a machine will perform this task. In practice, NLG is used for a tremendous number of applications like (Gatt & Krahmer, 2018):

1. Checking spelling and grammar to suggest text corrections.
2. Generating paraphrases or responses.

[4]Please visit: https://www.cs.rochester.edu/~gildea/palmer-propbank-cl.pdf for a guide of the labels.

3. Translating texts from one language to another.
4. Simplifying complex texts to make them easier to read for a broader audience.
5. Text summarization to automatically create abstracts from long texts.

Defining the difference between natural language understanding and NLG is oftentimes hard (Gatt & Krahmer, 2018). In practice, these aspects are combined to attain a certain result. To create a chatbot for instance, SRL is employed to make the machine understand what a customer is asking. Then NLG is used to formulate an answer to the customer's question.

3.4.2 Speech Recognition

Speech recognition once was considered a subfield of NLP. However, recently it has developed into a full-fledged interdisciplinary subfield of computational linguistics. The aim of speech recognition is to develop methodologies and coherent algorithms that enable computers to recognize and translate spoken language into text, or machine-readable format. A prime example of speech recognition usage is Apple's Siri, or the Alexa home appliance from Amazon. Both use a sophisticated speech recognition algorithm to capture and process spoken language with the aim to understand what a user is commanding them. These interpretations then prompt the program to execute whatever the user is asking. *Voice recognition* can be considered another aspect of speech recognition. Algorithms for voice recognition are not designed to understand a users' commands but recognizing different users. Again, like all ML techniques speech recognition algorithms learn from features, in this case the audio provided to train the algorithm. Compared to NLP speech recognition uses several different features:

1. Language weighting: When mentions of words are of interest, the algorithm can be trained to listen to a particular set of words. Training the algorithm to specifically identify these words increases the chance of filtering out conversations or audio of interest based on subject.
2. Acoustic training: Inevitably with some audio there is ambient sound or other noise pollution. Acoustic training serves to aid the algorithm to discern for instance background noise and speak.
3. Speaker labeling: For voice recognition, speaker labeling is important to understand who is speaking, and by extension who is saying what in a conversation. Algorithms trained on this aspect are able to discern several speakers at once and translate their contribution.
4. Profanity filtering: The use of profanity filtering is to detect specific words in a conversation to filter them out or extract them. This feature differs from language weighting as it is designed as a filter not to identify conversations of interest among for instance a set of audio fragments.

Like other ML architectures, speech recognition algorithms are made up out of several components. First, there is the speech input that consists out of multiple

audio fragments. Based on these audio fragments, a model is created containing several feature vectors that capture a myriad of relations between features like tone and length of a tone. When using the model in practice, a decoder is required to interpret the outcomes that have been attained through the use of the features. The decoder itself employs pronunciation dictionaries, language models, and one or more acoustic models to attain the right output.

Although great improvements have been made to speech recognition algorithms, the current best score was made by Google Cloud Speech in 2017. Their algorithm yielded a score of 95% with an error rate of 5%. In itself this error rate possesses no problem to the use of the algorithm. However, speech recognition algorithms are often used in combination with an NLP algorithm that also has an error rate compounding the errors in the final output. Consider the example of a speech assistant of Google Home, Amazon's Alexa, or Microsoft's Cortana; These systems employ speech recognition algorithms to understand when a command is given to them by whom and translate this to text with a potential error rate of 5%. In a sequential step, an SRL algorithm (NLP) uses the text created as output by the speech recognition algorithm (with the errors) as input to determine what the user has commanded. The SRL algorithm thus receives input with errors that in turn is far more prone to generate wrong output, not even taking into account the error rate of the SRL algorithm itself.

3.4.3 Image Recognition

Image recognition is a strand of AI methods that focuses on classifying images. The applications for these algorithms are around us everywhere. A prominent example is the face recognition on most smart phones. Image recognition is also used for self-driving cars that need to recognize obstacles on the road, or find persons of interest on camera footage. An image recognition algorithm can perform several tasks:

1. Classification: This task involves classifying that what "class" the image belongs. For instance, the depiction of a dog or a cat.
2. Tagging: Is a task similar to classification, but is more fine-grained. The tagging task involves identifying (potentially) multiple concepts and/or objects in an image. For one image several tags or labels can be appropriate.
3. Detection: When the algorithm is assigned to identify and locate an object in an image (or video), it is a detection task. An example for the use of such an algorithm is software for self-driving cars.
4. Segmentation: This task is similar to detection however, but is yet again somewhat more fine-grained. The algorithm is able to locate objects on a pixel level which is sometimes required for very precise identification.

As explained in Sect. 3.2, CNNs are predominantly used for image recognition. When training ML algorithms for image recognition, the trained weights and biases are assigned to several parts of the image that serve as the features so that they become indistinguishable from one another. Based on the knowledge on these

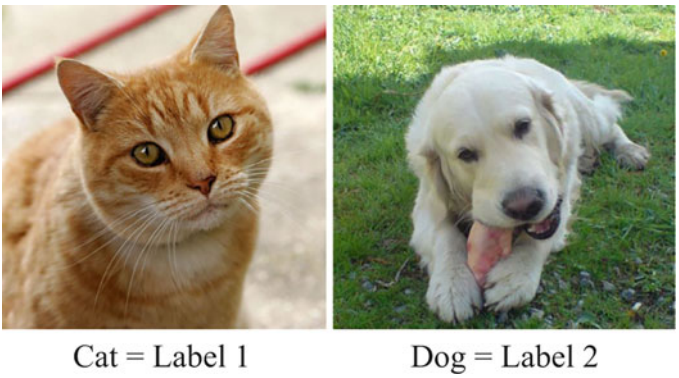


Fig. 13 Classification task input per image

distinctions, the recurrent neural network can be activated for several tasks like image recognition, object and face detection, and image recognition using a set of activation functions. The training examples (i.e., images) can be both labeled for supervised learning and unlabeled for unsupervised learning. The algorithm regards each input image as an array of pixels translated as a matrix. This matrix usually pertains data in the form of Height × Width × Dimension. To illustrate how this works, consider an image of 20 pixels × 15 pixels × 1 where the 1 denotes the RGB color. The range of the numbers that are stored in the matrix is referred to as the color depth. Hence, the color range strongly dictates the maximum number of colors that can be used. For RGB colors that are a mixture of red, green, and blue often used in images, this range is from 0 to 255. After converting the matrices to a plethora (sometimes millions) of features, labels can be added to the images to train the model.

In Fig. 13, a training example is shown for an image recognition algorithm that is trained perform a classification task. Figure 14 depicts how training images are labeled for image recognition algorithms that carry out detection and segmentation tasks.

Whereas in Fig. 13 the entire image is labeled, in Fig. 14 the objects in the image are "boxed" with a red line. Unless programmed to do so, an image recognition algorithm does not provide a "boxed" picture as output but only a tag (if any).

3.4.4 Process Mining

The purpose of process mining is to discover a process in the context of an organization and potentially make predictions or prescriptions about how the process takes place. Although in the past most processes were carried out by hand, nowadays most activities within an organization are performed using a computer. The fact that most processes are now carried out using a computer makes process mining easier and more predictable. When carrying out actions via a computer, a log of activities is

Fig. 14 Image recognition task with different classes of labels

created containing all data related to the sequence of activities a user has carried out. Specialized process mining algorithms can be used to mine a process from a set of logs. These algorithms discern several activities from each individual activity log to identify sequences that are shared across all provided audit logs. There are several statistical techniques available for this purpose but the most popular is clustering that is used for description of a process. AI and machine learning can further help in process description by detecting anomalies and finding processes that are similar based on an example.

When a process has already been discovered and laid out, *diagnosis* might also be useful. Consider for example the case where a process is known to be performed sub-optimal. A sub-optimal performance of a process might have many causes. ML can be harnessed to find the causes of a problem by reasoning back and generating a root-cause analysis of the problem. If any problems have been identified during the execution of a process, ML can also be employed to classify these problems. In turn, the classification of the problems makes it easier to remedy them. The evaluation of the changes that may have occurred to the process over time might also be mapped using ML to spot trends.

Knowing how processes have been carried out in the past is useful to prevent errors, delay, and other problems in the future. Machine learning based process mining is also employed in practice to monitor ongoing processes and make *predictions* about the next event that will occur, how the process will influence certain outcomes, or even the final outcome of a process. Similar to other AI applications, once a machine has learned how to predict events or outcomes it can act *prescriptive*. For instance, when a process will occur during a process it can send an email to the appropriate person to notify them of the problem. Or, when the AI is advanced enough, activate robots or programs to solve the problem.

3.4.5 Robotics

Robots are perhaps the most classical picture that we have in our minds when we imagine AI. However, robots do not necessarily possess AI to perform their tasks. The word robot was first introduced by Karel Čapek in 1920 with the connotation that we know it for today. In Czech the word "robota" means "labor" or "compulsory labor." A robot is a machine that performs physical labor in the form of one or many steps, usually in a specific sequence. In the car industry for instance welding robots are now commonplace to perform welding tasks. Robots have the following characteristics:

1. All robots are composed of a material mechanical construction that allows it to interact with its physical space and to manipulate it. This construction can include several sensors to perceive the environment, and mechanical instruments to perform actions.
2. Robots need a power supply to function and feed their mechanics with power. Not all robots use electricity for this purpose, e.g., steam is another supply of power that can be used.
3. At least some sort of algorithm is needed to instruct the robot what to do and how. The absence of the computer program would mean that the robot is a piece of simple machinery. Because the program instructs the robot what to do it is able to operate in the physical space.

At first glance, robots and computer programs seem alike and even interchangeable. An important difference between robots and computer programs is that a program does not carry out a task in physical space but rather only virtually (Luckcuck et al., 2019). The physical activities are made possible because robots are a combination of software and hardware components. Some of the software used for robots is simply an algorithm that always performs the exact same steps, or is programmed by a user to follow a sequence of different steps. More advanced robotics that employ AI to determine the sequence of steps they need to take are called *autonomous robotic systems* (Luckcuck et al., 2019). Like most applications of AI, autonomous robotic systems touch upon ethical and legal considerations, however compared to other AI applications safety is an even more import aspect. As autonomous AI based systems can manipulate their environment, they can physically harm their environment, including humans.

To prevent unsafe situations, the physical environment of the robot can be modeled. Two approaches are predominantly used for this purpose: the workplace of the robot is modeled or the environment itself is continuously monitored. The first approach has been proven to be extremely difficult in dynamic environment where all potential future circumstances that may lead to unsafe situations need to be captured. Continuously monitoring the robot leads to similar problems in that unsafe situations need to be known in advance in order for them to be prevented. Providing trust and required certification evidence is challenging for auditors (Luckcuck et al., 2019). Formal methods are a commonplace to ensure the correctness and safety of

(software) systems, and thus to provide trust and certification evidence. However, hitherto there is not one uniform widely accepted formal method that has been adopted for the development of autonomous robotics. Thus, developers are provided with few guidelines to select the appropriate formal method to build and verify an autonomous robot (Kossak & Mashkoor, 2016). More important, the technology for autonomous robotics is still in its infancy. Consequently, regulations on the topic are still being developed, making it difficult for certification bodies to establish criteria for an audit (Webster et al., 2014). Besides these safety concerns, another notable problem is how to coordinate swarms (several) of autonomous robots that have to operate in concert to attain a goal. Because these swarms magnify the pre-existing problems with autonomous robotics while adding a coordination problem. The introduction of machine learning enhances the complexity of autonomous robotic systems even further by obfuscating how the robot has made its decisions making it hard to monitor.

4 Cloud Computing

Cloud computing is a term used for computing services delivered via the internet. These services encompass a broad array of computing resources that are nowadays offered by hundreds of providers like Amazon, Google, Microsoft, IBM, and VMware. Although the term is often used, it is often ill defined. The US National Institute of Standards and Technologies (NIST) provides a broadly accepted and concise description of the generic properties of cloud computing (Mell & Grance, 2011):

- On-demand Self Services: Any client is able to procure computing resources without any human interaction.
- Broadly Accessible: Standard mechanisms and protocols enable the access to the cloud computing resources.
- Pooled Resources: A cloud computing service provider has a pool of computing resources that are allocated and provided to clients on demand.
- Rapid Elasticity: Computing resources can easily be provided, scaled up and down based on the clients' requirements and demands.
- Measured Service: A cloud computing system charges a client based on the resources used. To enable this feat the system must be able to automatically monitor, control, and report to the client how much of the resources have been used.

Hereafter the components of a cloud computing architecture are further explained in Sect. 4.1. The ecosystem of cloud computing is further described in Sect. 4.2.

4.1 Cloud Computing Architecture

To support cloud computing most cloud computing providers employ a three-layered architecture. The architecture encompasses a service layer, resource abstraction and control layer, and a physical resource layer. This architecture is depicted in Fig. 15.

According to the US NIST definition of cloud computing, the *service layer* of the services a cloud provider provides typically encompasses three components (Liu et al., 2011):

1. Infrastructure as a Service (IaaS)
2. Software as a Service (SaaS)
3. Platform as a Service (PaaS)

The *IaaS* component provides the computing resources to the client. These resources include virtual machines (VMs), data storage, connected networks, and other utilities through a service model. The promise and premise of cloud computing is founded on the hardware the cloud provider provides. *SaaS* is the next component in a cloud computing service layer, that refers to the delivery of an application. These applications are delivered via the network (infrastructure) to users of the applications. Users of SaaS divided into several groups: organizations that give access to the software applications, the administrators charged with the configuration of the software application, and end users. For each cloud computing provider, there are several manners to calculate the costs of deploying an application. Some cloud providers charge based on the number of end users that use the application, others on the time used, volume of the data stored or its duration. The *PaaS* component binds the other two layers together and provides a platform for the client of a cloud provider to use tools and other resources to develop, test, and deploy their applications. During the life span of the application, the PaaS layer also enables the management of the hosted application. Among the users of the PaaS are application developers, testers, owners, and administrators.

Between the service layer and the physical resource layer is the *abstraction and control layer*. The abstraction and control layer are employed by cloud providers to

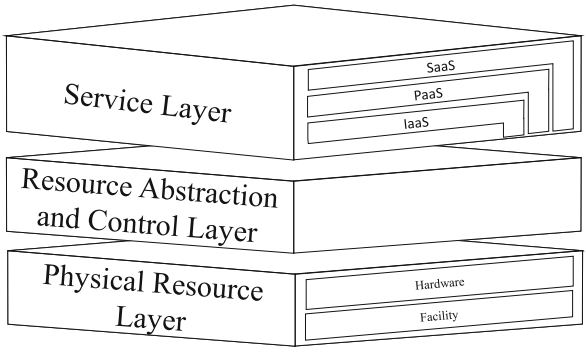


Fig. 15 Reference architecture cloud computing

control the accessibility to physical cloud computing resources. To enable the management and controlling of the physical resources, software abstraction in the form of hypervisors, virtual machines, and virtual data storage is utilized. The layer is especially important because it is used to allocate computing resources to clients, control access, and compute costs. The bottom most layer is the *physical resource layer*. All physical computing resources like CPUs, memory, networks, and data storage facilities are part of this layer. Most physical computing resources also need a plant with their own resources where they are installed. These plants and coherent facilities are also part of the physical layer.

4.2 Cloud Computing Ecosystem

A cloud computing ecosystem encompasses several actors. There are cloud clients, providers of the cloud, cloud carriers, brokers of the cloud and auditors. Within the ecosystem each of these actors are entities that are involved in the processes that take place using the cloud. *Clients of the cloud* provider make use of the services a cloud provider provides. Cloud providers usually offer a catalogue of the services they provide from which the cloud client can make selection. After selecting the desired services, the clients of the cloud provider can immediately make use of it. The services provided by a cloud provider are not for free and a service agreement for payments must be made. Not only the payment terms for the services are important. Clients of a cloud provider might have specific technical requirements for the services they consume. *Service Level Agreements* (SLAs) are commonly used to stipulate the technical requirements and performance the client and provider have agreed upon. These technical requirements may include specific details on the level of security, quality of the services, and potential remedies when the service fails to deliver.

Cloud providers are entities tasked with guaranteeing the availability of cloud services to interested clients. In effect this task includes providing the required infrastructure, managing and running the clouds' software, and providing access to clients of the cloud via a network. Maintenance is another task of the cloud provider that involves servicing any software and updating databases used by the clients. Because the clients develop the software applications themselves, cloud providers often offer several development and management tools for their platform. Some examples of these tools are integrated development environments (IDEs) and software development kits (SDKs). These tools aid the clients in developing and deploying their application on the platform of a cloud provider. Although clients can deploy and control their application via the provider, they have no control over the operating system and other aspects of the platform.

Recently cloud computing has become very complex and this makes it hard for cloud clients to manage their consumed services. This need is addressed by *cloud brokers* that indirectly offer the services of a cloud provider. Cloud brokers provide *service intermediation* by enhancing a service that is originally provided by a cloud

provider to enable some additional capability. Sometimes cloud brokers *aggregate services* from several cloud providers into one main service. Service arbitrage is akin to aggregating services with the difference being that the arrangement of the services is flexible.

5 Conclusions

The complexity of recent novel technologies like blockchain, AI, and cloud computing constitute a genuine challenge to IT-auditors tasked with auditing these IS to provide assurance. The first step towards a clear understanding of how these complex technologies can be audited is to understand the technology itself. This chapter provides the basis of such understanding. Blockchain technology is a complex technology because many sophisticated technologies are combined to create one IS that is able to process transactions without a trusted intermediary like a bank. Smart contracts add more complexity and potential to the technology by allowing for conditional transaction logic. Combined, this constitutes to a unique technology stack.

The term artificial technology is often used for a wide variety of algorithms with different tasks. In this chapter we discussed that the type of algorithm employed and how it learns to perform its task determines how to investigate a particular AI algorithm. To provide a broader perspective, we explain some of the fields for which AI is employed. This overview clearly shows that the term AI should be nuanced in terms of the algorithms discussed, and the task at hand.

Cloud computing is another complex technology that is explained in this chapter. A key takeaway from this discussion is that the term cloud computing is not always concisely used. The definition suggested by the NIST provides clarity by stating the properties of cloud computing. Cloud computing has a three-layered technology stack, that generally speaking provides three types of services to its clients. Nowadays a comprehensive ecosystem has developed around cloud computing. Within this ecosystem there are several actors that fulfill their own role.

References

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., ... Yellick, J. (2018, April). Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference* (pp. 1–15).

Ashmore, R., Calinescu, R., & Paterson, C. (2021). Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys (CSUR), 54*(5), 1–39.

Ayodele, T. O. (2010). Types of machine learning algorithms. *New Advances in Machine Learning, 3*, 19–48.

Back, A. (2002, augustus 1). *Hashcash: A denial of service counter-measure*. Hashcash. Retrieved from http://www.hashcash.org/papers/hashcash.pdf

Bishop, C. M. (2006). Pattern recognition and machine learning, 128(9) Springer

Buterin, V., Wood, G., & Wilcke, J. (2016). *Ethereum homestead documentation*. Ethereum Community. Retrieved from https://ethdocs.org/en/latest/

Chaum, D. L. (1979). *Computer systems established, maintained and trusted by mutually suspicious groups*. Electronics Research Laboratory, University of California.

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research, 61*, 65–170.

Kossak, F., & Mashkoor, A. (2016, May). How to select the suitable formal method for an industrial application: A survey. In *International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z* (pp. 213–228). Springer.

Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In *Concurrency: The works of Leslie Lamport* (pp. 203–226). Association for Computing Machinery.

LeBreton, J. M., & Senter, J. L. (2008). Answers to 20 questions about interrater reliability and interrater agreement. *Organizational Research Methods, 11*(4), 815–852.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. *NIST Special Publication, 500*(2011), 1–28.

Luckcuck, M., Farrell, M., Dennis, L. A., Dixon, C., & Fisher, M. (2019). Formal specification and verification of autonomous robotic systems: A survey. *ACM Computing Surveys (CSUR), 52*(5), 1–41.

McCarthy, J. (1995). What is artificial intelligence? *Annali di Matematica Pura ed Applicata., 169*, 321–354.

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards and Technology.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system* (Decentralized Business Review, 21260). Satoshi Nakamoto Institute.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics, 31*(1), 71–106.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR), 51*(5), 1–36.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering, 3*(1), 57–87.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development, 3*(3), 210–229.

Sherman, A. T., Javani, F., Zhang, H., & Golaszewski, E. (2019). On the origins and variations of blockchain technologies. *IEEE Security & Privacy, 17*(1), 72–77. https://doi.org/10.1109/MSEC.2019.2893730

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*.

Szabo, N. (2005). *Bit Gold*. Nakamoto Institute. Retrieved from https://nakamotoinstitute.org/bit-gold/

Webster, J. J., & Kit, C. (1992). Tokenization as the initial phase in NLP. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*.

Webster, M., Cameron, N., Fisher, M., & Jump, M. (2014). Generating certification evidence for autonomous unmanned aircraft using model checking and simulation. *Journal of Aerospace Information Systems, 11*(5), 258–279.

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., & Rimba, P. (2017, April). A taxonomy of blockchain-based systems for architecture design. In *2017 IEEE International Conference on Software Architecture (ICSA)* (pp. 243–252). IEEE.

Zheng, Z., Xie, S., Dai, H. N., Chen, W., Chen, X., Weng, J., & Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems, 105*, 475–491.

The Intercompany Settlement Blockchain: Benefits, Risks, and Internal IT-Controls

Rewin J. M. Doekhi

1 Introduction

Blockchain is a novel technology that can be employed for a plethora of business applications. In recent years, blockchain projects are gaining traction using new blockchain and Distributed Ledger Technology (DLT) platform providers, such as Ethereum, Corda, and Hyperledger (Revet et al., 2021). Blockchain technology establishes a protocol of trust between mutually distrusting parties by employing a combination of a Secure Hash Algorithm (SHA), encryption technology, and Peer to Peer internet networks (P2P). This combination of technologies ensures that a blockchain provides highly reliable transaction data, and as such the technology is finding increasing support among banks and other companies (Rauchs & Hileman, 2017).

A blockchain provides highly reliable transaction data because it has a number of properties that guarantee the integrity of the data that is processed and stored. Using encryption techniques, transaction data becomes immutable and thus tamperproof, while a consensus mechanism ensures that new transactions are validated and the nodes in the P2P network verify the transactions and broadcast them to the other nodes (Kloch & Little, 2019; Butijn et al., 2020; Yaga et al., 2019). These properties aid companies that employ a blockchain for accounting in exerting more control over their financial processes. Paradoxically, the use of blockchain technology is associated with peculiar risks that need to be controlled (ISACA, 2021; Bernsen et al., 2019). Therefore, the sophistication of this technology warrants a rethink of current

The author would like to express his gratitude for the support of Eric Westhoek from Erasmus School of Accounting & Assurance and the KLM Royal Dutch Airlines former Vice Presidents Internal Audit, Eric Wittgen and Gijsbert Woelders.

R. J. M. Doekhi (✉)
Amsterdam, The Netherlands
e-mail: Rewin.Doekhi@klm.com

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_4

IT-controls to manage accounting processes while novel controls need to be established to control the technology itself. In this chapter we discuss the impact of blockchain technology on IT-controls, and what IT-controls should be in place to mitigate risks using a case study of a blockchain implementation at the KLM Royal Dutch Airlines.

The remainder of this chapter will proceed as follows. In the next section (Sect. 2), a brief background on IT-controls will be provided. Furthermore, an examination of the impact of IT on financial processes, the external financial accountability, and the role of the IT auditor is provided. Subsequently, Internal Control over Financial Reporting, ICFR, and the interaction of the IT General Controls (ITGC), Application Control (AC) and Manual Controls (MC) in the Business Control Framework (BCF) quadrant are discussed. Finally, the SOx act in relation to COSO and COBIT is explained with a focus on IT-controls. The section concludes by discussing the benefits of blockchain for accounting. In the third section, the case study of a financial in-company blockchain settlement process is described. The case study depicts this process both with and without a blockchain application. In the fourth section, we present the analysis of the impact of blockchain on the IT-controls discussed in the case study. A graph shows the impact of the corporate blockchain at the process and IT level on IT-controls at the entity, process, and ITGC level. These IT-controls are related to the identified risks of the chosen DLT implementation. Finally, in the conclusion section we present several conclusions on the basis of the analysis in the case study.

2 Internal Control over Financial Reporting (ICFR): "IT-Controls"

Large companies are required to annually organize an internal audit on Internal Control over Financial Reporting (ICFR) by an external auditor, as a result of the Sarbanes-Oxly (SOx) Act 404, enacted in response to large-scale accounting fraud within the chemical group Enron. The internal audit control looks at the "design effectiveness" (type 1, design and existence) and "operational effectiveness" (type 2, operation) of controls. The costs and effort of setting up an ICFR and performing the annual audits are relatively big. Besides meeting legal obligations imposed by the SOx, an effective ICFR provides benefits such as (Center for Audit Quality, 2019):

- Reasonable assurance that financial records, knowingly or unknowingly, have not been misrepresented. It exposes the material weaknesses and imperfections.
- A structure that identifies and mitigates risks and raises awareness and alertness at all levels of the organization to prevent fraud.
- Guarantees the integrity of the financial reporting and creates an image of trust towards stakeholders.

Despite concerns raised by firms about regulatory burdens imposed by the US Securities and Exchange Commission (SEC) and the costs to ensure ICFR compliance, the Center for Audit Quality (CAQ) has provided evidence that the SOx act has a positive impact on the robustness of the US capital market, corporate valuation, and corporate fraud prevention (COSO, 2013). ICFR controls are designed to reduce internal and external business risks in a manner that allows day-to-day operations to be conducted in an effective and efficient manner. Internal control prevents that the objectives set by the company are not achieved. It constitutes a total of measures, guidelines, and controls in terms of IT and organization that should ensure that the reliability of the financial reports is safeguarded. Material errors in financial accounting negatively influence decisions and have an impact on the achievement of strategic and financial objectives.

2.1 IT-Controls

The Committee of Sponsoring Organizations of the Treadway Commission (COSO) provides companies with a framework for internal management and control. COSO is a "top-down and risk-based approach" framework. It has been inserted at the highest level of the Audit Committee, the board of directors, and senior management to create broader support for the urgency, importance, and follow-up. The COSO cube in Fig. 1 depicts the framework.

On the top face of the COSO cube the objectives that a company strives for are displayed, and on the front face the required five components to attain these objectives are shown. The right-hand panel indicates that it applies to all layers of

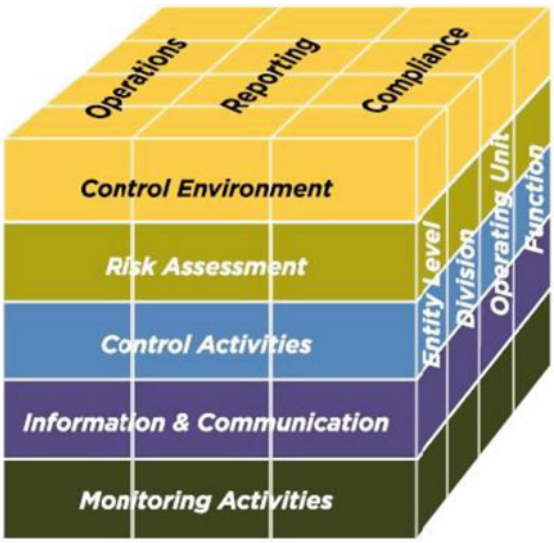


Fig. 1 The COSO cube (COSO, 2013)

the organization. The five components are the foundation for setting up your ICFR and can be set up iteratively and cyclically (COSO, 2013). IT-control objectives aim to ensure that systems can guarantee availability, continuity, and integrity and can be classified into preventive, detective, and corrective IT-controls (Otero, 2018). An example of a preventive control type is the arrangements on access to the IT or setting upper bounds to payment approvals above a certain amount. Detective IT-controls are for example the comparison of extractions from different databases and can be considered as IT-dependent manual controls. Patch upgrades to prevent vulnerabilities are covered with corrective IT-controls.

Within the ICFR, controls take place at all levels of the organization. The organization of IT governance, IT strategy, and IT-business alignment is controlled at the entity level. IT security choices such as software on premises, or in the cloud are decisions made at the entity level. The standard on which these decisions are based is what is more cost effective and more secure in current market conditions. At the process level, controls such as segregation of duties and manual controls are established to detect risks such as conspiracy. At this level, data in applications is checked by means of extractions in order to manually compare the data from independent data sources.

IT-controls can be deployed and set up at all levels and have a different relationship to each other. For example, IT access management at entity level control concerns issues such as access policies and procedures. On a process level, IT access management is about which function or department has decision-making rights in the system. At the ICFR, the IT General Controls (ITGC) and application controls (AC) are the most important IT-controls and are applied to the relevant information systems at application and infrastructure level. Examples of application controls are the establishment of authorization matrices, that define who has which rights, payment approvals, and an automated 3-way match for purchasing.

The ITGCs apply to the processes related to system components such as databases, operating system, and network that are present in an organization. The objective of ITGCs is to control the process in and around the information system. The most used and most relevant ITGCs are (ISACA, 2014):

- Logical access security: Granting access to systems.
- Program change management: Implementing changes on systems.
- IT Security: Securing systems.
- Backup and recovery: Safe data storage and recovery of data.
- System development: Assessing whether systems need to be replaced.
- Computer operations: Configuring and setting up systems.

Control measures, ITGCs, are necessary to prevent IT risks and thus guarantee the continuous operation of the application controls. The deployment of control measures, application controls and IT-dependent controls, in relation to ITGCs are visible in the BCF quadrant (Fig. 2).

The first quadrant includes the segregation of duties that describes the powers within the organization, what you are allowed to do. The plan has been established to segregate duties. In the second quadrant that concerns the IT General Controls,

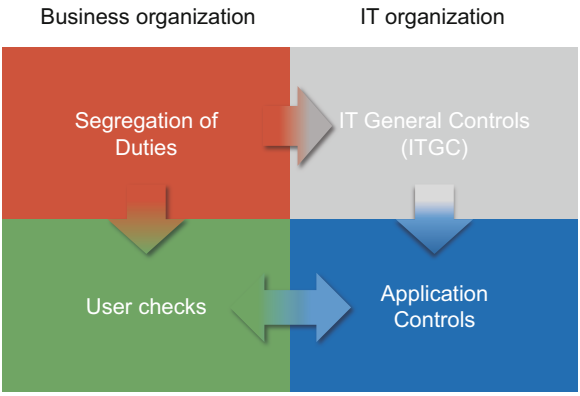


Fig. 2 The business control framework (Folkers & Westra, 2017)

comprehends several processes. Within this quadrant, the operation is tested over a period of time. A third quadrant is related to the Application controls (AC). The AC configuration device of an application allows automatic controls to be done within the application. In contrast to quadrant 1 (what you are allowed to do), in this quadrant it becomes clear what you can actually do. It is a recording of the existence of the control. The final and fourth quadrant concerns the manual controls (MCs). Manual procedures are obviated herein. Between the MC and AC there is an intermediate form "the IT-dependent control." An example of this is extractions from databases that are subsequently checked manually.

The effective design of application controls can be demonstrated by proof of existence. For application controls and IT-dependent controls to work continuously and undisturbed, it is important that the ITGCs have also worked effectively during a control period. The ITGCs are preconditional and test the operation of the controls over a period of time. Without a sufficient level of segregation of duties and ITGCs, it cannot be determined with reasonable certainty whether user controls or application controls have worked during the controlling period.

In the financial external audit, it is possible to deviate from relying on the ITGCs if these prove insufficient during the systems audit. Then firms can choose to use data analysis to determine whether there has been a deviation during that period, to which is referred as the data-oriented checks. If there is no deviation or unfamiliar data patterns in the sampled population of the data, it can be stated that the risk has not materialized during that period and that there is no reason to believe that a material error has occurred in that process.

In the SOx Act Section 404, there is hardly any description of the necessary controls that a company should apply, let alone IT-controls. The COBIT methodology supplements the COSO framework with a set of predefined IT-controls at different organizational levels. Depending on the company and risk analysis, the IT-controls required are implemented within a company's ICFR framework. Below are the minimal IT-controls in an IT environment of a large company from the COBIT framework in relation to the ITGCs (ISACA, 2019) (Table 1).

Table 1 COBIT controls (ISACA, 2019)

ITGC	COBIT controls
Logical access security	**DSS05—Managed Security Services** Protect enterprise information to maintain the level of information security risk acceptable to the enterprise in accordance with the security policy. Establish and maintain information security roles and access privileges. Perform security monitoring.
IT-controls	• Distribute access of privileged accounts for maintenance. Assign the account, its duration, and its revocation. • Monitoring privileged account access through logging of activities (change log) done in the application, database, or server.
System development and life cycle	**BAI03—Managed Solutions Identification and Build** Establish and maintain identified products and services (technology, business processes, and workflows) in line with enterprise requirements covering design, development, procurement/sourcing, and partnering with vendors. Manage configuration, test preparation, testing, requirements management and maintenance of business processes, applications, information/data, infrastructure, and services. **BAI07—Managed IT Change Acceptance and Transitioning** Formally accept and make operational new solutions. Include implementation planning, system and data conversion, acceptance testing, communication, release preparation, promotion to production of new or changed business processes and I&T services, early production support, and post-implementation reviews.
IT-controls	• IT development, Development, Test, Acceptance, Production (OTAP) in separate environments. • Service management, managing the IT solution based on acceptance criteria like usability, availability, security, and continuity.
Change management	**BAI06—Managed IT Changes** Manage all changes in a controlled manner, including standard changes and emergency maintenance relating to business processes, applications, and infrastructure. This includes change standards and procedures, impact assessment, prioritization and authorization, emergency changes, tracking, reporting, closure, and documentation.
IT-controls	• Changes in the production environment on hardware, software, interfaces, or network are registered, documented as implemented. • Urgent changes due to incidents that have a high business impact due to an error or bug in the IT service.
Security	**DSS05—Managed Security Services** Protect enterprise information to maintain the level of information security risk acceptable to the enterprise in accordance with the security policy. Establish and maintain information security roles and access privileges. Perform security monitoring. Minimize the business impact of operational information security vulnerabilities and incidents.
IT-controls	• Security violations are detected by the IT service and reported to IT departments who take appropriate action. • Vulnerabilities are regularly scanned at the different layers and threats are immediately reported to the responsible IT department who takes corrective action.

(continued)

Table 1 (continued)

ITGC	COBIT controls
	• The version of the operating and database management systems are regularly updated with the latest security updates for that version. • Security patches on the software and hardware should be checked and installed and monitored that they are up to date.
Backup and recovery	**DSS04—Managed Continuity** Establish and maintain a plan to enable the business and IT organizations to respond to incidents and quickly adapt to disruptions. This will enable continued operations of critical business processes and required I&T services and maintain availability of resources, assets, and information at a level acceptable to the enterprise.
IT-controls	• Recovery Time Objective and Recovery Point Objectives are configured in the IT architecture. • Periodic testing of backup and recovery scenarios and whether these processes work.
Computer operations	**DSS01—Managed Operations** Coordinate and execute the activities and operational procedures required to deliver internal and outsourced I&T services. Include the execution of predefined standard operating procedures and the required monitoring activities.
IT-controls	• The baseline settings for the infrastructure are checked based on the best practices. • Maintenance on IT application, database, network, or server is monitored with procedures and logging.

The further implementation of the IT-controls is elaborated in work programs based on the objective and the risk. The activities to test the controls are carried out annually. The test work is performed by the IT owner or IT auditor of this program at application and infrastructure level. For the ICFR, this framework covers the risks and the company can issue a letter of representation confirming that it has prepared the financial report in accordance with the applicable reporting frameworks such as IFRS or GAAP (KPMG, 2018).

2.2 Benefits of Blockchain for Accounting

Blockchain-based financial transaction processing affects the financial landscape of the accountant. The benefits created by the use of blockchain technology specifically have the following impact on quality aspects in a financial audit:

- Integrity—The transactions are permanently encrypted in the network after they have been validated. Once this is done, the transactions cannot be changed. The transactions become "immutable" thanks to the validation and encryption.
- Availability—The transactions are replicated and distributed to all participating nodes. These "shared ledgers" are identical in every node and are continuously

updated with new validated transactions. This increases the transparency and availability of transactions between companies, or within a firm.

- Continuity—An individual node that shares a ledger may fail or go down. Thanks to the decentralized nature and replication of data, the nodes in the P2P network can load the failed node with the historical data after activation. Once the node can fully participate again it can simply retrieve all historical data.
- Traceability—Each block has a hash pointer to the previous block. Because all blocks in a chain follow this principle, the entire history can be traced back to the origin of the transaction and makes it easier to audit this data.
- Reliability—The transactions cannot be changed; an audit trial makes all transactions traceable and the origin is unequivocally established. The validation of the transactions together with the above properties makes the network highly resistant to errors or fraud thanks to the consensus mechanism—the "trust protocol."
- Efficient/effective—High volumes of transactions are quickly shared in a blockchain network without the intervention of third parties (disintermediation). Participation with a node is easy thanks to the P2P network configuration that makes the network scalable (Fig. 3).

The shared immutable financial records prevent financial information from being misappropriated, falsified, or destroyed between trading partners or stakeholders. Keeping opaque financial records is more difficult in the blockchain. Using a blockchain agents, subsidiaries, or chain partners can share each other's financial information to be transparent about the transactions they make. Blockchain technology can also influence the following quality dimensions of the International Financial Reporting Standards (IFRS) (Bonsón & Bednárová, 2019; Liu et al., 2019):

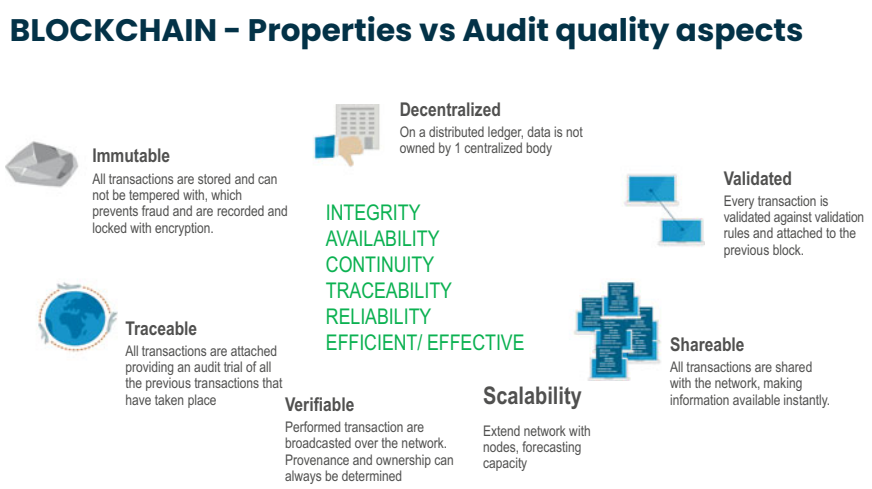


Fig. 3 Properties vs. audit quality aspects

- Accuracy—The financial data is validated with a consensus mechanism or smart contract in the nodes before being added to the blockchain.
- Timeliness—All transactions are continuously provided with an up-to-date timestamp. Also, in the hardware layer of the network, metadata is continuously updated as time.
- Completeness—The data in a node is the complete transaction consisting of a number of required fields. The predefined fields are easy to compare and should be filled in to get the transaction validated on the network (Fuller & Markelevich, 2020).

Blockchain is regarded as a promising technology to increase trust between parties. The advantage for the accountant would be that blockchain technology improves the processing of financial records, increases the detection of material errors and reduces the risk of human error. It would enable the accountants and auditors to save on cost and time in various audit aspects (Seibold & Samman. 2016).

- Evidence—The data integrity and quality are guaranteed and therefore data-oriented control of the entire population can be done easily and quickly. This allows "continuous auditing" for the ICFR to be applied in the blockchain. The time saved ensures that there is leeway for other system-oriented audit tasks regarding the blockchain system, such as detection of fraudulent smart contracts outside the blockchain, off chain.
- Transaction Validation and Verification—The validation and verification of transactions is done by the blockchain consensus mechanism. The use of financial shared service centers (FSSC) to perform checks or make interim bookings is unnecessary, which reduces the chance of errors.
- Reconciliation—The blockchain network provides all parties with the financial records. It makes the reconciliation process more efficient and no FSSC is needed to support the creation of settlement invoices. Paper administration can be reduced thanks to the digital audit trial created in real time.
- Financial Reporting—Due to the fast automated settlement and data encryption of transactions, the blockchain facilitates the efficiency and reliability of financial reporting.
- Compliance—The blockchain smart contracts have been drawn up to digitally record agreements and have them executed if the rules are met. The digital recording makes it easier to check whether the transaction complies with the smart contract.

In contrast to such advantages in doing an audit, the blockchain also has its limitations, e.g., the processing of sensitive financial information that may not be shared publicly, off-chain applications that fraudulently interact with the blockchain, or the lack of blockchain standards complicating "trading partners" to participate. The execution of the tasks of accountants and auditors is changing as a result of this disruptive technology (Seibold & Samman, 2016).

3 Case Study: "Intercompany Settlement Blockchain"

In February 2020, KLM Royal Dutch Airlines launched the financial intercompany settlement (ICS) in a private permissioned blockchain network, in this article referred to as the corporate blockchain. This Proof of Concept (PoC) places the semi-manual process of payments between trading partners in a blockchain. This blockchain records the conditions between trading partners with a smart contract and then books the payment when these conditions are met. This PoC, which is a simplified representation of the ICS process with intercompany bookings without VAT and forecasting, optimizes making payments between trading partners. In this case study, it focuses on a payment between a parent and subsidiary whereby the subsidiary provides services to the parent company. This case study will mainly focus on the IT-controls of the intercompany settlement process and will compare these IT-controls in the "AS-IS" situation with the "TO-BE" situation in a blockchain. The data sent herein are ledger bookings with transaction data and no privacy data.

Based on data from the ERP system, for this type of intercompany payments 230–300 bookings are made per month which are processed by at least 3 employees with an average processing time of approximately 10 min per employee without their control activities.

3.1 AS-IS: "Intercompany Settlement (ICS)"

Intercompany settlements are transactions between two or more related internal legal entities where one company invoices another. Managing these transactions is one of the biggest challenges for finance departments because in many cases, financial processing takes place across different departments and systems. Prior to the financial processing, agreements are made between the various companies about the goods or services to be purchased, the price, the payment, and the administration forms. When there are unclear agreements together with a long lead time before the actual invoicing, the chances of "imbalances" in the transit account and balance differences in the "month-end closing" increase.

Below you can see a schematic representation of the payment process with an intercompany document: the actions that take place with the financial entries recorded in the Enterprise Resource Planning (ERP) application. The actions of this process are manual and consist of: creating intercompany forms, Excel sheet uploads with bookings and entering payments into the ERP. This makes the process slow and labor-intensive with a high probability of errors (Fig. 4).

Subsidiary B provides services to parent company A. The relevant business controller of the department from company A has made agreements with Subsidiary B. The contract contains various details, including which services will be provided at

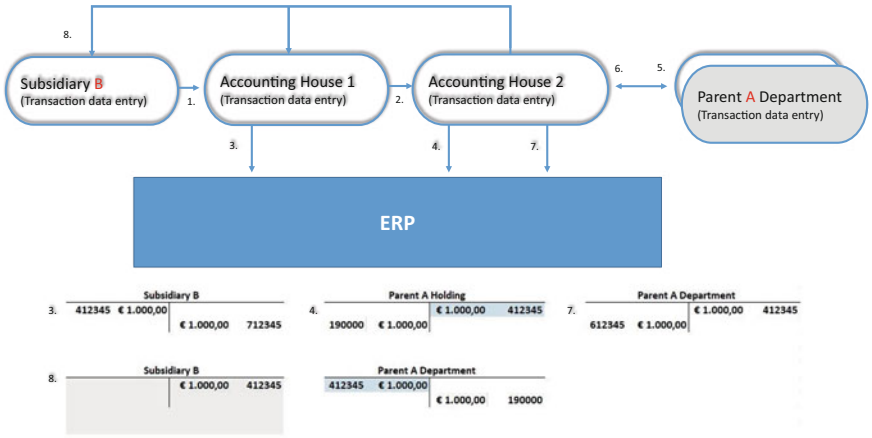


Fig. 4 Intercompany settlement process AS-IS

what price. In the "AS-IS" method, eight steps are required for the settlement of the payment between company A and B, carried out by different persons (Table 2).

In this process there are two Accounting Houses involved and multiple business controllers from different departments of Company A. Each business controller

Table 2 Steps in AS-IS intercompany settlement process

Process steps	"AS-IS intercompany settlement process"
1.	Company B provides a service for Company A and sends an Excel sheet with financial transaction entries to Accounting House 1 during the month.
2.	Accounting House 1 creates separate intercompany documents for the administration on the basis of the Excel sheets and sends them to Accounting House 2.
3.	Accounting House 1 then makes a manual entry in the ERP system to digitally record the payment in the "ledgers."
4.	Accounting House 2 receives the intercompany document and stores it in their administration. They make sure that the right department in Company A is charged. They enter the financial transactions in the ERP with the associated cost centers, amounts, intercompany document number, etc. This results in an intercompany booking.
5.	The Accounting House 2 contacts the business controller of Company A to inform that the administration and registration have been done and to ask which account should be used to book the amount on. This should result in a cleared ledger with the department.
6.	The business controller of Company A emails Accounting House 2 the "ledger" where the amount needs to be posted to.
7.	Accounting House 2 executes the booking to the relevant "ledger" in the ERP system and that ensures that all bookings are zero-balance.
8.	Accounting House 2 emails a confirmation to Subsidiary B 2 days after the month-end closing with the final settlement balances in the ledger of the ERP system.

handles his own domain with specific attributes such as cost centers, general ledger accounts, intercompany document numbers, etc.

3.2 TO BE: "Intercompany Settlement Blockchain"

The blockchain configuration uses a smart contract for validation of transactions in this financial ICS process. In the AS-IS situation, the validation of the transaction is done manually. In the intercompany blockchain, manual tasks are minimal (Fig. 5) (Table 3).

This ICS process starts with the one-time configuration of a smart contract between two trading parties. Company A checks the smart contract of Subsidiary B and accepts it or vice versa (Steps 1–3.) After that, all transactions from Subsidiary B are checked against the smart contract and no longer by the Accounting Houses. A total of three steps are performed once creating a smart contract, after that all entries are made automatically (Steps 4 and 5).

Before elaborating further on the IT-controls, this paragraph starts by elucidating the architecture of this customized corporate blockchain. In this case study, the network configuration is set up with a so-called "notary node," which is the node in the network that stores and activates the smart contracts. Company A and Subsidiary B are present in the network with their own network "node." After validation and execution of the smart contract, only the bookings are made to their ledgers. This is different in a "public permissionless" network such as bitcoin blockchain. In that type of blockchain, everyone receives a copy which is not the case in this corporate blockchain. Node-to-Node is the consensus mechanism that makes this possible and

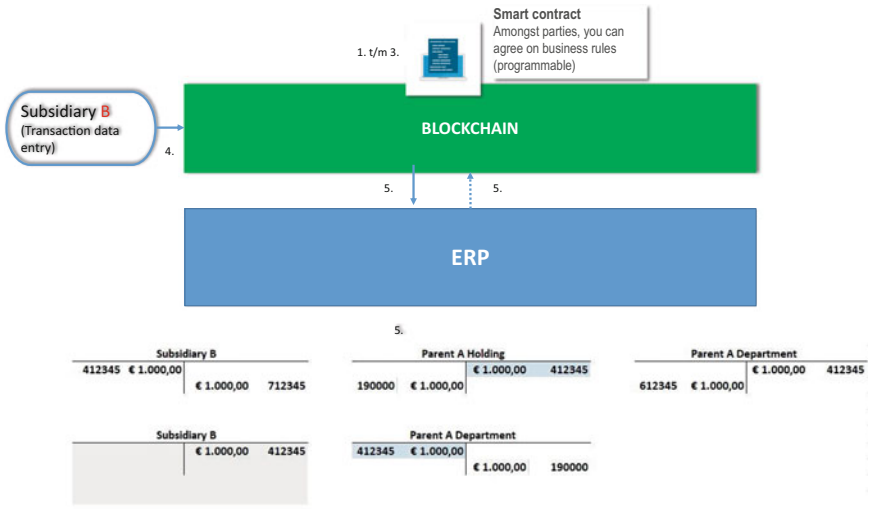


Fig. 5 Intercompany settlement process TO-BE

Table 3 Handling in TO-BE blockchain-based intercompany settlement process

Step	Handling "TO-BE intercompany settlement process" BLOCKCHAIN
One-time action setting up a smart contract (3 steps)	
1.	Company A and Subsidiary B periodically make agreements about the services to be provided. They make agreements about the type of services, the amounts and to which general ledgers (GL) accounts the payments need to be booked.
2.	In this phase, one of the companies (A) creates up a smart contract proposal. It contains the attributes needed to make the bookings to the right cost centers, GL account, type of activity carried out, intercompany document number, etc.
3.	This smart contract is stored and the other company (B) gets to see its own blockchain web application and can approve this proposal by accepting it. The smart contract has been activated and is ready to make the payments when the conditions are met.
Enter continuous transaction (1 step)	
4.	An employee of the subsidiary B makes a payment. He logs into a transaction screen in the blockchain and enters the type of service and the amount and presses submit.
5.	The transaction is validated against the smart contract. When it does not meet the conditions, the user will receive a notification that the payment cannot be made. When it does meet the conditions, all 8 bookings as in the "AS-IS" situation take place at once in real time. The ERP sends back an acknowledgement with the help of a feedback loop that the bookings have been made.

is a bilateral consensus mechanism between the nodes. It does not need an energy-consuming intermediary as miners or stakes for consensus. This makes it possible for this blockchain network to send the payments to the parties involved as agreed in the smart contract. Each smart contract can be drawn up by parties other than A and B. Companies B and D can make mutual agreements and include this in their smart contract. In this case, only B and D get the bookings but with the same attributes as cost center, GL accounts, type of activity, intercompany form number, etc. A new company can sign up and have a node configured and make agreements as a "trading partner" with all other nodes, "trading partners," in the blockchain network. This makes this blockchain scalable (Seibold & Samman, 2016) (Fig. 6).

One of the risks with blockchain applications is the integration with off-chain applications. The challenge is that both technologies function differently and are configured different. This is solved by an ERP node that ensures that all bookings are sent to the ERP system at once in the correct format. The ERP node facilitates the "transit" bookings to the Parent A Holding and Parent A Department (account 190000) that are necessary for processing the transactions in the ERP system. This guarantees the integration with backend ERP systems. This node and the "notary node" are the technical nodes that take care of the validation and handling of the transactions. The figure below (Fig. 7) shows a snapshot of the financial transactional data interfaced to the ERP backend system.

The total booking is interfaced in real time and the transactions are grouped. These transaction entries have been validated against the smart contract #23569 in the notary node and enriched with the intercompany transit account 190000 in the ERP node. The end state forms a zero-balanced account for the involved nodes: notary, ERP, A and B, each having this total booking recorded in their node.

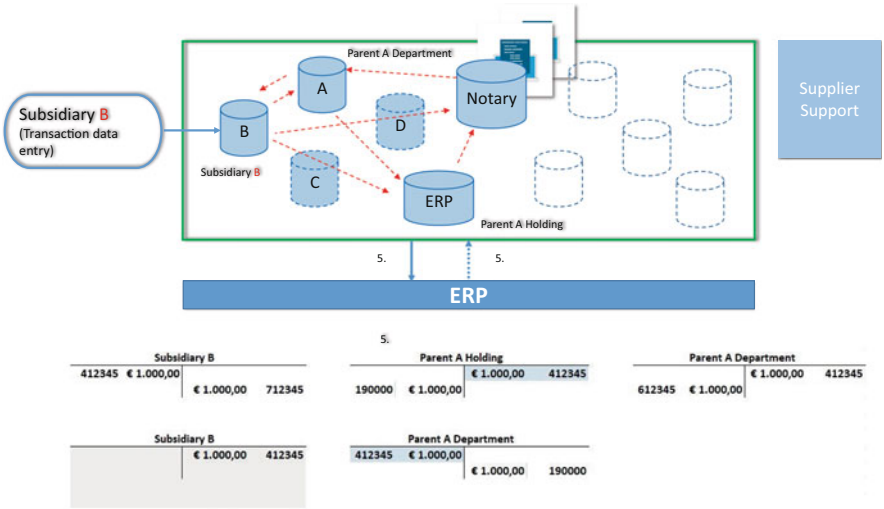


Fig. 6 Corporate blockchain node architecture

ERP_node_interface_23569 - Kladblok

Bestand Bewerken Opmaak Beeld Help

Date	Time	Debet/ Credit	Account number	Amount	Trading partners	Description	Smart contract
15-1-2020	14:02:54	D	412345	1000	Subsidiary B	ICS Blockchain	#23569
15-1-2020	14:02:54	C	712345	1000	Subsidiary B	ICS Blockchain	#23569
15-1-2020	14:02:54	D	190000	1000	Parent A Holding	ICS Blockchain	#23569
15-1-2020	14:02:54	C	412345	1000	Parent A Holding	ICS Blockchain	#23569
15-1-2020	14:02:54	D	412345	1000	Parent A Department	ICS Blockchain	#23569
15-1-2020	14:02:54	C	190000	1000	Parent A Department	ICS Blockchain	#23569
15-1-2020	14:02:54	D	612345	1000	Parent A Department	ICS Blockchain	#23569
15-1-2020	14:02:54	C	412345	1000	Parent A Department	ICS Blockchain	#23569

Fig. 7 Output flat file from the interface between ERP node and ERP backend

3.3 IT-Controls: "AS-IS Control Environment"

This case study includes several stakeholders. Together they form the "control environment." The risks that hinder the achievement of the goal are mitigated with controls. In this case study, the three categories are:

1. *Entity Level Controls (ELC)*

   Entity level controls are the controls on the basic structures on which the organization rests. This varies from risk matrices to management guidelines. In this case study, entity level control is the entire chain of companies and departments that do business with each other. In this example, the relationship between company A and company B is a parent-subsidiary relationship. The control environment of the "period-end closing cycle" is carried out according to the same accounting principles and procedures for both companies. In this case study, we only name the IT-related ELCs and their relationship to ITGCs.

2. *Transaction Level Controls (TLC)*
   This control, "Business-IT," focuses mainly on the transactions that take place within the ERP. It also looks at the interfaces between systems and the interactions between departments. Within this process, financial employees carry out transaction entries in the ERP system. To ensure the transactions, the following "controls" can be distinguished:

   (a) Manual controls
   For example, check whether the transaction input is correct with the intercompany document.
   (b) IT-dependent manual controls
   For example, comparing extracts from ERP tables with financial data with each other, checking whether the suspense accounts are "zero-balanced."
   (c) IT application controls
   For example, restriction on booking on certain balance sheet items or computerized entries.

3. *IT general controls (ITGC)*
   All IT aspects that have an impact on IT processes and IT technical matters are safeguarded by ITGCs. The processing in the IT landscape is thus controlled. They can be divided into manual controls, emptying flooded server buffers, and application controls, restrictions when logging in. It concerns IT-related matters such as IT processes and IT technology in operating systems and databases and not directly about the end-user interaction as with transaction level controls.

These TLCs are included in a work program and this control is carried out by a control employee checking the clearing of the transit accounts within the ERP application. The entity level controls are relatively general and generic compared to the transaction level controls. The ITGC for financial reporting has been elaborated in detailed control documents and is derived from COBIT 5-"IT-control Objectives for Sarbanes-Oxley" of ISACA (2014). In this framework, the risks, control objective, the owners, the audit trail/evidence and which tests need to be executed are documented in control matrices. In this case study, the focus will be on IT-controls.

3.4 Objective and Risks: "Top-Down"

The goal of this intercompany settlement process is to efficiently and effectively process financial transactions and ensure the reliability of financial reports. The risks of an intercompany process are present at different control levels. The different entities may have their own vision on ICFR control tasks and their implementation. At the process level, the employees may be tempted or instructed to book away the

differences on the transit balance sheets. Detecting unwanted actions takes time because of the manual and IT-dependent checks. The IT-control objectives are present on all control levels and apply to the central financial ERP system and its surroundings and are described in Table 4 below.

Table 4 Risks and control objectives AS-IS

Entity Level		
Risk	Control objective	
The entities are not compliant with accounting standards.	Use of international accounting principles	
The entities create insecure gateways to the central ERP system.	Central policy and standards for access to financial application, ERP (Logical Access Security).	
Transaction level		
Risk	Control objective	
Errors in transaction posting causing balance differences on transit accounts.	Regular checks on incomplete transaction entries and zero-balance on transit accounts.	
IT General Controls (ITGC)		
Risk	Control objective	ITGC
R1. Unauthorized access or data breaches affect business data on confidentiality, integrity, and availability	The information in the systems and related systems are designed to prevent unauthorized use, exclusion, and damage or loss of data.	Logical access security
R2. Suppliers do not facilitate sufficient certainty on the confidentiality, availability, or integrity of the information systems and their data.	Suppliers must comply with laws, regulations, and certifications that contractually ensure availability and integrity.	System development and life cycle
R3. Unvalidated and impactful system changes on the production infrastructure affect the integrity of the business data in the systems.	The system changes are authorized and properly tested before being moved to production.	Change management
R4. The confidentiality, integrity, and availability of business data have been compromised by unauthorized access to systems, security breaches, or data leaks.	The information systems and related systems are sufficiently protected against unauthorized access and against providing, modifying, damaging, or losing data.	Security
R5. Damage or loss of business data that can no longer be repaired.	That the recorded, processed, and reported data remains complete, accurate, and valid during the recovery and during the update and storage process.	Backup and recovery
R6. The confidentiality or integrity of business data is affected by insufficient control over batch transaction processing or by changes in software and hardware configuration items.	Job scheduling mechanisms are set up correctly and changes in the batch processing are identified and investigated. IT security configuration items are properly secured and protected against unauthorized changes.	Computer operations

3.5 Objective and Risks: "ICS Blockchain"

The purpose of the intercompany settlement process in a corporate blockchain remains unchanged; the real-time transaction processing and encryption contribute to an improvement of the efficient and effective processing of transaction data and the integrity of the financial data. Transparency has increased thanks to "shared ledgers." In the current situation, the financial ERP is leading in financial accountability. With differences in the blockchain "shared ledgers," the risk is that an entity can claim to have the financial "single source of truth." The smart contract can offer a solution in case of a difference, but parties can create and delete fraudulent smart contracts. The changing risks have an impact on the IT-control objectives and are described in Table 5 below.

Table 5 Risks and control objectives ICS blockchain

Entity Level		
Risk	Control objective	
The entities, full nodes, receive incorrect or no information from partner nodes.	– Securing segregation of duties of the nodes. – Centrally coordinated audit work for the ITGCs for all nodes.	
Transaction level		
Risk	Control objective	
Drafting and executing fraudulent smart contracts and collusion.	More eyes principle when drawing up a smart contract. Table 3 points 1–3	
IT General Controls (ITGC)		
Risk	Control objective	ITGC
R1. User access (authorization) and devices (API) to smart contracts and key pairs can indirectly compromise business data on confidentiality, integrity.	– Securing access to the critical components of the blockchain in such a way that unauthorized access is impossible and to prevent unauthorized permission to validate or broadcast transactions. – Provisioning of administrator nodes.	Logical access security
R2. Suppliers do not facilitate sufficient assurance of their systems and knowledge of blockchain implementations – Not being able to connect with other blockchains. – Upgrading the network is too complex to perform.	– Suppliers must comply with laws, regulations, and certifications and have suitable blockchain knowledge and skills in-house. – The blockchain can easily interact with off-chain applications and other blockchains, interoperability. – Implemented Life Cycle Management with a hard fork procedure.	System development and life cycle
R3. Changes in a blockchain that have a negative impact on partner nodes and other critical network components.	– Changes are centrally authorized and properly tested in consultation with other blockchain participants. – Changes to the data are "append only."	Change management

(continued)

Table 5 (continued)

R4. The confidentiality and integrity of business data have been compromised by theft of keypairs, code modification in smart contracts, hijacking of a node and making "malicious" changes to the consensus mechanism.	– The blockchain, various nodes and related systems are sufficiently secured against unauthorized access and against stealing, modifying or damaging smart contracts, keypairs, consensus mechanisms, or hijacking of nodes. – The blockchain has detection logging especially on the vital network components, e.g., notary node.	Security
R5. Corruption of nodes and business data that can no longer be repaired.	The entire business data can be quickly replicated from the partner nodes that are unaffected.	Backup and recovery
R6. The efficiency and effectiveness of transaction processing are hampered by changes in software and hardware configuration items. The scalability cannot be achieved.	– Secure oracles with off-chain applications and other blockchains or smart contracts. – Node scale-up procedures/standards. – Instant network communication of central and decentralized network components.	Computer operations

Table 6 AS-IS entity level controls

Entity	Entity level control (Total 13)
Company B (subsidiary)	Ownership of financial data Accounting principles Access security (IT related)
Accounting House 1	Accounting principles Access security (IT related)
Accounting House 2	Accounting principles Access security (IT related)
Company A (Department of holding)	Ownership of financial data Accounting principles Access security (IT related)
	Evidence e.g.: • IFRS16, Access policies, etc.
IT business unit	Disaster recovery plan IT Governance (manage IT configuration) Access security
	Evidence e.g.: • Design documents backup and recovery/access policies • Tests performed on backup and recovery/access tests • IT configuration management policy, standards

The risks described above assumes that the nodes run on decentralized data centers. As can be observed in the entity level of Tables 6 and 7, the ITGC control in the blockchain takes place at a "corporate level" instead of only application or system level. A blockchain is a network and therefore demands a need of

Table 7 TO-BE entity level controls

Entity	Entity level control (total 15)	New ELC
Subsidiary B (Owner full node B)	Ownership of financial data Accounting principles Access security	Consortium meetings: – IT Governance (all ITGCs) – Access security (IT related)
Accounting House 1	*Obsolete*	*Disintermediation*
Accounting House 2	*Obsolete*	*Disintermediation*
Company A (Owner full node A)	Ownership of financial data Accounting principles Access security	Consortium meetings: – IT Governance (all ITGCs) – Access security (IT related)
(Owner full node C)	*Under construction*	*IT Governance-chain deliberation*
(Owner full node D)	*Under construction*	*IT Governance-chain deliberation*
IT business unit: Owner "Notary node" "ERP node" and data center	Disaster recovery plan IT Governance (Manage IT configuration) Access security	Consortium meetings: – IT Governance (all ITGCs) – Access security (IT related)
		Evidence: IT Meeting structures

collaboration between blockchain parties when they decide to deploy changes on the network. In a consortium with blockchain participants, agreements will be made about these vital network components, which can impact the integrity and confidentiality of the blockchain network. The "AS-IS and TO-BE" Intercompany settlement objectives, efficient and effective processing of transaction data and the integrity of the financial data, remain the same in both situations. The risks and control objectives change. The existing risks are identified to secure a central ERP application. The new risks are based on a blockchain network and are having direct impact on the IT-controls on all levels.

3.6 Entity Level Controls: "Corporate Level"

Within the business units the ELCs are translated into for example "segregation of duties" in which the control activities are separated from the payment activities. In this case study, there are five legal entities: companies A and B, accounting houses 1 and 2, and the (IT) division where the applications are managed. Each entity organizes separately in its organization the control environment, risk assessment, control activities, and monitoring. These are periodically controlled by the control department in the Holding (Table 6).

In this intercompany settlement process, companies A and B are the owners of the financial data but have separate administration. The accounting standards are the mutually agreed international standards. The IT landscape is centrally set up and all entities use the same ERP system, single source of truth. The individual entities are

linked to the IT business unit with the central ERP system. The ELC-IT Governance is mainly done within this unit. Without IT Governance, there is a risk that each entity will create its own access to the central ERP system, which can lead to insecure accesses. Within the entities, they ensure that the entities have met the access standards before granting them access to the central data center where the ERP system is located. The IT Business unit is responsible for the entire ERP IT configuration, they organize this with an internal IT-controls framework.

3.7 Blockchain and ELC: "IT-Control Environment"

In this case study the entities are a virtual representation, a full node. A full node has its own operating system, database, and network configuration. Each node has a specific task, which creates a virtual segregation of duties. The notary node validates the transaction, the ERP node books it in the off-chain application, and the other nodes distribute the bookings among themselves. Every entity in the blockchain network owns the distributed financial data. In the corporate blockchain, the emphasis of control at ELC level is in the "IT-control environment" (See green area Figs. 5 and 6) and less in the separate entities. The technical integration of the entities results in this IT-control environment in which control activities and monitoring are needed to mitigate the new risks (Table 7).

The disintermediation property of the blockchain makes the accounting houses obsolete in the ICS process. The remaining entities will need to discuss common IT topics such as: encryption, consensus protocols, smart contracts, and access security in a consortium meeting which will be IT-related ELCs. The entities are now part of and have access to a blockchain network. This makes all IT General Controls such as manage IT configuration and Access security, ELC topics that need to be discussed on an entity level.

3.8 Transaction Level Controls: "IT-Dependent Controls"

The audit work is carried out in the financial department by an internal control employee using a control work program. It defines some 25 audit topics, ranging from negative credit account entries to IFRS16 special accounts. The TLC that are applicable for this case study is the control of the "transit/in between" balance items. This is to prevent an intercompany invoice from stranding on a transit account or landing on a wrong account. The total number of trading partners with a similar intercompany invoice are in total 31. In this case study we only discuss the audit procedures of 1 "trading partner" Subsidiary B (Fig. 8).

Due to the many manual actions in this ICS process, the risk that mistakes are made can be high. The checks are done based on extracts from the ERP system, IT-dependent control, with the support of self-created Excel sheet macros. These are

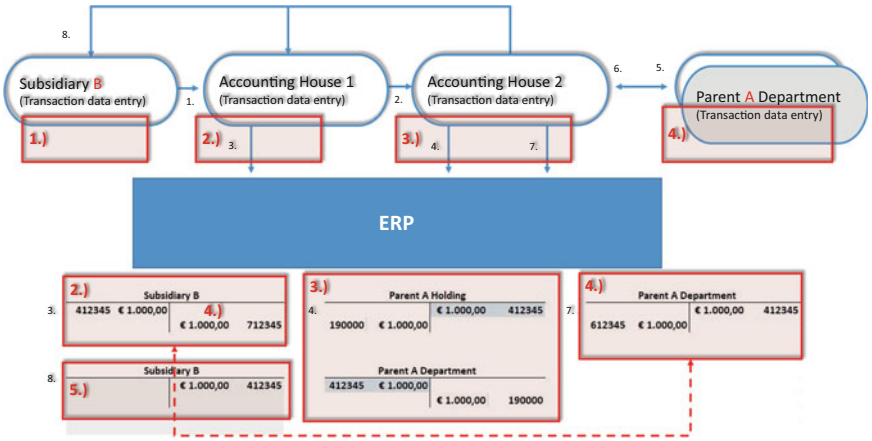


Fig. 8 Intercompany settlement control steps AS-IS

done on top of the user manual checks that take place daily and monthly. This check is done based on attributes: Trading partner, balance sheet item, date, intercompany document number, and amount. A total of five control steps are performed by eight persons. At process level there are seven types of TLC controls. The "evidence" can differ from downloads from databases, comparisons in excel sheets and paper administration (Table 8).

Table 8 AS-IS transaction level controls

Control steps	Transaction (8 pers.)	Controls	Type of control (total 7)
1.	Subsidiary B 2 persons	Input check: 4 eyes principle	Manual control at input
		Control on transaction booking: Intercompany account is zero-balanced	IT-dependent manual monthly
2.	Accounting House 1 1 person	Control on transaction booking: Employee checks the Excel sheet of Subsidiary B for the correct account.	Manual control at input
3.	Accounting House 2 For Company A 1 person	Intercompany zero-balance check: Transit account 412345 check, intercompany document number and amount for Parent A Holding and Parent A department are in balance.	IT-dependent manual, daily
	1 person	Transit account check: 190000 between Parent A Holding and Parent A department are zero-balanced	Application control Month-end close check day 2

(continued)

Table 8 (continued)

Control steps	Transaction (8 pers.)	Controls	Type of control (total 7)
4.	Company A Subsidiary B 2 persons	Revenue and cost accounts check: Company A and Subsidiary B control revenue and cost accounts between Subsidiary B and Parent A department respectively 712345 and 612345	IT-dependent control daily
5.	1 person	Intercompany balance check: Check between Company A and Subsidiary B—this will be the opening balance for the coming month.	IT-dependent control month-end close check day 3

3.9 Blockchain and Transaction Level Controls: "IT-Dependent Controls"

The manual controls in the intercompany process become application controls in the corporate blockchain. The entire intercompany ICS process is carried out automatically in a blockchain (Table 9).

Table 9 TO-BE transaction level controls

	Transaction	Controls	Type control (old)	Type control (new)
1	Drawing up a Smart contract (person A)	Setup digital agreements	Registration Compliance	Automated
	Accept Smart contract (person B)	Accepting the digital agreements	Acceptance Compliance	Automated
2	Subsidiary B 2 persons	Input check: 4 eyes principle	Manual control at input	Automated
		Control on transaction booking: Intercompany account is zero-balanced	IT-dependent manual monthly	Automated
	Accounting House 1 (Obsolete) 1 person	*Control on transaction booking: Employee checks the Excel sheet of Subsidiary B for the correct account.*	*Manual control at input*	Automated
	Accounting House 2 (Obsolete) For Company A 1 person	*Intercompany zero-balance check: Transit account 412345 check, intercompany document number and amount for Parent A Holding and Parent A department are in balance.*	*IT-dependent manual, daily.*	Automated

(continued)

Table 9 (continued)

	Transaction	Controls	Type control (old)	Type control (new)
	1 person	Transit account check: 190000 between Parent A Holding and Parent A department are zero-balanced	Application control Month-end close check day 2	Automated
	Company A Company B 2 persons	Revenue and Cost accounts check: Company A and Subsidiary B control revenue and cost accounts between Subsidiary B and Parent A department respectively 712345 and 612345	IT-dependent control daily	Automated
	1 person	Intercompany balance check: Check between Company A and Subsidiary B—this will be the opening balance for the coming month.	IT-dependent control month-end close check day 3	Automated

At the process level, the impact is large and the blockchain contributes to efficient and effective processing of Intercompany settlement invoices. All manual and IT-dependent controls are overtaken by a total of nine application controls. The transactions are verified and validated against the smart contract in which the agreements between a "buyer" and "seller" are digitally recorded. This digital process optimization eliminates the involvement of accounting houses in the ICS process and their controls are totally automated.

3.10 IT General Controls: "IT in Control"

The IT-controls framework can be traced back to some 23 IT-controls that relate to this case study where the entire spectrum of ITGCs is covered. The ITGCs apply to the ERP application. The goal is to guarantee the Confidentiality, Integrity, Availability of this financial application and this is done along two axes: by sufficiently controlling IT processes, applications, and infrastructure and by continuing to comply with internal and external laws and regulations for the purpose of ICFR. Below are the ITGCs further elaborated with IT-control activities and evidence with a reference to the risk in Table 10.

Table 10 AS-IS IT General controls (for associated risks R, refer to Table 4)

ITGC	Description IT-controls (Total 23)
Logical access security (ELC) R1. Total controls are 4	1. All facets that have to do with login authentication: user ID, session time, passwords are stored and encrypted, password length and renewal, number of login attempts, etc. 2. Access of privileged accounts for maintenance. Granting the account, its duration and its revocation. 3. Having local admin rights to install things on your PC or server. 4. Monitoring access of privileged account by logging of activities (change log) done in the application, database, or server.
Evidence e.g.,	Monitoring privileged accounts of joiners, leavers, and movers Login configuration files—user profiles for logical access management.
System development and life cycle R2. Total controls are 3	1. Purchase of IT hardware and software, cloud applications, and IT contract management. Issues such as security, certification, and financial maturity of the supplier are considered. 2. IT development street, Development, Test, Acceptance, Production (OTAP) in separate environments. 3. Service management, taking care of the IT solution, acceptance criteria. usability, availability, security, and continuity.
Evidence e.g.,	Relevant certifications, audit reports, and validity period.
Change management R3. Total controls are 2	1. Changes in the production environment on hardware, software, interfaces, or network are recorded, documented as executed. 2. Urgent changes as a result of incidents that have a high business impact due to an error or bug in the IT service.
Evidence e.g.,	Registration records and tracking of changes.
Security R4. Total controls are 7	1. Security violations are detected in the IT service and reported to IT departments that ensure appropriate action. 2. Up-to-date anti-virus software. 3. Vulnerabilities are regularly scanned for the different layers and threats are immediately reported to the responsible IT department who take corrective action. 4. The version of the operating and database management systems are regularly updated with the latest security updates for that version. 5. Known security vulnerabilities should be checked and appropriate action taken. 6. High impact security updates are registered, evaluated, and tested before they are installed on the IT services in a planned service window. 7. Security patches on the software and hardware should be checked and installed and monitored to be up to date.
Evidence e.g.,	Vulnerability scan and reports, security update registration, event logs, update information
Backup and recovery R5. Total controls are 3	1. Backup of software and data are regularly backed up. The critical database information is roll back enabled. 2. Recovery Time Objective and Recovery Point Objectives are configured in the architecture. 3. Periodic test of the Backup and recovery scenarios and process works.

(continued)

Table 10 (continued)

ITGC	Description IT-controls (Total 23)
Evidence e.g.,	Scenario test reports, RTO and RPO configuration items, logging enabled.
Computer operations R6. Total controls are 4	1. Access to job scheduling tooling and applications 2. The baseline settings for the infrastructure are checked based on the best practices of the hardware and checked after corrections to the software. 3. Data base management system (DBMS) and monitoring. Servicing the database, e.g., cleaning repositories or analyzing system loggings. This is done under a privileged account, actions are logged. 4. Maintenance on the IT application, database, network, or server are monitored with procedures and logging.
Evidence e.g.,	Database logs, change logs, batch processing logs

3.11 Blockchain and IT General Controls: "IT-Controls"

The full nodes with financial data in the network form an IT-control environment that requires more ITGC coordination between parties. For example, change management will have an impact on multiple nodes in the network and affect multiple trading partners. A centrally coordinated change management in this IT-control environment is needed to control the impact on partner nodes. The specific blockchain items will be added to the IT-control framework to mitigate the new blockchain specific risks. (See risks R1, R2, etc. in Table 11.) In the table below, the change of the controls are described with a reference to Table 10. (See also Appendix TO-BE framework)

Table 11 TO-BE IT General controls (for associated risks R, refer to Table 5)

ITGC	Description-IT-controls BLOCKCHAIN (Total 36)
Logical access security (consortium) R1. New IT-controls. Total controls are 8	The existing controls (1t/m 4 Table 10) continue to apply. The emphasis will be on mutual agreements and cooperation of participants in the blockchain when it comes to IT access and permission management.
	1. Consortium with administrator nodes that control access to the blockchain. 2. Monitoring on API connectors with the blockchain. 3. Setting up permission management to check transactions: "validate or write" and who can "broadcast" 4. Access to critical components such as smart contract, key pairs, and consensus protocols.
Evidence e.g.,	Know Your Customer (KYC) procedures when granting access to the blockchain. Traffic logs of administrator nodes and API connectors.
System development and life cycle (consortium)	The existing controls (1–3 Table 10) continue to apply. The cooperation aspects of joint IT topics require mutual agreements about standards and the recruitment of skilled suppliers.

(continued)

Table 11 (continued)

ITGC	Description-IT-controls BLOCKCHAIN (Total 36)
R2. New IT-controls Total controls are 6	1. Consortium on mastery of source code, encryption, data storage capacity, interoperability, and supplier selection 2. Setting up data migration strategy—replication from the old to the new nodes. (See also backup and recovery) 3. Life Cycle Management strategy with a (hard) fork upgrade procedure.
Evidence e.g.,	Provisioning of replication mechanisms, version control, fork guidelines, and procedures
Change management (consortium) R3. New IT-controls Total controls are 5	The existing controls (1 and 2 Table 10) shall continue to apply. The changes relate to the network and will be centrally regulated and monitored.
	1. Impact analyses on critical components such as smart contract, encryption, and consensus mechanisms. 2. Data changes or smart contract changes "append-only" functionality only correction bookings allowed no change of transaction data. 3. Monitoring for changes in off-chain applications.
Evidence e.g.,	Impact analyses, config item "append-only," off-chain application listing and change cycles.
Security (ELC) R4. New IT-controls Total controls are 10	The existing controls (1–7 Table 10) continue to apply and cover the entire blockchain network. The security consortium will ensure that the security and protection of the entire network remains controlled.
	Security consortium 1. Node governance and security 2. Key ownership and management 3. Protection of critical digital assets: smart contracts and consensus mechanisms.
Evidence e.g.,	Security standards and configuration implementation on nodes. Public key Infrastructure documentation and procedures Monitoring on access paths to critical digital blockchain assets
Backup and recovery (ELC) R5. New IT-controls Total controls are 2	The existing controls (1–3 Table 10) are immediately affected and are cancelled due to the redundancy of data in a blockchain network. The ITGC's system development and security, node governance, absorb part of this ITGC. System development when it comes to the data migration and node governance location of the nodes that are not affected by a "disaster."
	1. Consortium backup and recovery procedures. 2. Speed of reactivation of a node and the synchronization of data to affected partner node.
Evidence e.g.,	Node governance and data replication mechanism
Computer operations R6. New IT-controls Total controls are 5	The existing control (Table 10) 1 is cancelled, 2 is overcome in ITGC change management. Controls 3 and 4 will continue to exist but will be arranged centrally because the work has an impact on the entire network.
	1. Consensus mechanism maintenance. 2. Integration and interfacing of data with off-chain applications 3. Control of node scalability
Evidence e.g.,	Logging by consensus mechanism, interface descriptions, scale-up procedures

This case study demonstrates that the number of IT-controls within the ITGC are changing. In addition it shows, the need of a consortium discussing the network components centrally. Both changes are a consequence of the changing risk profile as described in Sects. 3.4 and 3.5.

4 Analysis: "IT-Control Change"

The blockchain is a new kind of IT network that can exchange value in addition to information. The semi-manual intercompany settlement (ICS) process is automated by employing the blockchain. This process and its control environment has been transformed into an IT environment. On the blockchain, financial business data is encrypted, validated and distributed. Risks of material errors in the blockchain lie in the mechanisms used such as smart contracts and the key pairs that provide the validation and security of the business data. An important implication of using blockchain is that the immediate risk of errors and fraud on business data has shifted to manipulation of these blockchain components which controls the business data. The new IT-control objectives therefore change with the focus on the vital blockchain components as described in the case study. In this analysis, we will further elaborate on the case study in activities performed on a process level and on an IT level.

4.1 Process Level Controls

The intercompany settlement process is optimized by the corporate blockchain for one ICS process with one subsidiary in this case study. Due to the disintermediation of the accounting houses, there are fewer manual invoicing and controls. The optimization can be seen in the control steps and in the transaction level controls (see Figs. 9 and 10).

The total number of process steps has decreased from eight in the AS-IS situation, to four in the new TO-BE situation. The one-time drafting of the smart contract consists of three steps and leads to the fact that the continuous steps, or daily back-office work, are reduced from eight to one step. The trading partner regularly executes the trade(s) and after validation with the smart contract, all bookings are automatically settled in "real time." As such less manual administrative activities are required, resulting in a faster throughput of the transactions from the seller and buyer.The control steps in the ICS process have decreased and are carried out by fewer persons. In the new TO-BE situation the only control step is the drafting and digital acceptance, after verification, of the smart contract by the "trading partners." Labor-intensive controls that take place in the ICS process are reduced to one thanks to the corporate blockchain. This results in lesser control administration pressure on the ICS process.

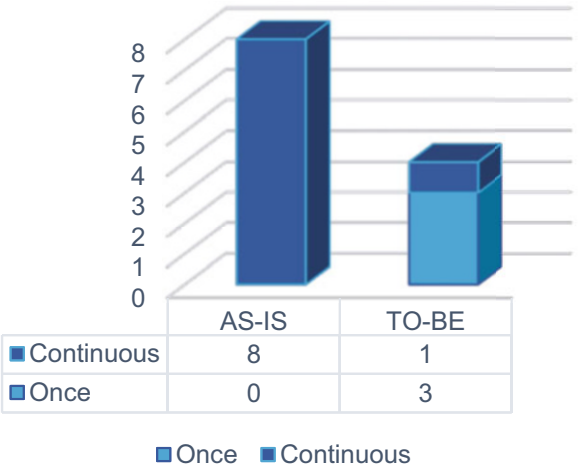


Fig. 9 Intercompany process steps

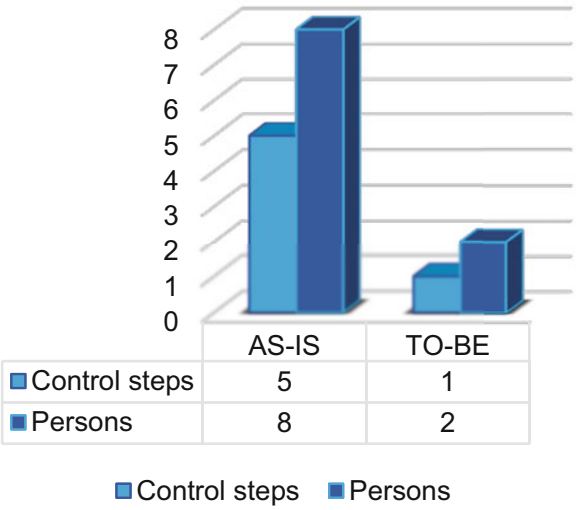


Fig. 10 Intercompany control steps

In more detail in Fig. 11, we see that the AS-IS control process consists of a total of seven controls across different departments ranging from manual, IT-dependent, and application controls. These controls are carried out daily or monthly by the various accounting houses.

In the TO-BE blockchain situation (depicted in Fig. 12), all transaction controls in this ICS process become application controls. The total, nine application controls, consist of the seven "automated" AS-IS controls, plus the two controls for drawing up and accepting the smart contract. Due to the effective and efficient reconciliation, the manual and IT-dependent controls from the internal control work program became obsolete and all entries are zero-balanced. In total there are about 31 similar

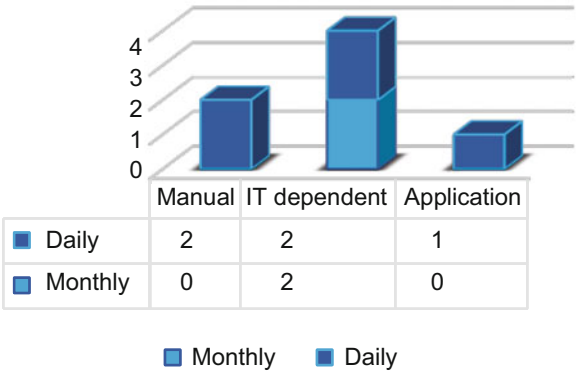


Fig. 11 AS-IS transaction level controls

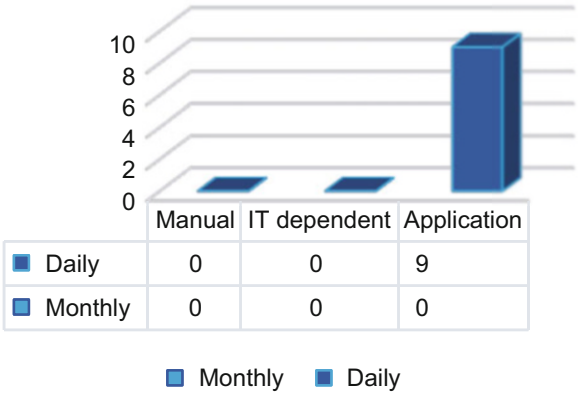


Fig. 12 TO-BE transaction level controls

intercompany settlement flows. The manual and IT-dependent transit account checks of the 31 ICS flows are shifting to application controls thanks to the corporate blockchain. The internal control at process level of transaction level controls are improved by reducing manual controls, facilitating fast reconciliation and creating real-time reporting.

4.2 IT Level Controls

Participants in the blockchain network own a full node with the shared financial data, which adds extra complexity to the control. The emphasis will be on collaboration

and central coordination when it comes to IT-control to mitigate the internal, external as well as blockchain specific vulnerabilities across the entire network. Figure 13 depicts a comparison between the entity level controls in the old AS-IS situation and the new TO-BE situation.

The increase in ELCs despite the disappearance of the accounting houses can be explained by IT consortium consultations. The purpose of these consortia is to keep control over the private blockchain network. The common configuration items and multiple participants make it necessary to inform each other about their technical status of their blockchain node. With regard to the number of ITGCs, there is no difference with or without a corporate blockchain (see Fig. 14). Only the IT-controls, at ITGC level, are increasing in numbers and are an addition to the ITGC (see Fig. 15). The ITGCs also require control over the entire network in a consortium on corporate ELC level, because the financial data is not only secured and stored in an "on premise" application, database, and server stack but can be operating on separate

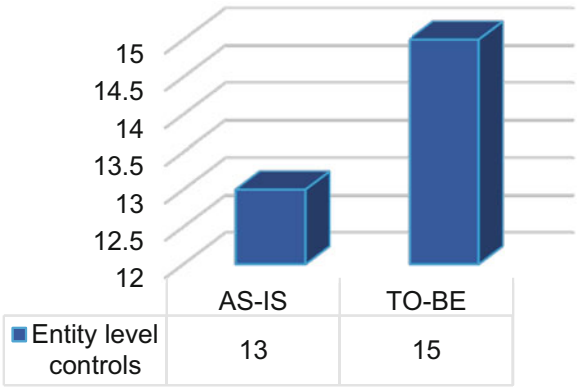


Fig. 13 IT entity level controls

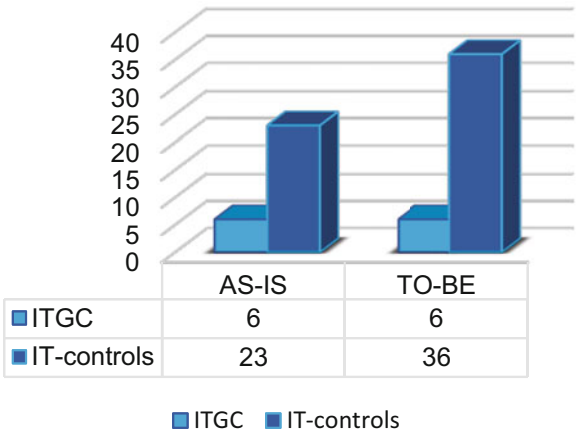
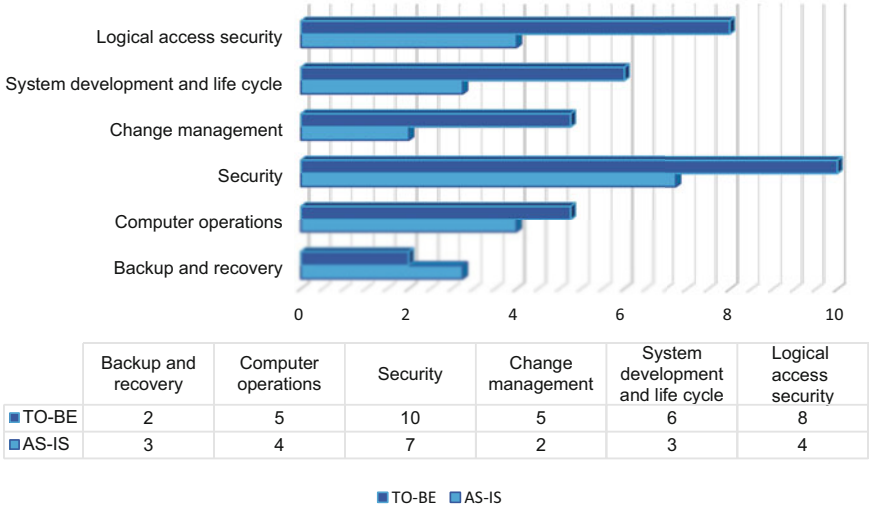


Fig. 14 IT General Controls (ITGC)



	Backup and recovery	Computer operations	Security	Change management	System development and life cycle	Logical access security
TO-BE	2	5	10	5	6	8
AS-IS	3	4	7	2	3	4

Fig. 15 IT-controls per ITGC

data centers making the financial data decentralized. The elaboration per ITGC on IT-controls is shown in Fig. 15. Most ITGCs have an increase on IT-controls. Only Backup and Recovery and computer Operations show a decrease or limited change in their IT-controls. The decrease in Backup and Recovery can be explained by the fact that the notary node and the ERP node are each other's backup. From the ledgers of the other nodes the Notary node and ERP node can be replicated or vice versa. The limited increase in Computer operations has to do with the real-time transaction processing feature in the blockchain. Job scheduling of external triggers become unnecessary. The explanation for the high increase in the number of IT-controls for Logical Access Security is the access to the blockchain nodes from various locations and the API connectors to the blockchain. Also, security of blockchain critical assets like key pairs and smart contracts desire new advanced IT security controls. Change management, system development, and life cycle management have a profound impact on the entire network with additional IT-controls on forking, and the source code of consensus mechanisms.

By placing the intercompany settlement process in a corporate blockchain, a change is taking place in the ICFR-COBIT IT-controls framework. The corporate blockchain automates its intercompany settlement control environment. The effective and efficient processing of transactions and assurance of the integrity of financial data are technically integrated into a financial blockchain network. The reduction of manual and IT-dependent controls by application controls is visible in the numbers in Fig. 11 in Fig. 12.

At the process level, the segregation of duties of the accounting houses are embedded in the blockchain by the separate nodes. A new risk at this level is drawing up fraudulent smart contracts through the conspiracy of the "trading

partners." It is therefore necessary that at process level a preventive control is designed to ensure the mitigation of this risk. At the IT level segregation of duties together with the ITGCs are preconditions for the proper functioning of the application controls during the control period. At this level it becomes necessary that logging on unwanted adjustments to smart contracts is available, either on the code or by deactivating the smart contract. This is a necessary IT-control to detect fraudulent smart contract.

4.3 Process Level: "Application Level"

The blockchain process optimization of the intercompany settlement (ICS) process has a direct impact on the transaction level controls, manual and IT dependent. The verification and validation of transactions are done by a smart contract. Reconciliation of ledgers then takes place "instantly," the reports are immediately available, and incorrect bookings are rejected thanks to the smart contract. The completeness, timeliness, and accuracy of transactions is covered by the blockchain partly due to the real-time settlement with a timestamp. This digital process optimization eliminates the control task "check transit/interim accounts for imbalances" for some 31 ICS flows.

4.4 IT Level: "Corporate Level"

The intercompany settlement (ICS) process is automated and each participant has their own node with the same financial data. The ICFR internal control measures for the ICS process are focused on a central financial ERP application running on an "on premise" data center. Most of the AS-IS IT-controls of this financial ERP application remains intact in a financial blockchain network. In this case, the IT-controls for the ITGCs, backup and recovery decreases and computer operations there is a small increase. The other ITGCs have an increase of IT-controls because of the new risks and blockchain critical configuration items. The automation of the ICS process further shifts IT-control monitoring to a corporate level due to the control over distributed databases on different data centers. A further shift from ELCs and ITGCs to corporate level becomes necessary due to the collaboration aspects of a blockchain network.

4.5 The External Auditor: "Controls"

At the process level, the blockchain audit trail together with the application controls provides validated and encrypted financial data on which substantive checks can be

carried out, on the entire population, to discover deviations in patterns. The evidence is easy to extract from a node with a sample and a search on a smart contract ID provides a complete audit trail. The data-oriented control of blockchain data in an external audit can be done relatively quickly and in less hours. In this case study, a data-oriented control of the blockchain data compared to the off-chain ERP system data would be a relevant data-oriented control. The system-oriented control will focus on the critical aspects around and partly in the blockchain at the IT level. The IT-controls at ITGC level play a major role because of the distributed databases with "shared ledgers." The essential notary node and ERP node have a major impact on the integrity of the data and the efficiency and effectiveness of the operation of transactions. These deserve extra attention in the system-oriented controls.

5 Conclusions

We conclude from the case study that a "corporate blockchain" for the intercompany settlement (ICS) process digitally optimizes and thus automates the mutual cooperation between companies. Potential new risks that are introduced by a corporate blockchain depend on the use case and the design of this Distributed Ledger Technology (DLT) solution. Some technical risks such as encryption cracking, failing consensus mechanisms, failing nodes, and hacked smart contracts exist on all blockchains.

The case study presented in this chapter demonstrates that the frequently cited risks associated with a blockchain, such as transaction fees, data block size, transaction processing time, and power consumption in consensus mechanisms, do not apply to all DLTs. These are more likely to occur on a public blockchain due to its consensus mechanisms. In this corporate blockchain, these risks are minimal or non-existent because of another type of consensus mechanism, the Node-to-Node (N2N). In this case study the new risks are located around the corporate blockchain and not directly within the blockchain data, because blockchain data is validated, immutable, verifiable and including audit trail. The new risks are the technical risks such as stolen encryption key pairs, interoperability and interaction with off-chain applications and relate to blockchain components that guarantee the integrity of the data.

The advantage of the corporate blockchain is that data quality and security is supported by this technology. The benefits for the auditor such as evidence, transaction validation and verification, reconciliation, real-time reporting, and "smart contract" compliance are increasing, as is the transparency of the transactions at the process level. The data is shared with different nodes which makes obtaining, verifying, and validating of evidence by sampling easier. The blockchain data makes continuous auditing possible and provides proper quality conditions for data-oriented auditing and foresees in correct, complete, and timely data, see the audit trail in Sect. 3.2, Fig. 7.

The objective effective, efficient, and reliable processing of transactions in the ICS process remains the same in the AS-IS and the TO-BE "corporate blockchain"

situation. As demonstrated in the case study, the corporate blockchain turns manual administration and control tasks into application controls on process level with the potential of facilitating back-office employees with data-driven administrative control tooling. At IT level the risks associated with encryption, consensus mechanisms, and/or node governance in a decentralized network requires further enhancements of the ICFR with blockchain-related IT-controls. The risks affecting vital blockchain components require coordination at the corporate level from an IT perspective as the risk effects all the nodes in the network. This makes it necessary to make agreements in a consortium on joint topics such as interoperability, trusted third parties and ICFR audit procedures. The complexity in the management and design of IT-controls is increasing because of the decentral nature and the multitude of participants in this automated and therefore "IT-control environment." The case study shows that ITGCs need to be done on corporate level instead of the usual application and system level. The segregation of duties and the ITGCs need to ensure the proper functioning of the increased application controls in the whole network during the control period.

The amount of test work depends on the decisions made in the consortium and the design of IT-controls, either through agreements between participants or through further automation of controls in the IT-control environment. Options are central IT policy on on- and off-chain components or the configuration of administrator nodes. The ultimate impact on the ICFR and the systems- and data-oriented audit work program depends on this.

The starting point for an Internal Control over Financial Reporting (ICFR) adjustment or external audit when confronted with a financial blockchain system begins with determining the use case, the chosen DLT platform provider, the type of blockchain, the type of consensus mechanism, and the infrastructure. In the preparation phase of your audit, these are the topics to take into account due to the impact on the audit scope and expected IT knowledge when performing the control tasks. Based on this research, a pattern emerges in the ICFR, IT controls. With a corporate blockchain you can observe following patterns in the ICFR:

1. At the entity level, the extensive network automation ensures that an IT-control environment is created between different trading partners and separate legal entities. This leads to more central coordination of IT topics on a corporate level to provide conditions and supervision on the blockchain network.
2. At the transaction level, the manual and IT-dependent controls are replaced by application controls. The process optimization ensures disintermediation reducing manual administrative tasks.
3. A corporate blockchain has big impact on ITGCs. The ITGCs shift to entity level. These require a coordinated approach due to the IT technical control over the entire blockchain network instead of a central application.

Figure 16 depicts the changes in controls on the different levels with the dashed lines. This pattern relates to a DLT, corporate blockchain. It is likely that the same pattern also applies to other blockchain systems with some nuances in IT-controls. This is based on the literature review and the case study. Further research will have to show whether this pattern applies to multiple blockchain solutions.

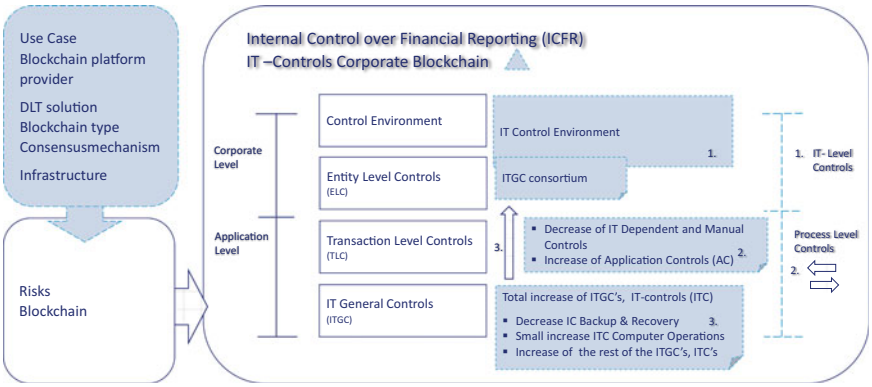


Fig. 16 Overview of the impact of blockchain technology on ICFR

Appendix: Framework TO-BE Corporate Blockchain

Risk	(IT) control objective	(IT) controls
Entity level controls		
The entities, full nodes, receive incorrect or no information from partner nodes.	– Securing segregation of duties of the nodes. – Centrally coordinated audit work for the ITGCs for all nodes.	
Transaction level controls		
Drafting and executing fraudulent smart contracts and collusion.	More eyes principle when drawing up a smart contract.	One-time check when drafting a smart contract. The former manual and IT-dependent controls have been replaced with application controls
IT General Controls (ITGC)		
R1. User access (authorization) and devices (API) to smart contracts and key pairs can indirectly compromise business data on confidentiality and integrity.	– Securing access to the critical components of the blockchain in such a way that unauthorized access is impossible and to prevent unauthorized permission to validate or broadcast transactions. – Provisioning of administrator nodes.	1. All facets that have to do with login authentication: user ID, session time, passwords are stored encrypted, password length and renewal, number of login attempts, etc. 2. Access of privileged accounts for maintenance. Granting the account, its duration, and its revocation. 3. Having local admin rights to install things on your PC or server. 4. Monitoring access of privileged account by logging of activities (change log) done in the application, database, or server.

(continued)

Risk	(IT) control objective	(IT) controls
	Logical access security (consortium)	The abovementioned AS-IS existing controls 1–4 will continue to apply. The emphasis will be on mutual agreements and cooperation of participants in the blockchain when it comes to IT access and permission management.
		1. Consortium with administrator nodes that control access to the blockchain. 2. Monitoring on API connectors with the blockchain. 3. Setting up permission management to check transactions: "validate or write" and who can "broadcast" 4. Access to critical components such as smart contract, key pairs, and consensus protocols.
R2. Suppliers do not facilitate sufficient certainty of their systems and knowledge of blockchain implementations – Not being able to connect with other blockchains. – Upgrading the network is too complex to perform.	– Suppliers must comply with laws and regulations and certifications and have suitable blockchain knowledge and skills in-house. – The blockchain can easily interact with off-chain applications and other blockchains, interoperability. – LCM with a hard fork procedure	1. Purchase of IT hardware and software, cloud applications, and IT contract management. Issues such as security, certification, and financial maturity of the supplier are considered. 2. IT development street, Development, Test, Acceptance, Production (OTAP) in separate environments. 3. Service management, taking care of the IT solution based on acceptance criteria. usability, availability, security, and continuity.
	System development and life cycle (consortium)	The abovementioned AS-IS existing controls 1–3 will continue to apply. The cooperation aspects of joint IT topics require mutual agreements about standards and the recruitment of skilled suppliers.
		1. Consortium on mastery of source code, encryption, data storage capacity, interoperability, and supplier selection 2. Setting up data migration

(continued)

Risk	(IT) control objective	(IT) controls
		strategy—replication from the old to the new nodes (see also backup and recovery) 3. Life Cycle Management strategy with a (hard) fork upgrade procedure.
R3. Changes in a blockchain that have a negative impact on partner nodes and other critical network components.	– Changes are centrally authorized and properly tested in consultation with other blockchain participants. – Changes to the data are "append only."	1. Changes in the production environment on hardware, software, interfaces, or network are recorded, documented as executed. 2. Urgent changes as a result of incidents that have a high business impact due to an error or bug in the IT service.
	Change management (consortium)	The abovementioned AS-IS existing controls 1 and 2 continue to apply. The changes relate to the network and will be centrally regulated and monitored.
		1. Impact analyses on critical components such as smart contract, encryption, and consensus mechanisms. 2. Data changes or smart contract changes "append-only" functionality only correction bookings allowed no change of transaction data. 3. Monitoring for changes in off-chain applications.
R4. The confidentiality and integrity of business data have been compromised by theft of keypairs, code modification in smart contracts, hijacking of a node and making "malicious" changes to the consensus mechanism.	– The blockchain, various nodes and related systems are sufficiently protected against unauthorized access and against stealing, modifying, or damaging smart contracts, keypairs, consensus mechanisms or hijacking of nodes. – The blockchain has detection logging especially on the vital network components.	1. Security violations are detected in the IT service and reported to IT departments that ensure appropriate action. 2. Up-to-date anti-virus software. 3. Vulnerabilities are regularly scanned for the different layers and threats are immediately reported to the responsible IT department who take corrective action. 4. The version of the operating and database management systems are regularly updated with the latest security updates for that version. 5. Known security

(continued)

Risk	(IT) control objective	(IT) controls
		vulnerabilities should be checked and appropriate action taken. 6. High impact security updates are registered, evaluated, and tested before they are installed on the IT services in a planned service window. 7. Security patches on the software and hardware should be checked and installed and monitored to be up to date.
	Security (ELC)	The abovementioned AS-IS existing controls 1 and 7 continue to apply and cover the entire blockchain network. The security consortium will ensure that the security and protection of the entire network remains controlled.
		Security consortium 1. Node governance and security 2. Key ownership and management 3. Protection of critical digital assets: smart contracts and consensus mechanisms.
R5. Damage to nodes and business data that can no longer be repaired.	The entire business data can be quickly replicated from the partner nodes that are unaffected.	1. Backup of software and data are regularly backed up. The critical database information is roll back enabled. 2. Recovery Time Objective and Recovery Point Objectives are configured in the architecture. 3. Periodic test of the backup and recovery scenarios and process works.
	Backup and recovery (ELC)	The abovementioned AS-IS existing controls 1–3 are immediately affected and are obsolete due to the redundancy of data in a blockchain network. The ITGC's system development and security, node governance, capture part of this ITGC. System development when it comes to the

(continued)

Risk	(IT) control objective	(IT) controls
		data migration and node governance location of the nodes that are not affected by a "disaster."
		1. Consortium backup and recovery procedures. 2. Speed of reactivation of a node and the synchronization of data to affected partner node.
R6. The efficiency and effectiveness of transaction processing are hampered by changes in software and hardware configuration items. The scalability cannot be achieved.	– Secure oracles with off-chain applications and other blockchains or smart contracts. – Node scale-up procedures/ standards. – Fast network communication of central and decentralized network components.	1. Access to job scheduling tooling and applications 2. The baseline settings for the infrastructure are checked, based on the best practices of the hardware and checked after corrections to the software. 3. Data base management system (DBMS) and monitoring. Servicing the database, e.g., cleaning repositories or analyzing system loggings. This is done based on privileged account, actions are logged. 4. Maintenance on the IT application, database, network, or server are monitored with procedures and logging.
	Computer operations	The abovementioned AS-IS existing control 1 will be obsolete, 2 will be overtaken in ITGC change management. Controls 3 and 4 will continue to exist but will be arranged centrally because the work has an impact on the entire network.
		1. Consensus mechanism maintenance. 2. Integration and interfacing of data with off-chain applications 3. Control of node scalability

References

Bernsen, M., de Jager, Y., Jongsma, A., Luitjes, W., Verdonk, S., Vousten, L., de Weerd, S., & van Wijk, Y. (2019). *Blockchain assurance, blockchain: kans én bedreiging voor auditor*. Nederlandse Orde van Register EDP-Auditors (NOREA).

Bonsón, E., & Bednárová, M. (2019). Blockchain and its implications for accounting and auditing. *Meditari Accountancy Research, 27*(5), 725–740.

Butijn, B. J., Tamburri, D. A., & Heuvel van den, W. J. (2020). Blockchains: A systematic multivocal literature review. *ACM Computing Surveys (CSUR), 53*(3), 1–37.

Center for Audit Quality. (2019). *Guide to internal control over financial reporting*. Retrieved from https://www.thecaq.org/guide-internal-control-over-financial-reporting/

COSO. (2013, May). Internal control—integrated framework, executive summary.

Folkers, G., & Westra, B. (2017). *IT-B@sed audit* (2nd ed.). Pentagan Holding Books.

Fuller, S. H., & Markelevich, A. (2020). Should accountants care about blockchain? *Journal of Corporate Accounting & Finance, 31*(2), 34–46.

ISACA. (2014). *IT control objectives for Sarbanes-Oxley: Using COBIT® 5 in the design and implementation of internal controls over financial reporting* (3th ed.). ISACA.

ISACA. (2019). *Governance and management objectives*. Retrieved from https://netmarket.oss.aliyuncs.com/df5c71cb-f91a-4bf8-85a6-991e1c2c0a3e.pdf

ISACA. (2021, August). *Blockchain technology audit preparation program*. Retrieved from https://www.isaca.org/resources/news-and-trends/newsletters/atisaca/2021/volume-25/new-resource-evaluates-blockchain-controls

Kloch, C., & Little, S. J. (2019). *Blockchain and the internal audit*. Internal Audit Foundation. Retrieved from https://www.iia.nl/kenniscentrum/vaktechnische-publicaties/publicatie/blockchain-and-internal-audit

KPMG. (2018, October). *Auditing blockchain solutions*. Retrieved from https://assets.kpmg/content/dam/kpmg/in/pdf/2018/10/Auditing_Blockchain_Solutions.pdf

Liu, M., Wu, K., & Xu, J. J. (2019). How will blockchain technology impact auditing and accounting: Permissionless versus permissioned blockchain. *Current Issues in Auditing, 13*(2), A19–A29.

Otero, A. R. (2018). *Information technology control and audit* (5th ed., pp. 12–17). Amsterdam University Press.

Rauchs, M., & Hileman, G. (2017). *Global blockchain benchmarking study*. Cambridge Centre for Alternative Finance Reports.

Revet, K., Durbha, S., & Koorn, R. (2021). *Blockchain/DLT: ERP integration, control frameworks, use cases*. ISACA. Retrieved from https://isaca.nl/wp-content/uploads/Downloads/Square%20Tables/2021/2021%2004%2007%20Blockchain%20ERP%20integration%20and%20ISO%20standard.pdf

Seibold, S., & Samman, G. (2016). *Consensus immutable agreement for the internet of value*. KPMG. Retrieved from https://assets.kpmg/content/dam/kpmg/pdf/2016/06/kpmg-blockchain-consensus-mechanism.pdf

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). *Blockchain technology overview*. National Institute of Standards and Technology (NIST).

Understanding Algorithms

Pieter Oosterwijk, Miranda Pirkovski, and Berrie Zielman

1 Introduction[1]

The central government has been using algorithms for decades now. An algorithm is defined as a set of rules and instructions that a computer follows automatically when performing calculations to solve a problem or answer a question. Algorithms come in many different forms, ranging from computational models, decision trees and other statistical analyses to complex data processing models and 'self-learning' applications.

Algorithms are growing ever more popular, thanks to advancing computerisation and digitisation. Social media, navigation systems and applications like weather apps all work with algorithms. Whenever questions are asked about algorithms (for example: What is their social relevance and which risks do they pose?), the responses can be both positive and negative, in some cases extremely so. The impression arises that algorithms are becoming increasingly intelligent. This is due to the fact that, as the volume of data increases and better hardware becomes available, algorithms can process more data at greater speed, i.e. they become more innovative and wide-ranging. They can also be used for more purposes (e.g. in robotics) and, in their most sophisticated form, 'are able to correctly interpret external data, to learn from such data, and to use these learnings to achieve specific goals and tasks through flexible adaptation (Kaplan & Haenlein, 2019)'. The latter is often referred to as 'artificial intelligence' (AI). AI and algorithms are topics attracting a high level of interest from both private citizens and central government. All hold high hopes for their future potential.

[1] This chapter is based on a publication published by the Netherlands Court of Audit (2020).

P. Oosterwijk · M. Pirkovski (✉) · B. Zielman
Netherlands Court of Audit, The Hague, The Netherlands
e-mail: P.Oosterwijk@rekenkamer.nl; M.Pirkovski@rekenkamer.nl; A.Zielman@rekenkamer.nl

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_5

The wide public interest in algorithms has prompted a plethora of initiatives, standards and guidelines, developed by different stakeholders from all sorts of different perspectives. Virtually all ministries are currently working on standards and guidelines for assessing algorithms. Several non-governmental organisations are also working on the same issue, among them NOREA, the Dutch professional association of IT auditors, and large accounting firms. No comprehensive, practical tools for assessing or analysing algorithms have been developed to date, however. We take the word 'comprehensive' to mean that no efforts have been made to date to bring together all relevant standards and guidelines for algorithms into a single all-embracing framework. The word 'practical' means translating standards and guidelines into specific points that need to be assessed, the concomitant risks, and the questions that need to be answered. The audit framework forms part of this chapter and is publicly available online.[2]

In presenting this chapter, we wish to deliver a practical contribution to the debate about the opportunities and risks associated with the use of algorithms and AI in central government. The developed audit framework may provide a basis for the responsible use of algorithms and underpin the debate about the assessment and monitoring of algorithms. This chapter consists of seven sections. In the section hereafter (Sect. 2) we will provide some background and basic notions about algorithms, and how they are used in governmental practice. The third section presents the framework to audit algorithms. In Sect. 4 the case studies to test and improve the audit framework are discussed. We analysed the results of the case studies and share our main observations in Sect. 5. The main observations of the case studies are discussed in Sect. 6 while also providing several guidelines for the use of algorithms. We conclude the chapter in Sect. 7.

2 Basic Notions of Algorithms

Algorithms are shrouded in mystery and many definitions exist of what constitutes to an algorithm. We maintain the definition that an algorithm is a set of rules and instructions that a computer follows automatically when performing calculations to solve a problem or answer a question. The aim of designing an algorithm differs and depends on the task for which it is created. Several types of tasks can be discerned. There are simple algorithms that given a certain input X produces an output Y by following a well-defined set of sequential steps. This type of algorithm is predominantly used in IS to automate simple processes and is most people have in mind when thinking about an algorithm. *Descriptive* algorithms are used to describe what is happening to an output based on the input data. Sometimes the aim is also to diagnose why modifications to an output variable(s) are happening with *diagnostic* algorithms. Predictive and prescriptive algorithms are most sophisticated and have a

[2]For the full report, please visit: www.rekenkamer.nl/algoritmes-toetsingskader.

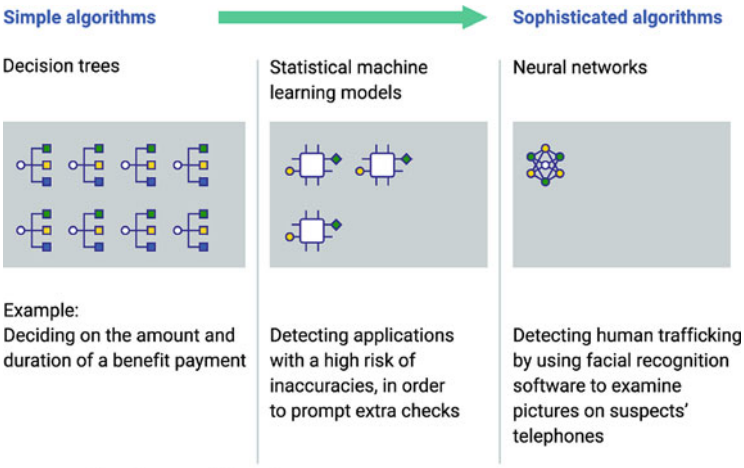


Fig. 1 Classification of algorithms

different aim. As the name suggests a *predictive* algorithm is designed to predict future outcomes based on past data. Predictive algorithms are used to answer the question 'What's going to happen next?'. Prescriptive algorithms go beyond this aim by not only calculating what is likely going to happen next, but in addition by making suggestions of what action should be taken. A prescriptive algorithm is used to answer the question 'What needs to be done?'.

Algorithms can also be classified based on complexity and explainability. In order to produce a detailed classification of algorithms, we used the information contained in the appendix to the letter of 8 October 2019 from the Minister for Legal Protection to the Dutch Parliament (Ministry of Justice and Safety, 2019a, b). The classification is based on the complexity of the algorithms, ranging from simple to complex. Figure 1 depicts the classification of the algorithms.

A *decision tree* is an example of a simple algorithm. Decisions made by such algorithms are easy to explain. An algorithm used for fixing the level of a benefit payment is a good example. A *deep-learning* algorithm, on the other hand, is a complex algorithm. Deep learning is a form of machine learning based on models similar to the neural networks of the human brain. Machine learning employs algorithms that allow computers to learn. The predictions made by this type of algorithm are difficult to analyse. It is not clear to the person making the assessment which data characteristics the algorithm regards as being more important than others. Siri (Apple's voice recognition app) and Alpha Go are two examples of such algorithms. The latter is a computer program developed by Google to play Go, a board game. In 2016, it defeated the human Go world champion.

Sitting between these two ends of the scale are algorithms of varying degrees of complexity and levels of explainability. Our analysis showed that the government uses both simple and sophisticated algorithms and both predictive and prescriptive algorithms. Most of the algorithms presented for our audit are simple algorithms and

medium-category algorithms. No more than 10% of the algorithms presented to us were categorised as sophisticated. The algorithms affect a wide range of government processes and units. A large proportion of these algorithms are used to support operating processes, thus improving efficiency. The government's use of algorithms has three purposes, each of which comes with different effects and risks. Half of the algorithms presented to us are used for the first of these purposes; the remaining half is evenly distributed over the second and third purposes.

2.1 The Use of Algorithms in Practice

We analysed the predictive and prescriptive algorithms used by the central government. This gave us an initial impression of the algorithms used in decisions affecting citizens and businesses. We asked all ministries to report the most important algorithms focusing on predictive and prescriptive algorithms. This gave us an adequate, though not comprehensive, overview of all the algorithms used by central government. We found that about one-third of the predictive and prescriptive algorithms listed by the ministries use automated decision-making. Our analysis did not identify any fully self-learning algorithms in central government, only learning ones. Automated decision-making is used only by algorithms that perform simple administrative tasks that have no effect on private citizens. Our investigation is also aimed at laying bare for what purposes algorithms are used within the Dutch government. The result of our analysis shows that the Dutch government employs algorithms for administrative activities and implementing simple legislation, to improve and facilitate operational management, and to better allocate resources based on risk predictions.

2.1.1 Automating Administrative Activities and Implementing Simple Legislation

A part of the algorithms is used to automate routine human activities. The government makes widespread use of such algorithms. This may generate big efficiency gains, particularly because they enable large volumes of data to be processed much more quickly. These algorithms often involve the (automated) implementation of legislation. A good example of one of these algorithms is the algorithm used for the listed dwellings grant scheme operated by the Cultural Heritage Agency. A decision tree (using simple 'if, then...' rules) is used to decide whether private owners of listed buildings are entitled to a grant. These algorithms are typically prescriptive and perform an automated administrative or financial activity without any human intervention. There is a low risk of errors affecting private citizens with these algorithms, as they are simple algorithms used to perform simple activities, with a high level of technical transparency and a low risk of error.

2.1.2 Improving and Facilitating Operational Management

Algorithms that are intended to boost the efficiency of government processes use more complex data. Experts cannot always blindly adopt their outcomes. These algorithms make a prediction or perform an analysis, which an expert then uses as an aid in his or her work. The Object Detection Sonar used by the Directorate-General for Public Works and Water Management is a case in point. This algorithm indicates the position of objects in the sea, based on seabed imaging, and is used to inform an expert whether it is safe to launch a hydraulic engineering project. Another example is the algorithm used to predict the number of calls made to a call centre, so that the management knows how many staff they will need. Many of these algorithms are predictive algorithms that do not involve any automated decision-making. Although there is a risk of the algorithm making errors affecting citizens or triggering a substantial level of payments, this risk is low. This is because the algorithm has only a preparatory function: it performs an analysis that an expert assesses before taking a final decision.

2.1.3 Targeted Deployment of Resources Based on Risk Predictions

The algorithms used for the third purpose are those that assist officials in selecting cases for further investigation. These algorithms help the government to deploy staff capacity and resources as efficiently as possible. The visa application process is a good example. The Ministry of Foreign Affairs uses an algorithm that helps to classify all visa applications in a number of different 'tracks'. The algorithm sorts applications into potentially successful and complex or high-risk applications, after which a governmental official checks the applications. The algorithm informs the official which applications are likely to need more time, without automatically deciding whether the application should be granted.

Previous audits have found that the central government makes widespread use of risk-based checks and our analysis confirms this. The Tax and Customs Administration does this a lot (Netherlands Court of Audit, 2019a, b, c), for example for the purpose of performing targeted audits of tax returns. The algorithm typically makes a recommendation, and it is then up to an official to decide, based on their professional judgement, whether to follow this recommendation. In other words, no automated decision-making is involved.

The algorithms supporting risk predictions carry a risk that the assumptions underlying the risk profile are not consistent with the law or may produce (undesirable) anomalies due to certain hidden limitations in the input data. The result may be a form of discrimination or the use of special category personal data. There is also a risk of the recommendation made by the algorithm influencing the official's decision.

2.2 Opportunities and Threats of Algorithms

In its Strategic Action Plan for Artificial Intelligence, submitted to the Dutch House of Representatives on 8 October 2019, the Dutch government stated that AI is a key technology (Ministry of Economic Affairs and Climate, 2019). The government is planning to invest €23.5 million in 2021 in the Dutch AI Coalition, a public-private partnership in artificial intelligence. Virtually all the ministries are either developing or already using applications. Some of these involve highly innovative algorithms using artificial intelligence. Algorithms support and in many cases improve operational management and service delivery by organisations. For instance, they enable organisations to deploy people and resources in a highly targeted way when undertaking audits or inspections. Algorithms also enable decision-making processes to be made more transparent and easier to audit. This is because the technology underlying an algorithm, the data used by the algorithm and the algorithm's interactions with these data, are all clearly defined in the form of instructions—instructions that are often absent in human decision-making processes.

In tandem with the advantages and opportunities algorithms offer, the use of algorithms by government organisations also poses several threats. The way in which an algorithm works in central government and its impact on government actions may not be sufficiently clear or may not be clearly explained to the general public. This may be related to the technology used (e.g. neural networks) or to its complexity (e.g. the algorithm may involve too many variables or components). There is also a risk that the algorithm or the data set used by the algorithm may contain certain biases that lead to discrimination. Humans also have certain in-built biases, but there is a risk in using an algorithm that it may be primarily dependent on decisions taken by the programmer or data scientist (for example, on the data used). The programmer or data scientist may lack specific knowledge and experience about the context, e.g. detailed knowledge of a decision on a grant application, even though this knowledge is essential in order to reach an informed decision. Another threat posed by algorithms that learn from data is that we often do not know or cannot foresee in advance what the algorithm will exactly learn, and to what extent there may be undesirable learning effects. Certain correlations in the data used may for instance produce an algorithm that discriminates. Finally, many algorithms used by central government have been obtained from external suppliers. This also applies to IT systems with built-in algorithms. The exact data and mechanisms used by these algorithms are often owned by the external supplier in question, who may wish to protect this information. Where liability or aspects such as the processing of personal data are concerned, the government cannot, or may not wish to, simply rely on the information provided by the supplier. This makes analysing and managing the risks associated with the algorithm more difficult for the government.

Besides being accompanied with threats and opportunities, algorithms are surrounded by myths and hypes. Algorithms are sometimes compared with human intelligence and some of them outperform humans when making certain decisions. The idea may take root that the government has lost control of its own decisions,

which may understandably lead to great unrest. When interacting with its environment, an algorithm may make a very 'intelligent' impression. However, algorithms are not intelligent. They possess neither consciousness nor sense of reality. The basic premise in the government's use of algorithms is that they should lead to greater efficiency in its operational management and the delivery of public services. Algorithms are a means to an end, and not an end in itself. Currently, most algorithms take the form of instructions that a computer follows with the help of data to reach a decision. At the same time, they are becoming both more complex and faster-acting. Combined with the potential for social unrest, this development has created a growing need among auditors and regulators for clear guidelines and assessment criteria that they can use to analyse and assess algorithms.

3 An Audit Framework for Algorithms

Algorithms bring about both opportunities and threats for governments. In this section, we present a framework to maximise the benefits algorithms have to offer while addressing potential risks. The framework was constructed by conducting an elaborate analysis of the extant literature, other frameworks, brainstorm sessions and practical analysis. A more detailed description of the methodology followed to construct the audit framework for algorithms is included in Appendix. Our audit framework contains five different perspectives for investigating algorithms that are depicted in Fig. 2. It provides concrete answers to the questions which risks are associated with algorithms, of which aspects need to be assessed.

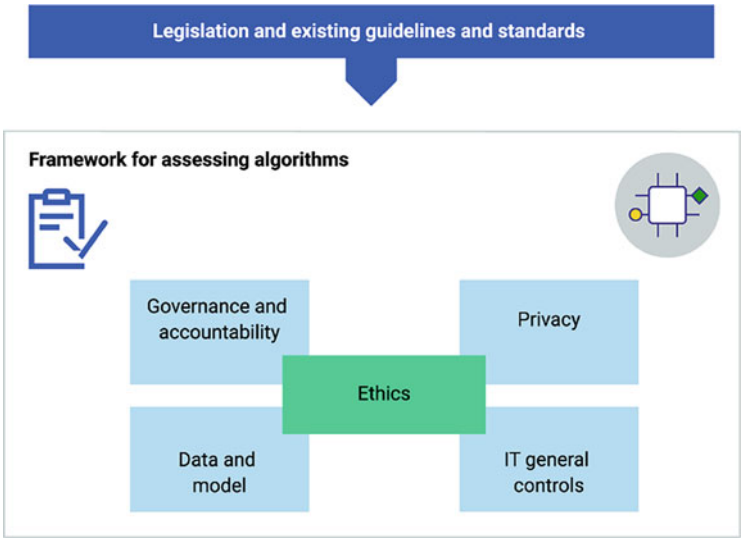


Fig. 2 Five perspectives of the framework

3.1 Ethics

Rather than forming a separate aspect of the assessment of algorithms, ethics are an integral part of the four aspects described above. In other words, ethics are relevant to all other four aspects. We identified four themes from an ethical perspective, based on existing sources (European Commission, 2020) and standards (Bergmann et al., 2019):

1. *Respect for human autonomy*—The decisions made by the algorithm are open to human checks.
2. *Prevention of damage*—The algorithm is safe and always does what it is supposed to do. Privacy is safeguarded and data protected.
3. *Fairness (a fair algorithm)*—The algorithm takes account of diversity in the population and does not discriminate. During the development of the algorithm its impact on society and the environment was taken into account.
4. *Explainability and transparency*—It is possible to explain which procedures have been followed to attain the results. It is possible to explain how the algorithm works.

3.2 Governance and Accountability

The requirements for governance and accountability focus on defining the various elements, i.e. the roles, responsibilities and expertise, the management of the algorithm's life cycle, risk factors in the use of the algorithm, and agreements with external stakeholders about aspects such as liability. We used existing IT governance standards to plan our assessment of the governance and accountability aspect of the algorithms we examined. The assessment of the governance and accountability aspect included in our audit framework is based on COBIT (Control Objectives for Information and related Technology) (ISACA, 2012) (Table 1).

3.3 Model and Data

The model and data criteria deal with questions about data quality, and the development, use and maintenance of the model underlying the algorithm. They include questions about possible biases (from an ethical perspective) in the data, data minimalization, and whether the model's output is tested. We drew on the scientific literature and the day-to-day practice of machine learning. Although the requirements we formulated as part of our audit framework focus mainly on the development of the model, they also cover operation, use and maintenance. Our audit framework is intended to cover the entire range of algorithms, from simple decision-making models to machine-learning models. Given this broadly applicable

Table 1 Risks and controls related to governance and accountability

Nr	Risk	Control
1	There can be no management or accountability without clarity about the purpose of an algorithm	The goal of the algorithm must be defined, also in relation to the social result (outcome)
2	Without an up-to-date analysis of the risks, it is impossible to reach an informed decision as to whether the benefits of using the algorithm outweigh the drawbacks	A structured and documented process for risk management
3	There is a greater risk of error without adequate resources in both qualitative and quantitative terms	An overview of the available resources (qualitative and quantitative) and management thereof
4	No full picture of the life cycle, making the algorithm impossible to manage	Lifecycle management for algorithms or the systems they are part of
5	Lack of clarity about roles, tasks, responsibilities and powers creates risks	Defined roles, described tasks, responsibilities and authorities
6	Performance and quality targets cannot be measured if there is no policy in place	An established approach to quality and performance goals for algorithms
7	A dependency on external experts who leave after developing the algorithm, taking their knowledge and experience with them, means that continuity and management are no longer safeguarded. The algorithm is not monitored and managed	Established agreements with external parties, safeguards to prevent lock-in and excessive dependence. Including exit strategy. Also consider ownership of the data used for the algorithm
8	The algorithm cannot be managed without any monitoring, leading to a higher level of risk	Organised process for monitoring the aforementioned aspects

approach this may inherently mean that certain aspects do not apply to a specific algorithm (Table 2).

3.4 Privacy

Some algorithms use personal data, including special category personal data. Sensitive data such as data revealing a person's racial or ethnic origin, religious beliefs or health status is referred to as special category data and is subject to additional legal protection (Dutch Data Protection Authority, 2022). Algorithms must comply with the statutory regulations on the processing of personal data. The General Data Protection Regulation (GDPR) is an important source of input for our audit framework (Table 3).

Table 2 Risks and controls related to a model and data

Nr	Risk	Control
1	Risk that the algorithm is not fit for purpose. Without agreement on the objectives, there is a greater risk of error and differences of interpretation	Strategic objective has been worked out in concrete terms in aspects/criteria/indicators
2	Without agreement on the objectives, there is a greater risk of error and differences of interpretation	Multidisciplinary approach and bodies
3	The operation of the algorithm cannot be explained or is difficult to explain	Explanation explicitly and, if applicable, making explicit the trade-off between explainability and performance
4	The reasons underlying the choices made in the design and implementation of the algorithm can no longer be traced (explained)	Record considerations and choices in design (such as choices between models, ROC curves) and during implementation. An ROC curve is an aid in assessing the model
5	No continuity in the process or the performance of activities, due to lack of documentation	Up-to-date, complete and accessible documentation
6	Hyper-parameters were selected at random, and the wrong choices were made in doing so	Conduct peer review (four-eyes principle)
7	A lack of transparency for private citizens, businesses and stakeholders; non-compliance with transparency legislation	Publish model (code) to a site such as github.com, including description of operation, data used and/or description thereof
8	The algorithm uses automated decision-making even though this is not permitted, or no opportunities for human intervention	Comply with applicable laws and regulations regarding automatic decision-making
9	Very limited sources of input mean a higher risk of error and non-compliance with objectives and legislation	Involve stakeholders/end users from different backgrounds in development
10	The algorithm does not operate as planned	Implementation of structural checks for correct operation
11	The model was based on the legislation applying in year $t-1$, and is now being used in year t. The legislation (e.g. on margins and limits) may have changed in the meantime, or certain legal provisions may no longer apply	Periodic checks on compliance with and in line with current laws and regulations
12	Incorrect training or testing may lead to overfitting or underfitting, or bias	Among other things, the proven separation of training, test and validation data, 'foreign eyes'/peer review and recording of process/discussions/choices
13	The model leads to undesirable systematic variance for certain individuals, groups or other units (i.e. bias)	Measures to limit, counter and/or compensate for bias
14	There is an undesirable systematic variance (bias) in the data	Check/test for bias and take countermeasures if necessary
15	A lack of separate processing leads to overfitting, which means that the model cannot be used for new observations	Visibly separated training, testing and validation data

(continued)

Table 2 (continued)

Nr	Risk	Control
16	The data is not representative	Test, check.
17	Dependency on third parties with respect to data used	Arrange for all data sources/data used that there are no restrictions/obligations
18	Violation of basic premises and rules pertaining to data minimalization and proportionality	Steering on data minimization, explicit consideration with regard to proportionality
19	The performance metrics are not consistent with the purpose of the algorithm	Good reporting/audit trail (ROC curve)
20	The data on which the model is based is available only after the outcome has been identified	Control on the mentioned aspect (target leakage)
21	The prediction meets the requisite standard	Instruments like ROC curve, confusion matrix
22	The model does not always work in practice	Monitoring output, assessing and reporting
23	People do not know that they are dealing with an algorithm. They are not aware of the consequences this has or of the algorithm's limitations. This may result in incidents, errors or claims for damages	External communication about the model/ algorithm
24	There is a risk that all efforts are concentrated on developing and producing the algorithm, and that no account is taken of the officials responsible for managing the algorithm or of the business aspects of maintenance	Maintenance and management of the technical components, the model, the data used, parameters, etc.

3.5 IT General Controls (ITGCs)

IT general controls (ITGCs) are controls adopted by organisations to ensure that their IT systems are reliable and ethically sound. These controls include conventional IT controls, such as the management of access rights, continuity and change management. The IT general controls incorporated in our audit framework focus on logging data, access rights and password management in relation to the algorithm. The requirements seek to establish whether such aspects have been built into the application and underlying components such as the database and the operating system. The main standards used for IT general controls are the international ISO/IEC 27002 standard and the Government Information Security Baseline (Table 4).

Table 3 Risks and controls related to privacy

Nr	Risk	Control
1	Not compliant with statutory regulations under the GDPR	Keeping a register according to GDPR
2	The design of the algorithm does not take sufficient account of the need to protect privacy	Design principles that ensure privacy
3	Not compliant with statutory regulations under the GDPR	Execute DPIA
4	The algorithm uses automated decision-making even though this is not permitted under the GDPR	No automatic decision-making or no documentation (for example in a privacy impact assessment) why it is allowed
5	Not compliant with statutory regulations under the GDPR; not serving mankind	Established and communicated procedure with those involved
6	Disproportionate use or collection of personal data	Recording principles, work instructions
7	Unlawful action	Recording in PIA, processing agreement/ register
8	Not compliant with GDPR or not fit for purpose	It has been established that the processing of personal data with the algorithm is compatible with the original purpose (purpose limitation)
9	Not compliant with statutory regulations under the GDPR	The lawful basis for processing personal data by the algorithm has been established
10	Violation of Article 1 of the Constitution or Article 14 of the ECHR	Think of ethnicity, skin colour, gender, sexual orientation but also zip code. Not only is checking on data itself relevant, but also so-called proxies, model bias, and so on
11	Profiling as defined in Article 4 (4) of the GDPR; risk of contravening the GDPR	Recording this review
12	Not compliant with statutory regulations under the GDPR	The data subjects are informed about the processing of personal data by the algorithm and the expected consequences
13	Not compliant with statutory regulations under the GDPR	The logic, operation and data used related to the algorithm are described and accessible
14	Not compliant with statutory regulations under the GDPR	Description and substantiation of (possibility of) human intervention in algorithm
15	Data subjects are not informed of their rights or of the algorithms and data used	There is a public privacy policy that also covers the algorithms and data used

4 Case Studies

The audit framework presented in the prior section has been submitted to a practical usability test by assessing three algorithms as case studies. Another aim of the practical usability test was to improve the framework. The aim of the practical usability test was not to arrive at any individual judgements about the algorithms, but rather to aggregate the lessons learned from the analysis. Therefore, we

Table 4 Risk mitigated through ITGCs

Nr	Risk	Control
1	Without any logging information, there is no audit trail for tracing when adjustments were made	Log information is retained and accessible until retention periods have expired. The retention period is geared to the requirements of legislation and regulations and to the control and audit cycle of the data concerned
2	Access rights are no longer up to date	Access rights are periodically reviewed and reconfirmed by the responsible management. If necessary, incidents or amendment proposals are submitted
3	Unlawful access to the algorithm	Job changes and terminations of employment are monitored for adjustment of access rights and for revocation of means of identity and authentication
4	Access rights are issued by unauthorised staff	Access rights are issued to users and administrators upon approval by an authorised officer
5	Risk of the algorithm being manipulated in cases where access rights are incompatible	Access security is implemented according to the 'nothing is allowed unless necessary' principle on all IT resources
6	The more users are granted special powers, the greater the risk of manipulation	Generic administrative accounts (root, administrator) are blocked or can only be used under registration and supervision
7	User groups are difficult to identify	Naming conventions and a system of access rights per user group and/or role apply to setting up access rights to promote maintainability of management
8	Managers and users are difficult to identify	Naming conventions are in place to identify users and administrators to aid management maintainability
9	Unclear who made changes to or worked on the algorithm	Admins perform admin work and regular user work under 2 different usernames
10	The database is open to manipulation if holders of user accounts have access to underlying components	Users have the same rights and restrictions at the application level and beyond.
11	The database is open to manipulation if holders of user accounts have access to underlying components	Job changes and terminations of employment are monitored for adjustment of access rights and for revocation of means of identity and authentication
12	The database is open to manipulation if holders of user accounts have access to underlying components	Using two-factor authentication in high-risk zones, periodically changing passwords, locking accounts when inactive, and blocking after a preset number of false login attempts.
13	Unauthorised access, changes, damage to and/or loss of data. Non-compliance with the law	Changes are tested and approved. Periodic monitoring takes place on the processed changes.

(continued)

Table 4 (continued)

Nr	Risk	Control
14	Unauthorised access, posing a risk of the algorithm being manipulated (changes, damage, loss of data)	Security
15	Back-ups are not consistent with the back-up policy. There is no recovery option, and hence a risk of data loss, if the algorithm stops working	Backup and restore policy
16	There is a much higher level of risk if there is no security by design	Security by design has been used and can be seen as the starting point. Aspects of this can be found in the ISO/IEC 27000 series and beyond.

generalised the findings of the usability test across the algorithms. A further objective was to collect more information on the risks attached to algorithms, in order to supplement the information, we had already gathered in performing our analysis. This enabled us to identify areas in which improvements are needed for the further development of algorithms in central government.

4.1 Selection of Algorithms

To test the audit framework, we selected three specific algorithms as a case study. The first algorithm is a decision tree designed to make recommendations for checks or extra checks of applications from private citizens (depicted in Fig. 3). As a second case study we selected an assessment system for detecting non-standard objects, generating information for regulators and inspectors (see Fig. 4). A facial recognition system for granting individuals physical access to a site or building was picked as the third case study. This algorithm is depicted in Fig. 5. These three predictive and/or prescriptive algorithms were selected because they are daily used, have substantial impact on both private citizens and business, and employ different techniques.

5 Analysis and Main Observations

From the analysis of the case studies, we attained some interesting observations about the use of algorithms by the Dutch government. Hereafter we will discuss these observations and their implications using the framework.

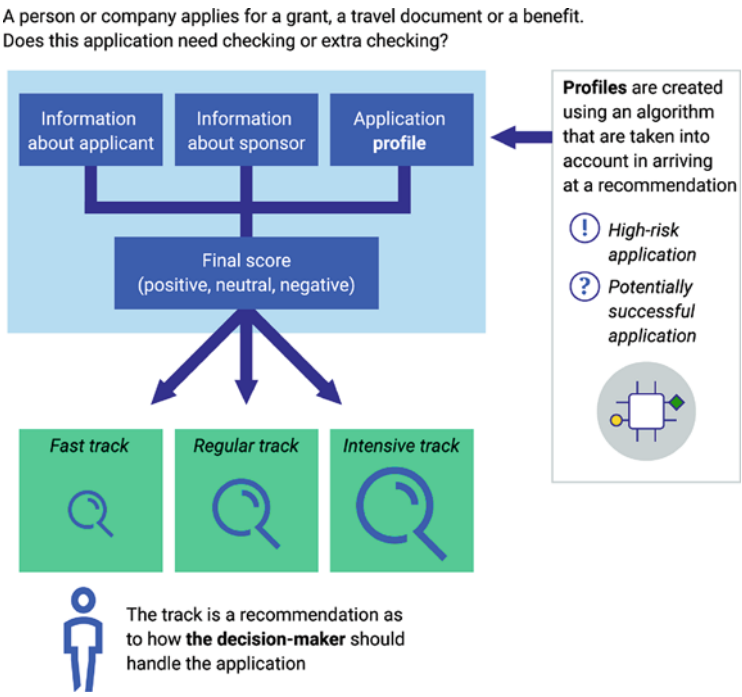


Fig. 3 Decision tree algorithm

5.1 Governance and Accountability

The extent to which the audited algorithms comply with the governance and accountability requirements differs. In the case of one algorithm, we found documentation and records extending over a number of years, explaining the basic principles and requirements applying to the algorithm. In the case of another algorithm, the documentation did not provide any clarity. This does not mean, however, that the ministry in question has no clear picture whatsoever of the purpose and operation of the algorithm. The ministry officials involved have a basic understanding of the principles underlying the algorithm. All three algorithms are subjected to regular assessments and reviews. A review means that the algorithm is reassessed in order to establish whether it still complies with the relevant standards.

In all three cases, we found that the agreements, roles and responsibilities of the parties involved in the use of algorithms in central government need to be allocated and clarified. This is necessary so that each ministry or executive agency, acting under the guidance of the CIO, can obtain a systematic understanding of whether the algorithm is doing what it is intended to do. We also found that, in many cases, no

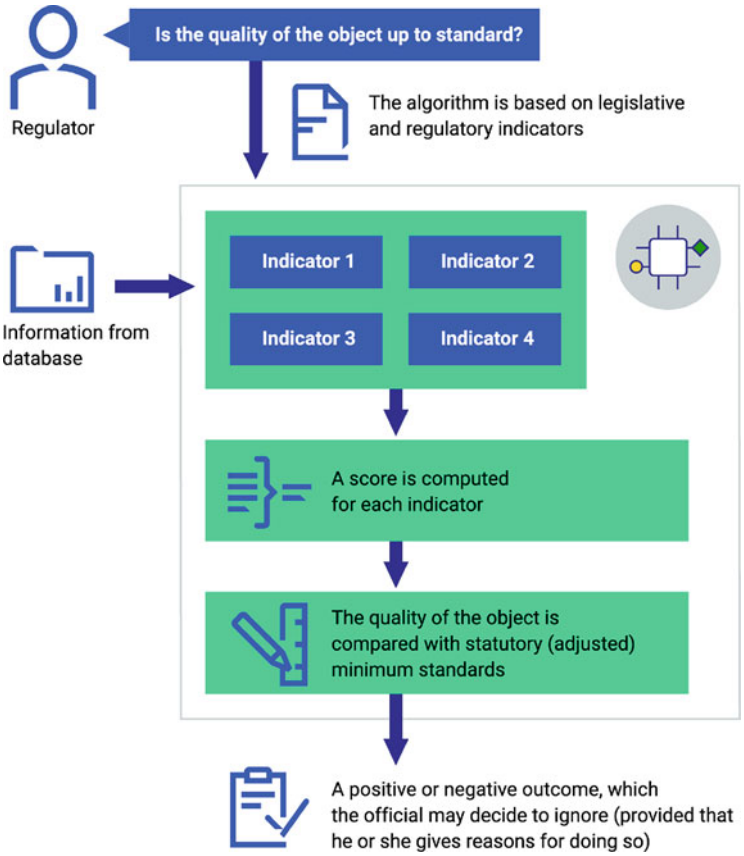


Fig. 4 Assessment system for detecting non-standard objects

system of life cycle management has been adopted for algorithms.[3] While a great deal of time and energy is spent on the design and implementation of algorithms, this does not apply to their sustainment and maintenance. This has both technical and budgetary ramifications. An inadequate maintenance budget, inadequate maintenance or inadequate staffing levels may ultimately cause the algorithm to fall short of new ethical or legal standards.

[3]The term 'life cycle management' as used in this context means the regular maintenance of algorithms during their entire life cycle, so that they remain part of a sustainable and future-proof IT landscape.

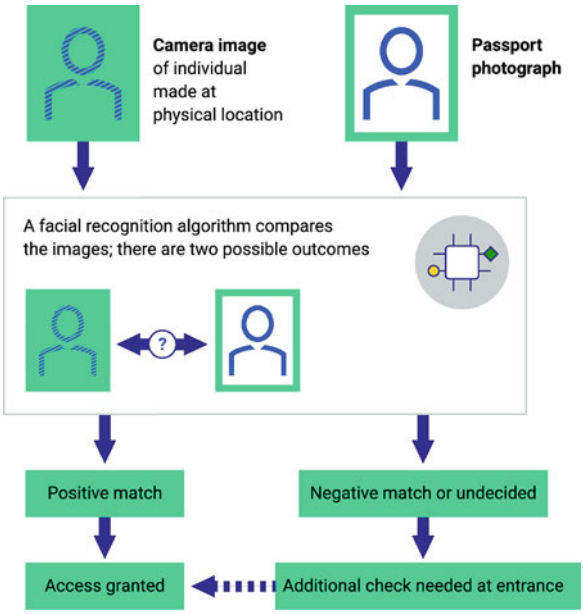


Fig. 5 A facial recognition system

5.2 Model and Data

The principle of explainability is not consistently applied. In the case of one of the three algorithms, efforts had been made to explain the model's outcome. In another case, there was a deliberate policy of avoiding transparency. The algorithm in question indicates only that there is a problem with an individual's application, without explaining why. By designing the system in this way, the executive agency wants to encourage assessors to undertake their own checks and to prevent decisions from being taken automatically without any human intervention.

The issues raised in connection with the model and data aspects include both the methods of algorithm model design and data quality. Where model design methods are concerned, we found that most officials possess sufficient expertise. There are two potential risks here in relation to data management.

The first of these is the use of historical data, which may not reflect certain social changes. This means that practices from the past are applied to the present. For instance, which competencies should a good manager possess? The answer to this question changes in accordance with social trends. If no current data is available based on new legislation, the algorithm cannot be used.

The second risk is data bias. If a specific population group was treated differently in the past, the algorithm will adopt this bias.

Our analysis of the three algorithms shows that not all relevant specialist disciplines are involved in the development of algorithms. While privacy experts, programmers or data specialists are often involved, legal experts and policy advisers tend to be left out. This may result in an algorithm failing to comply with all legal

and ethical standards or not furthering the policy objective in question. Equally, in many cases no action is taken to limit ethical risks such as biases in the selected data.

5.3 *Privacy*

The EU General Data Protection Regulation (GDPR) is the main regulatory framework for privacy and data protection. We tested the three algorithms against our audit framework. The privacy aspect involves elements such as the GDPR personal data processing register, privacy impact assessments, the legal basis for the use of data, and data minimisation. The three algorithms we assessed comply more or less fully with the privacy requirements that we believe apply to algorithms. In the case of one algorithm, the privacy policy, the data used and the algorithms were not publicly available in sufficient detail. This is important in order for third parties such as private citizens to know which data is used, how the algorithm works and how it affects them. This will become an even more important issue in the future, as the volume of data use rises, and algorithms become more complex.

As far as the algorithms we assessed are concerned, we found that there is no easy way for private citizens to obtain information about the algorithms and data used by central government. How, then, can private citizens know what impact these algorithms will have? It is not enough merely to comply with the formal requirements of the GDPR. Personal data and information submitted by private citizens belong to them, and they must know what is done with their data.

Data processing registers are not publicly available in all cases, and privacy statements linked to the algorithms we assessed are not always clear and sufficiently accessible. Although, in some cases, the operation of algorithms and the variables used have been explicitly laid down in legislation, this information is often not easy to read or understand. As a result, private citizens have only a limited understanding of algorithms. In the case of one of the algorithms we assessed, we saw that the officials involved made an extra effort to explain the variables in simple terms. This they did by translating the legislation into a list of frequently asked questions and by producing a video clip.

Building on the *Regie op Gegevens* ('Control of Data') (Dutch Government, 2022) and *MijnOverheid* ('My Government')[4] programmes, private citizens must know who they can contact with their questions about algorithms, how to notify the government about data errors, and how to object to the use of data or the outcome of algorithms. At present, Data Protection Impact Assessments (DPIAs), privacy statements and data processing registers are not sufficiently accessible and are not sufficiently clear to non-specialists.

[4] MijnOverheid is the name of a government website that members of the general public can use to receive digital messages from the government and to view their personal data.

5.4 IT General Controls (ITGCs)

It is clear from the limited amount of documentation that we received from the auditees that, of the four perspectives of our audit framework, it is the ITGC requirements that are given the lowest priority. The main functions addressed by ITGC are access rights and their management, and back-ups. In two of the three algorithms we assessed, little or no information was available as to whether the relevant ITGC standards were met,[5] and auditees were either unable to provide this information or unable to provide it at short notice. In the case of the third algorithm, we did receive the documentation we requested after providing a further explanation. In conclusion, two of the three algorithm owners were unable to provide sufficient proof that they are in sufficient control of the relevant risks. We believe there are two reasons for this.

The algorithm is managed by an external service-provider. Although the relevant officials assume that these external service-providers have proper IT controls, they do not know whether this is actually the case. When we asked for proof, the officials at the ministry in question were unable to provide it or were unable to provide it at short notice.

Although the organisation in question has set higher or different ITGC standards, these have not been laid down in sufficient detail for the algorithm in question.

Our government-wide analysis of algorithms confirms the existence of the first cause, i.e. that the management of algorithms has been outsourced to external suppliers. This applies to two of the three algorithms in our practical test. In the case of one of these, a public-sector shared service organisation (SSO) had been made responsible for managing the algorithm. In the second case, the algorithm was managed by an external service-provider.

As a result, we were unable to establish whether the algorithms complied with a large number of ITGC standards. In the case of the algorithm managed in-house by a ministry, the officials concerned were able to provide documentation on all perspectives of our audit.

5.5 Ethics

Rather than forming a separate aspect of the assessment of algorithms, ethics are an integral part of the four aspects described above. We analysed each use case based on the ethical principles that underpin the framework (see Sect. 3.1).

[5] The relevant standard here is the Dutch Government Information Security Baseline, based on the international ISO/IEC 27002 standard.

5.5.1 Respect for Human Autonomy

Our audit showed that the three algorithms work as an assistive resource; they do not (or do not yet) take any automated decisions. In one case, the technical application (i.e. the algorithm) allows officials to consult several different sources, thus enabling them to take efficient decisions. In other words, the algorithm assists officials.

5.5.2 The Prevention of Damage

In order to prevent any damage, it is vitally important that the algorithm should always do what it is supposed to do. In addition, people's privacy must be safeguarded, and the relevant data must be protected. Unauthorised access may lead to data being changed, damaged or lost. Our findings are explained under the heading ITGG.

5.5.3 Fairness

Fairness means that the algorithm takes account of population diversity and does not discriminate. If no effective measures are taken, the algorithm may acquire an undesirable systematic bias in relation to certain individuals, groups or other entities. In the case of one of the three algorithms we assessed, an external supplier tested the algorithm for any undesirable outcomes. In another case, an external supplier tests all data in advance, in order to assess whether it is absolutely necessary for the algorithm to fulfil its purpose.

5.5.4 Explainability and Transparency

Owners of algorithms are obliged to explain how they designed the algorithm and how it works. All three algorithms were explainable and in all three cases the model designers sought to strike a balance between explainability and performance. Self-learning algorithms were not involved in any of the three cases, and this is one of the factors that make the algorithms in question relatively easy to explain.

In order for procedures to be explained, they need to be clearly documented. We found that this was an issue both in the case of algorithms managed in-house and in the case of those that are fully managed by external suppliers. In the former case, the parameters had been documented, but the model design had not.

In order to assess whether an algorithm adheres to the ethical principles of fairness, explainability and transparency, independent assessors must be able to identify and check the data used. In the case of one algorithm, the data needed to comply with privacy legislation was not stored. This means that, as independent assessors, we were unable to check the data after the algorithm was run (although an

external service-provider did check the data before the algorithm was run). As a result, while the algorithm does comply with privacy legislation, we were unable to establish whether the ethical principles were observed.

6 Discussion

The main observations we derived from our analysis raise some interesting points for discussion. In this section, these points will be discussed and some guidelines are proposed to control the use of algorithms.

6.1 An Algorithm Does Not Have to Be a Black Box

Algorithms are used to support human actions. Our analysis of algorithms used in central government did not reveal the existence of any algorithms that act fully autonomously. We did find algorithms that take simple decisions or perform routine activities in a non-complex environment. Automatically generated letters and messages are examples of such algorithms. Choices about explainability and transparency are part and parcel of the process of developing algorithms. Accountability is another aspect choices. If priorities are given to these aspects in the development of an algorithm, it does not become a black box, but instead a means of assisting an operating process. It should be clear which data it uses, how the data model works, which outcomes it delivers and what sort of impact these outcomes have. It should be possible to make it easier to verify the outcomes of an algorithm than would be the case with the results of a human analysis. Algorithms obtained from private suppliers are a potential problem here. They must comply with the same requirements as those developed by the government itself.

6.2 No Insight Information: Need for Specific Tools

Algorithms are often developed from the bottom up, i.e. on the basis of day-to-day working practices. Senior ministry officials and Chief Information Officers (CIOs) at ministries have little insight in this process. As a result, ministers are unable to mitigate the potential adverse effects of algorithms on government service delivery in a timely manner. The analysis in this audit should help ministers to gain a clearer picture of the way in which algorithms are used by their ministries. A further problem is that there is no standardised terminology in relation to algorithms. This accounts for our finding that ministry officials use different definitions of algorithms and different terms in describing how algorithms are developed, the associated risks and the means of mitigating these risks.

The assessment frameworks in current use are inadequate for the purpose of assessing algorithms. Ministries use universal standards such as the General Data Protection Regulation (GDPR), the Government Information Security Baseline, the Information Technology Infrastructure Library (ITIL) (ITIL Foundation, 2019) and COBIT (ISACA, 2012) for improving the quality and reliability of algorithms and for mitigating the risks attached to their use. This does not apply to all ministries, however. Ministries also use letters to the House of Representatives about big data and algorithms as guidance.

Officials from just three ministries told us explicitly that they regarded ethical aspects as an important component of algorithms. This finding is confirmed by the outcome of our practical test, in which we generally found that no action had been taken to curtail biases (e.g. in the data selection and the risk of discrimination) and a lack of attention for ethical aspects such as profiling. The general standards frameworks do not apply specifically to algorithms and are not used as an interconnected whole. Without any adequate management of and accountability for algorithms, it is impossible to make a clear analysis of the pros and cons of their use. Moreover, the effects of an algorithm are difficult to explain. They may have a significant impact on private citizens in the form of discrimination, inaccurate profiling or financial implications.

Ministry officials all agree that there is a need for a set of standards containing clear, practical definitions of algorithms. At present, there are often differences of interpretation. Opinions differ on whether these definitions should be specific or generic. Some officials regard algorithms as IT tools to which the same generic standards could apply. Other officials claim that the risks attached to algorithms are not always generic, which means that a single, generic set of standards would be impractical. The results of our brainstorming session confirm these findings.

Observation 1: publish clear, consistent definitions and quality requirements.

We urge the cabinet to adopt a clear, uniform set of terms and specific quality requirements for algorithms. Clear, consistent definitions and quality requirements will foster knowledge sharing, streamline processes and prevent misinterpretations. The officials participating in our brainstorming session provided more detailed information about this need for clear, consistent definitions in central government, and in doing so laid the foundations for a ‘common language’ for algorithms. We organised this brainstorming session in conjunction with the Ministry of the Interior and Kingdom Relations, the Ministry of Justice and Security, and the Radiocommunications Agency of the Ministry of Economic Affairs and Climate Policy. The brainstorming session presented these organisations—as pioneers in the use of algorithms in central government—with an opportunity to formulate clear, broadly applicable guidelines and quality requirements for algorithms.

6.3 Predictive and Prescriptive Algorithms Still Under Development: Limited Impact on Private Citizens to Date

Our analysis has shown that central government makes widespread use of both simple and complex algorithms. Broadly speaking, algorithms are used for three purposes:

1. For automating administrative work and simple legislation.
2. For facilitating and improving operational management and/or service delivery.
3. For performing risk-based checks and ensuring that staff and resources are deployed in a targeted manner.

We did not find any fully self-learning algorithms in central government, only learning ones. Only those algorithms that perform simple administrative activities with no substantial impact on private citizens take automated decisions.

6.4 Insufficient Account Taken of Private Citizens

Currently, Data Protection Impact Assessments (DPIAs), privacy statements and data processing registers are not sufficiently accessible and are not sufficiently clear to non-specialists and non-professionals. Private citizens do not know who they can contact their questions about algorithms, how to notify the government about data errors, and how to object to the use of data or the outcome of algorithms. In our opinion, it does not suffice merely to comply with the formal requirements of the GDPR, as this does not generally provide citizens with sufficient information about the algorithms that affect them. Central government can prevent prejudices about algorithms from arising by communicating transparently about the use of algorithms, about the effects they may have on private citizens, and about its own accountability.

Observation 2: inform private citizens about algorithms and explain how they can obtain further information about them.

We urge the cabinet to enable private citizens to access, in a logical location, information on which data is used in which algorithms, how these algorithms basically work and what impact their outcomes have. The algorithms involved here would be those that have a substantial impact on government behaviour or on decisions relating to specific cases, individuals or businesses. One option would be to create a dashboard similar to that created to provide information about large IT projects.

6.5 Improvements for the Responsible Use and Refinement of Algorithms

6.5.1 Governance and Accountability

We found that the agreements, roles, tasks and responsibilities of the parties involved in the use of algorithms in central government need to be further defined and clarified. This is necessary in order to allow ministries to obtain a systematic understanding of whether an algorithm is doing what it is supposed to do. This applies especially to cases in which multiple parties are involved in the development, operation and maintenance of the algorithm. We want to draw attention to the quality of testing of algorithms and continuous monitoring by the ministry.

We found that, in many cases, no system of life cycle management has been adopted for algorithms. While a great deal of time and energy is spent on the design and implementation of algorithms, this does not apply to their sustainment and maintenance. This may ultimately cause the algorithm to fall short of new ethical or legal standards, for instance, or simply to become technically obsolete.

Observation 3: document agreements on the use of algorithms and make effective arrangements for monitoring compliance on an ongoing basis.

Our recommendation to the cabinet is to ensure adequate documentation of the terms of reference, organisation, monitoring (e.g. in terms of life cycle management: maintenance and compliance with current legislation) and evaluation of the algorithm, as this makes clear whether the algorithm is and remains fit for purpose. This also enables the algorithm to be adjusted, if necessary. Especially if algorithms are outsourced or purchased from another (outside) supplier, it is important to ensure that all arrangements relating to liability are laid down in a contract. Our audit framework contains a number of key requirements that can be used as input for documenting such agreements.

6.5.2 Model and Data

Central government uses algorithms ranging from simple decision trees to complex algorithms for image analysis in a wide range of areas. This means that not all the aspects of our audit framework apply to each algorithm. Context also plays an important role in assessing the findings about an algorithm. While explainability may be an important means of providing citizens with information in one particular case, the same level of explainability may be undesirable in another situation, as this would influence decision-makers too much. Moreover, transparency might actually encourage fraudulent behaviour on the part of private citizens. Our audit framework can be refined into a set of standards or minimum quality requirements for any given algorithm.

The issues raised in connection with the model and data aspects include both the methods of algorithm model design and data quality. Where model design methods

are concerned, we found that most officials possess sufficient expertise. There are two potential risks here in relation to data management. The first of these is that the use of historical data may not reflect certain social changes. This means that practices from the past are applied to the present. The second risk is data bias. If a specific population group was treated differently in the past, the algorithm will adopt this bias.

Our analysis of the three algorithms shows that not all relevant specialist disciplines are involved in the development of algorithms. If legal experts and ethical specialists are not consulted, this may result in an algorithm failing to comply with all legal and ethical standards or not furthering the policy objective in question. Equally, in many cases no action is taken to limit bias (for example, in data selection or a risk of discrimination) and ethical risks.

Observation 4: ensure that the audit framework is translated into practical quality requirements for algorithms.

We recommend that the cabinet instructs the Minister of the Interior and Kingdom Relations to ensure that the Chief Information Officer at each ministry is made responsible for translating the audit framework (which is designed to assess algorithms already in use) into a practical set of design standards or into quality requirements for the development of algorithms. The objective here would be to ensure that quality requirements are more practical and could already be applied during the development stage of an algorithm.

Observation 5: ensure that all relevant disciplines are involved in the development of algorithms.

Our recommendation to the cabinet is to involve all relevant disciplines and types of specialist expertise in the development of algorithms. This means involving legal experts, ethical specialists and policy advisers alongside technical specialists.

6.5.3 Privacy

There is no easy way for citizens to obtain information on the privacy guarantees applying to the use of algorithms. This translates into the following practical issues:

Merely complying with the formal requirements of the GDPR is not an adequate means of informing private citizens about how algorithms work, the data they use and their impact.

The government's online data processing register[6] gives readers the impression that it contains all processing registers. This is not the case, however. Nor is there any legal obligation for all processing registers to be published on this website. Our recommendation for privacy is included in Sect. 6.4.

[6]For the register, please visit: www.avgregisterrijksoverheid.nl.

6.5.4 IT General Controls (ITGCs)

In those cases, in which the management of an algorithm has been outsourced to an external supplier, we found that official working with algorithms do not know whether adequate ITGCs have been put in place. Although this is not a problem in itself, we do see certain risks in the current arrangements made for the algorithms we assessed.

Ministries that have outsourced the development and management of algorithms have only a limited knowledge of these algorithms. The outsourcing ministry assumes that the supplier is in control and complies with the ITGC and other standards included in our assessment. We found no proof of this: the responsible minister does not have any information on the quality of the algorithm in question nor on the documents underlying compliance with the relevant standards and refers to the supplier instead.

Where ministries have outsourced the management of algorithms to a public-sector shared service organisation, the situation is the same as where management is outsourced to an external contractor. The department using the algorithm refers to the ITGC guidelines applying at a higher or different level of the organisation. In other words, while disclaiming responsibility, the officials at the ministry using the algorithm cannot explain how the organisation-wide standards apply to the specific algorithm in question.

Observation 6: ensure that clear information is produced now and in the future on the operation of IT General Controls.

We recommend that the cabinet instructs the Minister of the Interior and Kingdom Relations to ensure that the relevant ministers and state secretaries see to it that officials working with algorithms have and retain access to information on the quality of the ITGCs in relation to the algorithms in question. This they can do by asking the party managing the algorithm to present formal statements, such as IT auditors' reports, showing that the ITGCs are of an adequate standard.

6.5.5 Ethics

We found that legislation is sometimes inconsistent with ethical standards. In order to assess whether an algorithm adheres to the ethical principles of fairness, explainability and transparency, independent assessors must be able to identify and check the data used. The demands of privacy legislation mean that a large volume of data is not kept for very long, making it impossible for an auditor to audit it in retrospect. Independent auditors would already like to see an amendment made to the privacy law applying to complex algorithms, and this need is only likely to increase as algorithms grow more complex. This will become clear from the way in which algorithms develop in the coming years.

7 Conclusions

The audit framework that is presented in this chapter makes maximum use of existing information, guidelines and standards. Our audit framework is a practical tool that we intend to use in our future audits. Other government organisations are also free to use our framework to assess whether their own algorithms meet certain quality standards, and whether the risks are sufficiently clear and/or are being mitigated. We hope to have been clear and transparent about any questions that may arise in future audits of algorithms. Our audit framework already gives the ministries a good idea of the risks that we have identified, which means that they can start taking action to mitigate these risks now. The audit framework enables auditors to analyse algorithms from five perspectives:

- Ethics.
- Governance and Accountability.
- Model and Data.
- Privacy.
- IT General Controls (ITGCs).

We investigated how algorithms work in practice in central government and identified potential improvements. Questions about algorithms—what they can do and what risks do they pose?—elicit a wide range of reactions, ranging from extremely negative to extremely positive and everything in between. The audit framework we developed may serve both as a basis for the responsible use of algorithms and as a starting point for discussions on how to manage and monitor algorithms. Our intention is to promote transparency and to foster an open debate about the potential risks arising from the use of algorithms. Transparency about algorithms and control of their operation must become the rule rather than the exception.

Our main conclusion based on the algorithms we analysed in Sect. 4 is that central government pays a great deal of attention to mitigating the privacy risks at play in the use of algorithms. We found automated decision-making only in algorithms performing simple administrative activities that have no impact on private citizens. We also found that the complex algorithms that we analysed do not take independent decisions. Government officials play a prominent role in the use of these algorithms, which assist them in performing analyses and taking decisions.

We also found that algorithms are not a black box for us as independent auditors: we were able to examine and assess them. This does not detract from the fact that there is still room for improvement in 2021, as the use of algorithms is set to increase in the coming years. If algorithms become self-learning, i.e. more complex, they will produce better decisions in terms of speed, quality and objectivity. This will put officials at a greater distance from government decisions on private citizens and businesses. This chapter presents our conclusions and recommendations.

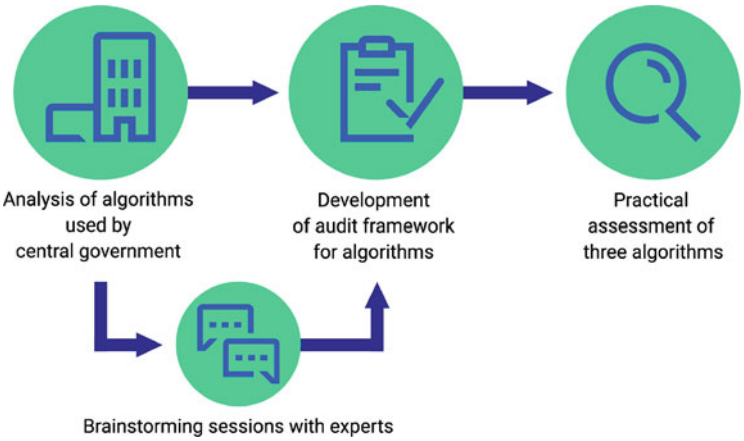


Fig. 6 Method used to construct the framework

Appendix: Methodology of the Audit

We performed an exploratory assessment of predictive and prescriptive algorithms that have a relevant impact on the operating processes of and/or service provision by central government and its associated organisations. This audit was premised on the following audit questions:

1. For which activities and processes do central government and its associated organisations use algorithms, which types or categories of algorithms are there, and what are the risks and effects associated with the use of algorithms?
2. How do the central government and its associated organisations manage the operation and control the quality of algorithms?

In order to answer these questions, and to construct the framework we followed the method depicted in Fig. 6.

Analysis of Existing Algorithms

As a first step, we analysed the types of algorithms used by central government and the activities for which they are used. Our audit builds on the classification described in the appendix to the letter to Parliament about the safeguards against the risks posed by data analysis performed by government (Ministry of Justice and Safety, 2019a, b). The appendix also differentiates between the way in which algorithms are used and the impact that they have. The impact ranges from small in the case of descriptive algorithms to big in the case of prescriptive algorithms.

We asked the ministries to submit examples of prescriptive and predictive algorithms with a relevant impact on the government's operating processes and/or

service delivery. We asked ministries for their most representative algorithms. There was space in the questionnaire for ten algorithms, but this was merely an indicative number. For the purpose of this audit, we wished to receive information about algorithms that have both: (1) a *predictive* or *prescriptive* function, and (2) a substantial impact on government behaviour, or on decisions made about specific cases, citizens or businesses. We looked at the purposes for which these algorithms are used, the impact that they have on citizens, and how they are managed and documented.

As the focus of our audit lies on substantial impact, we elected to analyse predictive and prescriptive algorithms. We wish to stress that we did not seek to undertake a comprehensive analysis of all the algorithms used by central government. We asked the ministries to self-report on the algorithms they used which they believed met our specifications. We explored certain issues in more detail during interviews. We drew up reports of the interviews, which we then asked the interviewees to check.

Brainstorming Session in September 2020

During the course of our analysis, it became clear to us that operational staff responsible for the design, implementation and management of algorithms wished to see closer cooperation among the ministries and needed practical tools for using algorithms in a responsible manner. In order to meet these needs, we organised a brainstorming session on 22 September 2020 in conjunction with the Ministry of the Interior and Kingdom Relations, the Ministry of Justice and Security, and the Radiocommunications Agency of the Ministry of Economic Affairs and Climate Policy. These organisations are pioneering the use of algorithms in central government. Thirty experts from both within and beyond central government took part in the session.[7]

When it became clear during the course of our research that all the stakeholders involved in the use of algorithms worked with different definitions of algorithm-related terminology, we organised a brainstorming session on 22 September 2020. We did this in conjunction with the Ministry of the Interior and Kingdom Relations, the Ministry of Justice and Security, and the Radiocommunications Agency of the Ministry of Economic Affairs and Climate Policy. The aim of the brainstorming session was to identify, discuss, and, if possible, bridge the differences in the terminology used for algorithms. The brainstorming session was broken down into five themes:

- Data-driven
- Data quality

[7] In compliance with Covid-19 restrictions, only a small number of experts were allowed to attend the brainstorming session.

- Artificial intelligence and algorithms
- Artificial intelligence in central government
- Transparency

Constructing the Audit Framework

The audit framework that we used for this audit is based on various types of existing information, parameters and standards. Our audit framework is a practical tool that we intend to use in future audits. However, other government and private-sector organisations are also free to use it to assess whether their algorithms meet specified quality criteria, and whether the accessory risks have been properly identified. The audit framework is a component part of this report and is publicly accessible at: www.rekenkamer.nl/algoritmes-toetsingskader.

Practical Assessment of Three Algorithms

Subsequently, we selected three algorithms from our list and tested them with the help of our audit framework. Our purpose was to refine our audit framework by submitting it to a practical test. By assessing algorithms we can identify those areas where improvements are required in how the central government manages the risks relating to its use of algorithms.

We analysed the predictive and prescriptive algorithms used by the central government. This gave us an initial impression of the algorithms used in decisions affecting citizens and businesses. We asked all ministries to report the most important algorithms focusing on predictive and prescriptive algorithms. This gave us an adequate, though not comprehensive, overview of all the algorithms used by central government.

We found that about one-third of the predictive and prescriptive algorithms listed by the ministries use automated decision-making. Our analysis did not identify any fully self-learning algorithms in central government, only learning ones. Automated decision-making is used only by algorithms that perform simple administrative tasks that have no effect on private citizens.

The ministries' responses show that, with the exception of the Ministry of General Affairs (which does not use any algorithms that are within the scope of this research), they all use both predictive and prescriptive algorithms for delivering services (depicted in Fig. 7). The ratio of predictive to prescriptive algorithms is virtually the same: 60% of the algorithms used are predictive.

The number of predictive and prescriptive algorithms submitted for the purpose of this audit differs from one organisation to another. Large organisations such as the Employee Insurance Agency and the Social Insurance Bank distribute funds, benefits and grants in accordance with statutory regulations. These institutions typically use prescriptive algorithms. The number of algorithms used is not necessarily a

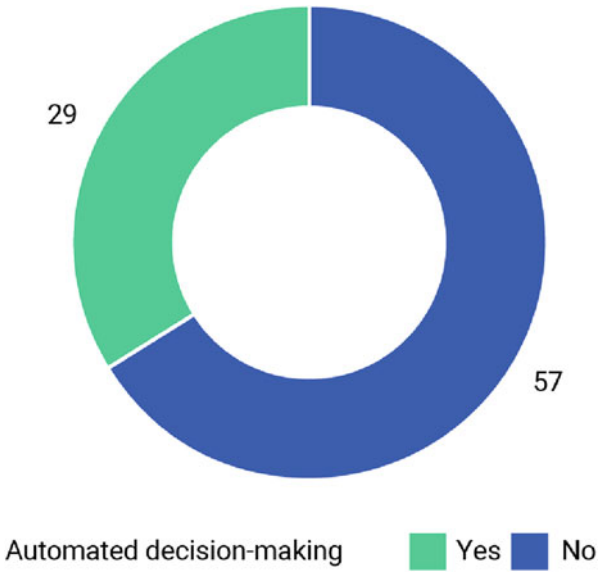


Fig. 7 Overview of the types of algorithms used by the Dutch government

reflection of the degree of expertise on algorithms that a given organisation possesses, as they differ in terms of their complexity and potential impact. We also found that central government does not have any uniform definition or standardised classification of algorithms, which resulted in differences of interpretation among the ministries when submitting their algorithms.

Virtually all the ministries, as well as the central government CIO, informed us that they have no comprehensive, centralised list or overview (i.e. maintained by the ministry itself) of the algorithms used by the ministry in question. As a result, the ministers are unable to timely mitigate the risks and potential adverse effects of algorithms on government services. The same lack of overview also applies to organisations associated with ministries (see the figure above). A number of ministries and the central government CIO told us that our audit was the first step towards obtaining a realistic picture of their use of algorithms.

References

Bergmann, U., Bonefeld-Dahl, C., Dignum, V., Gagné, J.-F., Metzinger, T., Petit, N., et al. (2019). *Ethics guidelines for trustworthy AI*. European Commission. Retrieved from https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf

Dutch Data Protection Authority. (2022, June 3). *Wat zijn persoonsgegevens?* Retrieved from Dutch Data Protection Authority: https://autoriteitpersoonsgegevens.nl/nl/over-privacy/persoonsgegevens/wat-zijn-persoonsgegevens

Dutch Government. (2022, June 3). *Digital Government*. Retrieved from Control of Data: https://www.digitaleoverheid.nl/overzicht-van-alle-onderwerpen/regie-op-gegevens/

European Commission. (2020). *Whitepaper on artificial intelligence—A European approach to excellence and trust. European Commission*. European Union.
ISACA. (2012). *COBIT 5, a business framework for the governance and management of enterprise IT*. ISACA.
ITIL Foundation. (2019). Axelos.
Kaplan, A., & Haenlein, M. (2019). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons, 62*(1), 15–25. Retrieved from https://www.sciencedirect.com/science/article/pii/S0007681318301393
Ministry of Economic Affairs and Climate. (2019). *Strategic action plan for artificial intelligence*. Ministry of Economic Affairs and Climate.
Ministry of Justice and Safety. (2019a). *Waarborgen tegen risico's van data-analyses door de overheid*. The Hague.
Ministry of Justice and Safety. (2019b). *Waarborgen tegen risico's van data-analyses door de overheid*. Dutch Government. Retrieved from https://www.tweedekamer.nl/kamerstukken/detail?id=2019Z19084&did=2019D39751
Netherlands Court of Audit. (2019a). *Cyber security and critical water structures*. Netherlands Court of Audit. Retrieved from file:///C:/Users/bjbut/Downloads/Vertaling+Cybersecurity+WR.pdf.
Netherlands Court of Audit. (2019b). *Data-driven selection of tax returns by the Dutch Tax and Customs Administration*. Netherlands Court of Audit. Retrieved June 11, 2019, from https://english.rekenkamer.nl/publications/reports/2019/06/11/%E2%80%A2data-driven-selection-of-tax-returns-by-the-dutch-tax-and-customs-administration
Netherlands Court of Audit. (2019c). *Informatiebeveiliging Verantwoordingsonderzoek*. Netherlands Court of Audit.
Netherlands Court of Audit. (2020). *Cyber security of border controls operated by Dutch border guards at Amsterdam Schiphol Airport*. Netherlands Court of Audit. Retrieved from file:///C:/Users/bjbut/Downloads/Cybersecurity+Engels+WR.pdf.

Keeping Control on Deep Learning Image Recognition Algorithms

Tjitske Jager and Eric Westhoek

1 Introduction

Can computers become smarter and faster than humans? This question is hard to answer. Yet, the learning capacity of systems provides rich insights into things that we as humans simply cannot see. This involves patterns and connections that have hitherto taken place outside our field of vision. The applications to provide insight into this not only make use of criteria or business rules devised by humans, but also independently search for emerging patterns and deviating observations. Not surprisingly, AI has been recognized by several governments as a key technology for the future. There is broad consensus among practitioners, scholars, and governments AI offers many and new opportunities. Algorithms for instance often support and improve the business operations and service delivery processes of organizations. In addition, algorithms also offer opportunities to make decision-making processes transparent and more controllable.

Using AI algorithms also introduces novel threats to organizations. The complexity of these algorithms (too many variables or components) and the fact that AI oftentimes entails the use of neural networks means that the processes of how the algorithm attained its results become a black box. In addition, AI algorithms and the data that has been used to train the algorithm can contain biases. Further, it is not known or predictable in advance what the algorithm learns, which can lead to undesired effects, especially with algorithms that learn themselves. Another threat relates to algorithms sourced from third-party vendors, where data and algorithms

T. Jager
3Angles Audit, Risk and Compliance, Harkema, The Netherlands

E. Westhoek (✉)
Achmea, Den Haag, The Netherlands
e-mail: westhoek@ese.eur.nl

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_6

are often owned by the third-party vendors. Organizations need a framework to control for these risks while reaping the benefits of AI.

Like most organization insurers have also started to employ AI for their own operational processes. An important process for an insurer is to assess damage to an insured object in case of an insurance claim. As an insurer, how do you quickly identify this damage and help the customer get back on track? In this chapter we present a case study of an insurer ABC that uses image recognition via machine learning to damage to glass horticulture greenhouses. The main benefit to ABC of using image recognition is that it decreased the time to assess the damage, thereby potentially saving more crops that are grown in the greenhouse and thus reducing the claim amount. Using this case study we will present and explain a framework to control and monitor ML algorithms.

Hereafter we will first introduce some aspects of machine learning and image recognition. Then, we will discuss other related frameworks aiming to provide organizations with more control over their algorithms in Sect. 3. In Sect. 4 we present the case study that has been used to establish the framework, and that will aid in explaining how the framework is used. We briefly discuss the case study to demarcate any interesting observations in Sect. 5. The framework that is based upon this analysis is presented in Sect. 6. Because the framework seeks to aid IT-auditors in their work when auditing ML algorithms, we discuss the role of the auditor in Sect. 7. The chapter is concluded in Sect. 8.

2 Machine Learning and Image Recognition

Machine learning allows computers to learn using algorithms. Machine Learning (ML) is about creating algorithms that can learn from data. The novel developments in the field ML have sparked a revolution in which people no longer program (if this, then that) rules within programs, but in which machines themselves derive rules from data. A machine learning algorithm is able to independently extract patterns from data, build models, and make predictions about various things without pre-programmed rules.

Learning in the context of ML differs from programming rules, as a rule-based system does. In a rule-based system, strict rules must be followed by the IS that are programmed into software in advance by humans. The problem with rule-based systems is that the program needs to be instructed step by step what it is supposed to do, while considering its impediments and ensuring that it only does what it is supposed to do. This is a time-consuming and error prone activity as all possible scenarios/situations that might or might not occur in the future must be taken into consideration. In theory, ML has the potential to relate to the intelligence level of a human being, because it is possible to let the system think like a human being, so that the system itself proposes a solution for the established situation. Mimicking this intelligence can be achieved by training the system. ML algorithms can be used to

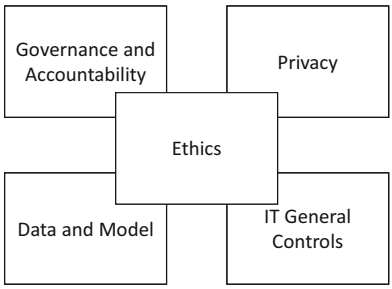


Fig. 1 Five perspectives on algorithm controls. (Source: Netherlands Court of Audit, 2021)

recognize things on an image. In this context, recognize means that the algorithm can classify whether something is on the image or not.

A simple example of such a training exercise is for instance providing an ML algorithm several pictures of Chihuahuas and muffins which can be presented to a computer (input), telling which picture is what (output). If the computer gets enough pictures, it learns to make connections between the different pictures and the computer is able to tell if there is a Chihuahua or muffin in the picture. So, there has been no person who has told the algorithm what the rules are for recognizing a Chihuahua or a muffin. However, humans are required to tell once what the correct output should be, so that the algorithm can make the connections itself between the input and output. This technique has developed enormously in recent years.

3 Related Frameworks

Despite the great social attention for ML algorithms, hitherto there are little concrete instruments to test or analyze algorithms, which is why the testing framework presented in this chapter has been developed. The assessment framework has been established based on existing guidelines and frameworks presented in other works. One of the prime foundational sources used to create our framework, is the framework presented by the Netherlands Court of Audit, (2021). This framework encompasses five perspectives (depicted in Fig. 1), where ethics is not considered separate but integrated in the other four perspectives. This is visually shown in the figure below and will be briefly explained hereafter:

3.1 Steering and Accountability

The "management and accountability" perspective concern the recording of various aspects related to governance: the assignment of roles and responsibilities, gathering of expertise, lifecycle management of the algorithm, risk assessments when using algorithms, and agreements with external parties about, for example, liability.

COBIT (Control Objectives for Information and related Technology) was used to design the assessment of these elements.

3.2 Data and Model

In this perspective, the aspects that deal with the quality of the data and the development, use and maintenance of the model underlying the algorithm are discussed. Whereby possible prejudices (based on the ethical perspective) in the data, data minimization and/or the output of the model are also recognized and tested. The assessment framework is based on scientific literature and machine learning practice. The focus of this perspective lies with the development of the model. Within the perspective, attention is also paid to the operation, use, and maintenance in practice of an algorithm. The researchers note that the testing framework has been made applicable for the entire spectrum of algorithms: from simple decision models to machine learning models. This can lead to a part of the assessment framework not being applicable to a specific algorithm.

3.3 Privacy

This perspective addresses the requirements that the GDPR poses and relevant considerations regarding the processing of personal data, in particular personal data. Legal requirements for an algorithm in the context of the General Data Protection Regulation (GDPR) must be met. Therefore, the GDPR is an important source for the assessment framework.

3.4 ITGC

Traditional IT arrangements should also be in place when using algorithms. Examples of such arrangements are the management of access rights, continuity of the algorithm, and change management. This concerns the embedding in the application and the underlying components that are relevant for the functioning of the algorithm, such as the database and the operating system.

3.5 Ethics

The starting point for the elements of the ethics perspective is the ethical framework proposed by the European Union that describes several ethical principles. Ethics are

not considered a separate element in the testing of algorithms but should be interwoven in the four other perspectives that make up the testing framework. These aspects from this perspective address:

- Respect for human autonomy.
- Preventing damage.
- Fairness (a fair algorithm).
- Explainability and transparency.

Different perspectives come together in the assessment framework. Although various guidelines/testing frameworks were available for these aspects, there was nowhere available an integrated testing framework specifically aimed at an algorithm. The testing framework is a general framework in which the various elements that are important in the control of an algorithm are addressed. The testing framework serves as a practical instrument for the auditor and is a means of control afterwards. Of course, this framework can also be of great value and input at the front end for the quality requirements surrounding the creation and use of algorithms, at the front end of the process. The assessment framework addresses the following aspects:

- Management & accountability
- Model & Data
- Privacy
- ITGC
- Ethics

The assessment framework is generic in nature, which has advantages and disadvantages. The framework provides a good solid foundation to be aware of the risks associated with an algorithm. Prior to the application of this testing framework, general questions were formulated in order to obtain a general picture and the context of the algorithm. The context in practice must guide the interpretation of the assessment framework in practice. Organizations must be aware of all risks that may arise and determine for themselves which aspects apply in this context. This can also mean that other risks can be identified from the specific situation. It is therefore not as simple as finishing the frame and that there is then a controlled algorithm.

The assessment framework first defines which risks are related to the various perspectives. Tied to these risks several safeguards and measures are proposed to control these risks. One element of "People" or "Culture" is not pointed out as a separate aspect in the assessment framework. The culture aspect is less prominently discussed in the assessment framework. However, literature suggests that this is an important aspect not to be overlooked. Ultimately the people within an organization will implement and work with the algorithm and that is why it is important to involve them early in the development so that no resistance to the use of the algorithm might emerge. The framework partially addresses this need by suggesting that multidisciplinary teams should be set up to involve a diversity of people from the organization. As indicated, the testing framework functions as a retrospective check on the algorithm and is not so much focused on the development phase. However,

the assessment framework can serve as input there. It is precisely in this phase that it is important to address these risks.

Outsourcing is not specifically mentioned separately as an important aspect but is briefly mentioned under the perspective of management and accountability and does not appear explicitly in the other perspectives. However, the outsourced processes should be assessed as they might lead to an increased risk. The fact that the part of the process has been outsourced does not mean that you are not responsible as an organization, on the contrary. It is therefore important to recognize this aspect, to estimate the risks and to include them in the research. We note that the nature of the critical questions will not change if the process is internally organized or outsourced.

The Netherlands Court of Audit treats privacy as a separate perspective. The question is whether privacy is an aspect that must be considered when controlling an algorithm. In the context of this research, this aspect is less relevant. The privacy aspect is covered by the data that is used as input for the algorithm, but also access to this data, etc. This is where the risks surrounding privacy come back. If only reliable operation of an algorithm is considered, the privacy aspect is irrelevant. However, when considering the data as important input for the algorithm the privacy aspect is equally relevant.

4 Case Study

In this chapter it will be discussed how models/algorithms are applied in practice. In this chapter we discuss the case study Project Greenhouse. We will first explain the case and then continue to explain the control aspects of the algorithm used using this case. Based on the case in ABC, we will discuss the relevant aspects regarding IT controls in order to realize a complete testing framework for the assessment of robotics algorithms. The ABC has built up considerable knowledge in the various sub-areas of AI. Several AI initiatives have been put into practice. A good example of this is the greenhouse project that focuses on recognizing damage based on aerial photos using a machine learning image recognition component.

4.1 Motivation for the Project

The idea to use robots to inspect damages for the insurance coverage was sparked during the aftermath of a major hailstorm that caused severe damage to greenhouses in two provinces. A helicopter was employed to make an estimation of the damage to the greenhouses. The helicopter flight yielded several aerial photos that provided a basis for the assessment of the damage and provided ample information on how to repair it. The speed of the assessment is important in this context because if the greenhouses remain damaged for too long the ABCs grown in it will be destroyed. A swift assessment of the damage enables countermeasures that prevents further damage. The IT department of ABC was directly involved in the process and

mapped the photos made from the helicopter to coordinates on a map. This enabled other staff that assessed the damage to directly link the helicopter photos to the reported damage. Not only does this process accelerate the assessment of the damage, but by doing so also allow the firm to inform their clients faster about the extent of the damage.

4.2 Image Recognition Greenhouse Damage

The greenhouse project has started at ABC. The aim of the project is to use image recognition to determine the damage to insured greenhouses within 24 h, so that experts have all the information about the insured in the affected area the day after a storm or hailstorm. Within 24 h, ABC wants to know the extent of the damage, and which insured objects are present in the area. For example, a loss adjuster can estimate based on the information whether the ABCs in the greenhouse can still be saved and where repair work must first take place.

After a disaster, an estimate is made of the damage to greenhouses by means of image recognition. This makes it possible to prioritize which greenhouses should be visited first by the damage-experts. This is displayed in a dashboard for the claims adjusters. The dashboard provides practical benefits for ABC who can prevent claims by responding in a timely manner and for customers who can continue to use part of their cash. If greenhouses are damaged, the crops being grown can be lost if, for example, the temperature drops due to broken and damaged windows. As a result of the above case and its evaluation, the company asked itself the following question: Can this be done smarter, easier and could machine learning do something in this?

With this question in mind, a project/innovation team set to work using machine learning and an image recognition algorithm to analyze these aerial photos from an aircraft or drone. The aim is to determine the damage to insured greenhouses via image recognition within 24 h after a major storm or hailstorm. This makes it possible to quickly analyze which crops can still be saved with rapid recovery. In the long run, the amount of damage can possibly be determined based on aerial photos. What is the greatest need and where ABC can still be of added value to limit further damage.

If action is taken promptly, temporary solutions can be used to limit the damage. In order to display the results in a usable dashboard, it is necessary to link the estimated greenhouse damage to the geographical data of the insured greenhouses. It is necessary to geo-code the data of the greenhouses insured within ABC. The coordinates of the greenhouse have been added to the policy for this purpose. A dedicated dashboard for damage-experts was developed that provides all the necessary information to prioritize which greenhouses should be visited first and to act immediately. The data required to make the prioritization process possible consists of a combination of internal data about the greenhouse and the results of a machine learning process that applies image recognition. An estimate can then be made of the damage to a greenhouse.

4.3 Process

In order to get a picture of the situation after a disaster, an external party is used that supplies aerial photos of the affected area within 1 day. The photos are automatically retrieved from the database of the third party BirdsEye, with a dedicated third-party server. The photos are then treated in the database. A roster is then created that contains tiles (squares) using the photos in combinated with GPS coordinates, effectively linking the coordinates to the pictures. The photos are assessed by the algorithm, whereby each tile is assessed in order to be able to determine whether there is damage to the respective pane or not. To assess the damage the tiles are processed by an IS that encompasses different machine learning algorithms.

The first of these algorithms determines the damage and a second algorithm determines whether it is a checkerboard or corner damage (type of damage). In the case of checkerboard damage, the damage is spread over the greenhouse. When this type of damage occurs, windows are broken on several points and little can be done to save the crops of the insured. However, if there is only limited corner damage to the greenhouse, actions will be initiated to limit the damage to the crops of the insured, and to help the insured get back into operation. These outcomes are then combined with the known data of the insured. Using this combined data, a rule-based system then determines the damage compensation that the insured attains based on whether the greenhouse is classified as a "total loss" or whether it can still be saved, also taking into consideration the type of crop harvested. Are they expensive orchids, for example, or is it lettuce, in other words, is it a plant that costs a few euros or a plant that costs a few cents. The results are presented in a power BI dashboard and the damage is prioritized based on these parameters. This ends up in the dashboard that is made accessible to damage-experts.

4.4 IT Department

The IT department is organized at a central level within ABC. ABC has set up the Internet of Things (IoT) platform in collaboration with a large third-party software provider. Within this platform, a private environment in the cloud was realized where the project could be brought into operation.

The IT department focused on building infrastructure gathers gigantic amounts of photos in a few hours, linking them to the firm's data, classifying them, and then providing this information to the loss adjuster using the dashboard. Once the damage-experts had finished their job, the resulting assessment should then be provided to the management after a (major) calamity.

From day one of the project the IT department was closely involved in the project, as it was new within ABC to develop a project in Blue which is a third-party platform. An external consultant from the large third-party software provider was involved in the project to help the organization with the development of the project.

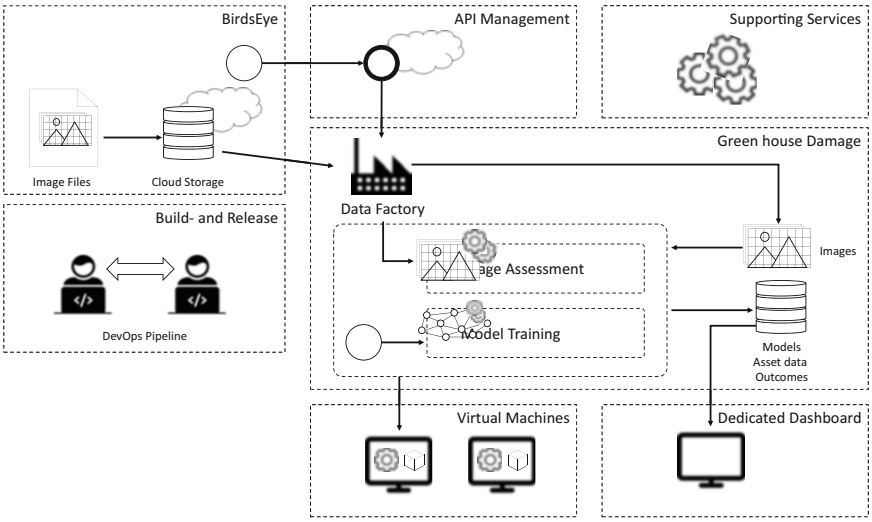


Fig. 2 Graphical depiction of the IT architecture

A development, test, acceptance, and production environment were created for the greenhouse project. Within these environments, all components were deployed. Via IDM it has been arranged who has access to these environments and who has which rights. The IT architecture developed for the project is portrayed in Fig. 2.

The environment that includes both the infrastructure and the code of the application was developed and deployed with Blue DevOps. The data factory takes care of the data transport of the data from the supplier to the storage environment that Databricks uses. The flowchart in Fig. 3 below provides insight into how the AERIAL application processes the data and provides it to the dashboard.

An external party is used to supply photos of the area affected by the calamity within 1 day. The conditions of the photos and other agreements are laid down in a Data Delivery Agreement (GLO). The supplier and recipient of the data have agreed that the photos will be delivered in accordance with a set of quality requirements. The quality of the results from the AERIAL application depends on the timely and correct delivery of greenhouse and ABC data. In the event of an emergency, it is essential that the data in the AERIAL application is up to date.

The quality of the photos is checked before they are offered as an entrance check. Some control aspects are whether the photos are not corrupted and conform to the correct projection as agreed in the GLO. If "errors" appear here, these are logged in the database whereafter the application discards them. The photos are delivered in one set, this is also recorded in the GLO. Upon receiving the set of photos, a sample is taken from that set and if there are no errors, the set of photos is approved. If the photos are removed because they have not been approved, this set of photos (which contained the error) will not be accepted. In this situation, the GLO is serves as a guideline that decide which photo does not meet the requirements and will not be

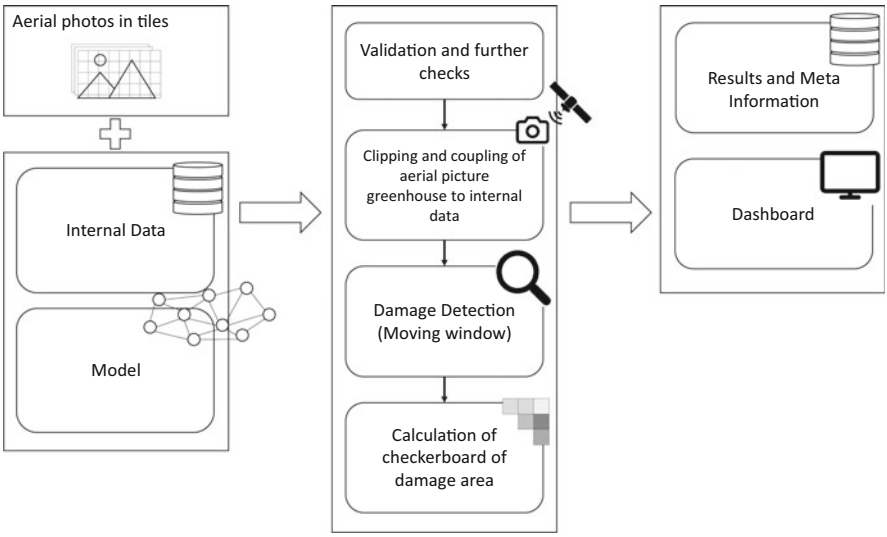


Fig. 3 Flowchart of process image recognition

accepted. The result of this check is provided as feedback to the external party. The aerial photos that are being used are placed in a database on the storage environment and sent to the Databricks environment. Data stemming from internal sources, like as customer data, data about the crop, the insured amount, the coordinates, etc. are included during this process using the Datafactory. The most recent, accurate and most up-to-date model stored in Databricks is used to classify the photos.

Access to the models is arranged via Identity Access Management (IAM) that contains IDM roles. When adjustments to the models are needed, the correct IDM role is required to perform that action. Based on meta data associated with the photos and the internal data of the greenhouses whose coordinates are known, the greenhouses are identified in the photos. Then these photos are classified with an algorithm. The results of the classification process are made available to Power BI via an IDM link. Experts have an IDM role that allows them to consult the database. At ABC there are two administrators who can also change the database, but only in terms of how data is displayed. The management roles to recreate or adjust the models have been assigned to the Data Science department. Any output of the process is thereafter made available to the claims adjuster.

4.5 Project Output

The product resulting from the process is a prioritization dashboard. The loss adjuster sees for each insured that has been affected, what percentage of his greenhouse is damaged, is it corner or checkerboard damage, or is there anything

that can be saved, can measures still be taken to save the crops together with the insured? The address is displayed as a location on the map. The estimated percentage of damage can be seen per cash/policy number. It also states the insured amount, the name of the greenhouse owner, which crops grow in it, etc. Not all data is automatically disclosed. The policy data is now manually updated every few months by someone from ABC, after which it is transferred by the data scientist to the database in Blue. This concerns advice to the loss adjuster based on a prioritization dashboard on which the loss adjuster still makes his own decision. There is no direct decision towards the insured without a human act, assessment, having taken place.

4.5.1 Training and Testing the Model

A machine learning model has been developed that is able to recognize damage on the greenhouses. This is based on classification. This first model was developed with the aim of being able to process a lot of data and train the model as simply and quickly as possible. The photos were tagged using Google Capture. The data scientist has built an application for this. A random photo of a greenhouse is taken and then zoomed in on a part, after which it is labelled by the assessor. This can click on these pieces (see opposite) based on the question is there damage "yes" or "no." A dataset was obtained from the external party to train the model. The prediction of the model was compared with the assessment of the loss adjuster. This results in a total overview, as shown in Fig. 4 below. The damage is plotted on the photo via points.

By training the model it learns to identify the greenhouses. For the training damaged and undamaged photos are provided as input each of them reviewed and tagged by a data scientist. The model learns from these examples. The tagging process is currently still performed by the data scientist. The intent for the future is that this is carried out by the loss adjuster, after which these labelled photos are presented to the model to further train it by employing supervised learning.

Actions have not yet taken in case of deviations from the expectations of the model, at least not automated. When a deviation occurs, a data scientist needs to take an action. The backlog for the further training of the model is developed to automate

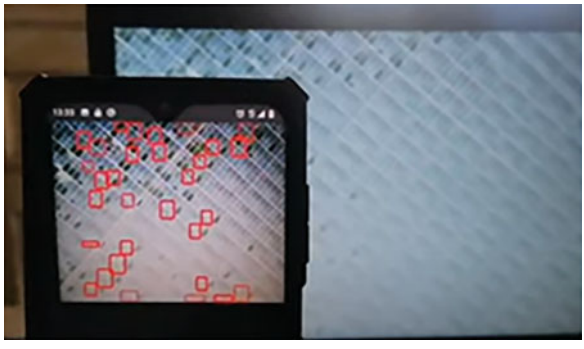


Fig. 4 Plotted damage points on a screen

this process. The "new" models are further trained on the initial model. The model can be trained with many variables and parameters. Each of these variables optimized by looking at a lot of photos that already have a label on them. Depending on the context, a model trained for a specific situation performs better than another. Therefore, the model to classify the images must be selected based on the context as it affects the accuracy of the predictions. Which model is chosen depends on the weather, for example? If there is a lot of cloud cover, the model is chosen that performs well when there is a lot of cloud. If there is also reflection from the sun, then another model is chosen that performed better under these conditions. It is important that the loss adjuster has flexibility in the choice of model. The system now chooses the model itself and projects the model on the data.

Hundred percent accurate classifications are the ideal but will never be achieved. This has to do with the circumstances, which can be different every time. A percentage of 90–95 is more plausible; this number is increased using the feedback loop in the process that allows for further refinements of the model. However, as explained this feedback loop is not yet in place, at least not automated. Currently, the loss adjuster informs the data scientist if there are doubts as to whether something went wrong, after which the data scientist adjusts this in the model, so that the model is improved.

Furthermore, currently there is still no structural recurring process to ensure that the model continues to do what it is supposed to do, that a test run is carried out once every 3 months during which it is checked whether everything still works technically. The following parts can be distinguished here:

- Assignment of ABC-crisis team to start up the IT-system
- Process the photos
- Interpretation of the photos
- Linking the photos to GEO and customer data
- Provide advice to experts

What has not been tested is whether the IS correctly links to other parts of the organization, such as reinsurance, and the back office to receive feedback from the experts. This has not yet been set up in the process. The documentation of the user stories describes the requirements of the end users and what tests need to be performed using what scenarios. All materials related to the tests for the components are included in the use cases. The management team takes care of the automatic regression test.

4.5.2 Finetuning the Model

In a neural network, labels are added that form the recognition of the damage. Based on the training set, the algorithm learns to recognize tiles as "damage" or "no damage." With a limited data set, machine learning models are less accurate, because too much value is assigned to noise. This problem is resolved by offering more than a thousand photos of greenhouses to finetune the model with this larger dataset.

The difference between the old and the follow-up model lies in the technology, namely classification or detection. In the new model, a classification technique is applied to divide a photo into many planes. This technique is potentially much less accurate than the detection technique and can never reach the level that a Yolo V3 or similar new detection models can achieve. Data scientists involved in the project have built an app to show the power of this technique. The latest model is placed in a mobile device, which can then be used to "screen" a photo of a greenhouse for damage. The entire photo is interpreted in one go and the damage, if any, is detected. The center, length, and width of the damage are also identified. Therefore, the output of the neural network is detecting these two aspects.

The follow-up model that will be used is based on the detection technique. This model has already been trained once; however, it still needs to be trained with labelling. As such it has not yet been implemented and remains a task for the loss adjuster. That means that a loss adjuster needs to keep developing the model. For this task, a new front end has been developed together with the damage-experts, to enable the loss adjuster to carry out the task himself. Taken together this also enables the damage expert to train and implement his assistant (model) himself, within the Cornerstone environment that allows for data analytics. In the future, the same flexibility will also enable to remove a model and transfer it to a drone to bring it along to a location.

4.6 Organizational Aspects of the Project

4.6.1 Involvement of the Business Unit

ABCs damage-experts themselves came up with the project proposal themselves. Therefore, there was strong support for the project from the business. During the development of the IS the support of the damage-experts was invaluable as their knowledge as experts was required to label the data and to receive their input on how it would be presented. As a person you very quickly can discern damaged greenhouses from not damaged greenhouses. When training the model, we soon found out that for machines this is far less easy. For instance, glass is transparent and confuses the model, greenhouses are not all the same. Moreover, there are different types of glass. Some greenhouses are partly covered with cloth, others have chalked windows. Not all greenhouses have crops in the greenhouse, others do, and here too a difference can be discerned, one growing orchids, the other tomatoes. Combining these factors makes it really complex to train a model that can account for all these different parameters.

4.6.2 Involvement of Other Departments

Besides the IT departments involvement for obvious reasons, other aspects within the organization also required attention. Within ABC quality and manageability of

data and algorithms was a new topic when the project commenced. At that time, there was far less know how to manage these aspects then there is now. Through this project ABC has gained significant experience.

During the project the innovation team has initiated the Project Impact Assessment (PIA) process, which results in a PIA. Compliance, legal, and security were also involved in this process to provide input from their perspectives. The departments jointly went through the process of creating the PIA. The project/innovation team, of which the business was also part, provided a description of the initiative based on a set of questions. With the help of this set of questions, each specialism then answered the set of questions from the perspective of their own discipline. The answers to the questions laid bare the possible (negative) consequences of the use of personal data for the persons and organization(s) involved where then mapped in a structured manner. In addition, the risks were identified as much as possible. In a joint session between the departments, a coordinated plan was set out to answer the outstanding questions. In addition, during the joint session answers to the questions were discussed. Finally, actions are defined. The process is summarized in Fig. 5 below.

From the assessment of the data, the data stewards were also involved, and this resulted in a BIA that deals with the aspects of information classification, availability, integrity, and confidentiality. All those involved have made separate, individual plans/given advice in their area of expertise. There is not one place of central recording, but this is distributed in the organization in the departments where specialism is invested.

4.6.3 Compliance

Compliance participated in the development of the DPIA and the risks were mapped out. Control measures are then formulated based on this. Specifically, from the perspective of Compliance, the privacy aspect has also been assessed here by means of a Data Protection Impact Assessment (DPIA). DPIA is a risk inventory prior to the processing of personal data. Whether a DPIA should be performed can be determined using the PIA test. To assess whether a DPIA test should be performed, nine criteria have been drawn up by the European privacy supervisors to assess whether the intended processing of personal data poses a high privacy risk for the persons involved. As a rule of thumb, it is prescribed that a DPIA must be performed

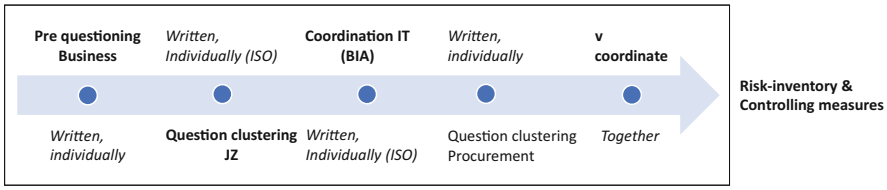


Fig. 5 Process to involve other departments

if the processing meets two or more of the nine criteria. In addition, the project needs to satisfy some criteria drawn up by the Autoriteit Persoonsgegevens (AP). Based on another assessment against these criteria, the conclusion from compliance was that: "Performing a DPIA is not necessary for the project." This is based on the fact that there is no large-scale and/or systematic processing of location data from or can be traced back to natural persons if we take photos incidentally (for example after a calamity or damage report) in execution of the insurance contract. However, it is stated as a point of attention that this should be considered in the contracts with the parties with whom we work together in this regard.

4.6.4 Security

Prior to the implementation, "Threat Modelling" was applied by the Security department, to control the security threats as much as possible. The process to develop sufficient security controls involves identifying potential threats and developing tests or procedures to detect and respond to those threats. It is important to understand how threats can affect systems. A threat model was developed for this purpose, which is based on STRIDE (Kohnfelder & Garg, 1999) threat modelling. STRIDE is a threat model created by Microsoft engineers intended to guide the discovery of threats in a system. The STRIDE model is meant to assess several types of threats to the security of an application. Table 1 shows the different types of threats that can be used to mount a cyber security attack:

Table 1 STRIDE: the different types of threats

Threat	Definition	Example
Spoofing	Impersonating something or someone else	Pretending to be any of Bill Gates, Paypal.com, or ntdll.dll
Tampering	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the network
Repudiation	Claiming to have not performed an action	"I didn't send that email," "I didn't modify that file," "I certainly didn't visit that web site, dear!"
Information Disclosure	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site
Denial of Service	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole
Elevation of Privilege	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but going from a limited user to admin is also EoP

4.7 Benefits of the Project

The "new" process offers many advantages. Without projects like these, damage-experts are less likely to have a clear picture of the damage after a disaster. After a few weeks, claims are still being received from a greenhouse that might have been "saved." The time gain is since there is faster insight into the damage, which means that prioritization can be done more quickly. This provides practical benefits for ABC, which can prevent consequential damage by reacting in a timely manner, and for customers who can continue to use part of their greenhouse. This insight also means that policyholders can be proactively approached to ensure that parts of their greenhouse remain in operation. The model is being further developed and expanded, for example:

- Automatic retrieval of policy data
- Tool to train model for experts
- Algorithm to count number of diamonds or in other words the amount of damage
- Analyze drone photos
- Unlocking photos to customers
- Automatically create a claim and inform the insured

The data scientist emphasizes the essence of the feedback loop, when this is part of the process, the model will get better and better. As a result, the expert is ultimately in charge of training models and giving feedback. Whereby everything around that is automated, so when a data scientist is superfluous.

5 Analysis of Case Study

First of all, it is good to mention that the challenge in this project was to keep the project small in order to make it manageable. The innovation manager indicated that you quickly become enthusiastic about the project and the technology that you quickly think bigger in terms of possibilities. The future wishes have been placed on the backlog. The model, the backlog, and the experience gained form the basis for further development in the coming year.

It has been a good choice to keep the project small, clear, and manageable. A project/innovation team has been started as a basis, in which the necessary disciplines have been involved, namely the data scientist who built the model and the IT department for setting up the environment within Blue. By involving the expertise in the project in this way, attention is also paid to the specifics from each specialism. An example of this is the configuration within Blue that had not been done before and where the expertise from the third-party software provider is used at the initiative of IT. By involving IT in the project in a timely manner, which in the beginning mainly focused on the layout, it shows that a good foundation has been established. As a result, no problems with regard to the technical infrastructure arose in the further

course of the project. Gradually, all relevant departments have been involved in the development and have provided their input. This concerns the compliance, legal, and security departments, but also the data experts, mainly focused on the privacy aspect.

Documentation of activities, assessments, and evaluations are recorded within the department. Within ABC this has been arranged per department, so that the recording is not fixed in one place but is spread throughout the organization. You could also opt for a multidisciplinary approach in which the input is recorded in a central project file. That's a choice. Most importantly, it is implemented and well documented and the relevant disciplines are involved.

The employees who are involved in the project in practice are also directly involved in the project. Separate training program has not been discussed here. The size of the people involved makes it possible to realize direct training on the job. It is a project that came about together and of which everyone saw the added value. This culture and motivation certainly aided in making the project successful.

It could have been better in some respects. These aspects mainly focus on the model itself. In the process, the feedback loop is not adjusted, where the damage-experts provide feedback to the algorithm on the basis of the output, but this is a condition for making the model better or for training. Continuous improvement of the model maximizes the benefits. This is also recognized by the organization and is a wish that is high on the backlog in terms of prioritization. Given the aim of the project to give direction in terms of prioritization where the loss adjuster should go, good results have been achieved here, namely:

- Better prioritization with a focus on saving ABCs.
- Customers back in business faster.
- Faster information from reinsurer.

It mainly serves the customer's interest, which indirectly also entails a financial interest. This is closely related. When is the model good enough? That depends on accuracy and practice. Hundred percent accuracy will not be achieved in practice, because it is different every time. An accuracy percentage of 90–95% should be feasible is also indicated by the data scientist. This is also related to the feedback loop, which can then be adjusted within the process, so that the model becomes more accurate. The desire to further develop the model and the changes and extensions to the backlog are a good basis for arriving at an improved model. Taking together we can identify several management aspects from the case study. These aspects are outlined in Table 2 and will be further discussed in the following section.

6 A Framework for ML Algorithms

Algorithms are getting smarter and are getting ever closer to rivalling human intelligence. The possibilities that machine learning has to offer are developing rapidly. Machine Learning is about creating algorithms that can learn from data. Machine learning allows computers to learn using algorithms. Algorithms are

Table 2 Aspects related to AI-control. Specific use of the aspects is situation- and context-dependent. The maturity level of the organization with regard to the use of these aspect plays an important role in the implementation of the controls

Controlling aspect	Aspect	Orientation
Controlling aspects aimed at:	Control	System oriented
	Processing (incl. feedback loop)	System oriented
	Contents	System oriented
	Outsourcing	System oriented
Prerequisite aspects	ITGCs	System oriented
	Governance	Data oriented
Other controlling aspects	Culture	Data oriented

increasingly influenced our decision-making and are replacing humans evermore for several tasks. An algorithm in the context of computers can be described as a set of instructions that serve to carry out a task. This concerns systems, with "simple" calculation rules based on data, to make decisions or give advice, but also to constitute to more complex learning and/or predictive systems. For rule-based algorithms it is possible to determine how they have produced a certain outcome. However, the complexity of ML algorithms has proven to be far more difficult to unravel.

Therefore, these novel developments in the field of ML also bring about additional risks and have prompted the desire within organizations to get a firmer grip on this technology. ML has a profound impact on the decision-making process within an organization and understanding that impact is key when exerting control. If the decision-making process takes place in a transparent way, firms can also take responsibility for it. Understanding how to create transparency in the decision-making process of an ML algorithm requires insight in what ML is and how algorithms are used. This insight can be harnessed to gain insight into what management aspect is relevant when controlling ML algorithms.

6.1 Fostering Trust in ML Algorithms

The research that study human–robot interaction trust in an algorithm is defined as: "the willingness of users to provide confidential information, accept the recommendations, and follow the suggestions of a robot" (Siau & Wang, 2018, p. 49). Although this definition was originally used in context of robotization, Siau and Wang suggest that the same definition could be applied to ML algorithms. The demand for trustworthy algorithms is only increasing as their influence on society can already be heard felt. In the article: "What /IF—What if auditors play a role in taming algorithms," the Dutch association for accountants (NBA) has outlined three societal trends they observe with regard to the influence of algorithms (NBA, 2020):

- Firstly, our decisions are increasingly driven based on data and the algorithms that use this data.
- Secondly, we use the technology slavishly and trust it blindly without questioning the inner workings of the algorithm.
- Thirdly, if something goes wrong, a culprit is sought as soon as possible without further investigating the underlying problem in the algorithms.

Algorithms that aid in decision-making are in fact not a novel phenomenon; however recently they have become more commonplace and are increasingly being used in a broader sense due to the emergence of Big Data applications. It is relatively easy for these algorithms to determine whether the calculation rules are "good," or whether they meet the standards set for them. These calculation rules have gradually become more complex over time because there are more (input) variables, and the underlying neural network is more complicated. This makes it not only more difficult to check the algorithm, but also to explain how the algorithm works. As a result, decision-making rules have become much more complex due to AI, with learning systems also doing their own reasoning to arrive at a decision. Some of the reasoning that the system then follows to arrive at a decision cannot (or is not easy to) make transparent. The decisions of an AI-based system are difficult if not impossible to analyze. Therefore, frameworks should not focus on testing the technology, but more about testing whether the development of that technology meets the standards to be set. We will now discuss the control areas that will serve as the basis for these standards.

6.2 Control Areas of the Algorithm

Quality of and trust in an algorithm must start at the source by setting clear, unambiguous requirements for the functioning of the algorithm and making careful choices when designing, developing, and implementing it. The creation of the algorithm precedes its use. However, this aspect will be disregarded for further elaboration on the control of the ML algorithm. The management tasks that are involved in exerting control over the algorithm are mainly focused on the aspects of control, process, and content. The preconditions that can be recognized particularly in the field of IT and governance are also important here.

6.2.1 Control

The first aspect to manage in ML algorithms is who is responsible for the algorithm and its functioning. Another aspect of ownership is the responsibility for the data from different sources that serves as input for the algorithm. This data and the associated resources are often managed by different departments within an organization. This also means that they have a different owner that is responsible for the

data provided and the associated quality aspects thereof. This raises the question who is responsible for entering this data as input into the algorithm. The responsibility for and ownership of the algorithm should be recorded. This will be further discussed in the governance section.

6.2.2 Process

An important factor in more complex forms of algorithms like machine learning algorithms is that the creation of such algorithms is fundamentally different from traditional algorithms. Traditionally, the development of a system is a static and well-organized process, and an auditor can make a statement with a certain degree of certainty about the functioning of the system using a conventional audit approach. However, developing systems with predictive algorithms (based on AI) involves a semi-autonomous and iterative process. Under human supervision or even without, an algorithm then processes a large amount of data, which autonomously creates a predictive algorithm. Statistical methods and mathematical techniques are then used to determine that the predictive algorithm does what it is intended to do.

If deviations arise, the same statistical methods and mathematical techniques are used to optimize the algorithm to the desired result. The end-goal of the ML algorithm is ultimately to predict an outcome. Therefore, a relevant question is how well the algorithm performs this task. Signals from other sources like a complaint process for the algorithm should also be gathered. A tool to recognize these signals and undertake action if necessary is recommended in such instance. Concluding, a form of output monitoring that assesses the output generated by the algorithm is relevant here. To further improve the functionality and performance it is recommended to create a feedback loop for the algorithm, so that the algorithm can continuously be evaluated and improved.

6.2.3 Contents

The dataset to train the ML algorithm is crucial to attain the desired results, as confirmed by several studies (Liebchen & Shepperd, 2008). If the data for the machine learning algorithm is inconsistent or inaccurate, the results will also be inaccurate and inconsistent. The principle of garbage in, garbage out is very much applicable in this context. A dataset must always be structured and well-balanced. By (structured) we mean that data should be annotated consistently with labels that describe the data. The more labels you add to the data, the more options there are to train models for specific solutions in the future. In addition, a qualitative dataset must be well-balanced, meaning that for each case (class) the algorithm has to identify there should be an equal number of training examples. An unbalanced set will contain a “bias” to an item and thus make inconsistent predictions. Once a certain amount of data has been labelled, the system will then recommend labels and help users label the remaining data quickly and correctly. With each iteration, the model

makes better predictions, allowing the user to work more efficiently and ensuring labels are properly assigned to the data.

No less important is the risk of whether the data contains prejudices that can lead to, for example, data discrimination. For reliable applications of AI, it is important that the data with which an application has been trained is insightful, in order to be able to find out what a suggestion or decision is based on. Therefore, the origins of the data should be clearly traceable, and it is important that the composition of the data set is reliable and representative to the predictions it is trying to make.

6.2.4 Preconditions Aspects of IT General Controls (ITGCs)

Traditional IT measures are also preconditions for algorithms. Think of the management of access rights, continuity, and change management. Specifically for the control of the algorithm, it is important to have insight into the applications that are relevant to the algorithm and to have insight into the effectiveness of the relevant application controls and the underlying ITGCs. Specifically for algorithms, one can think of the logging information, the access rights, and the password management of the algorithm. The following GITC processes are important here: Logical Access Security (LTB), Change Management (WB), Operations (OPR), and IT Security (ITSEC). Since the GITC play a role in assessing the continuity and verifiability of the algorithm, the process surrounding Business Continuity Management (BCM) is also important.

6.3 Governance

As mentioned before, governance and ownership is an interesting issue in the field of AI. From a governance perspective, the business is responsible for the primary processes it serves. However, using ML usually is part of application functionality using an application. From the first moment that organizations start working on AI they work together with the IT department to install this new application, connect existing applications and set up an infrastructure on which the algorithm can safely perform processes. This raises the question who owns the ML algorithm and who is responsible for the algorithm's actions. It is important to have insight into the tasks and responsibilities of the algorithm. In addition to the benefits of cost reduction and process improvement, robotization also raises questions about its control. What does using the ML algorithm mean for internal controls in the process, now that the separation of functions, as we know it, cannot be realized in this way, for example? The regular risks also remain relevant, such as development, management and maintenance and access security of the algorithm. There are various IT governance frameworks that can provide guidance on this aspect. The perhaps best-known framework is COBIT.

When automating tasks, the general IT risks as we know them in the regular IT audit continue to apply. The difference is that these are now focused on a different object. The IT auditor will have to pay more attention to the management of risk associated with digitization. Internal audit professionals also have a responsibility to understand the risks introduced by ML algorithms and to ensure that their company's controls are well designed and working effectively to mitigate those risks. Unlike humans, who can skip a process step or be inconsistent in the way they process a transaction, an algorithm performs the task in a standard way, without bias or any variation, ensuring a high degree of accuracy. But ML algorithms can also involve risks if the proper controls are not in place and monitored. For example, because the actions an algorithm perform are consistent, any error becomes a systemic and widespread problem in that business process and data set. Or, if there is a business process change, but the ML algorithm has not been modified to reflect that change, it may not perform or introduce inaccuracy. Another potential risk is that if someone gains unauthorized access to an algorithm or the app it is integrated into, it can be modified or used to carry out unauthorized processing. Establishing AI governance and relevant controls in advance should help mitigate risks effectively. By embedding governance, risk management, and controls into the enterprise's mobilization and implementation of AI, organizations can catch problems before they arise. Doing right from the start is much more effective and cost-effective than putting together a patchwork of policies and controls later.

6.4 Human Aspect

Every development or (technological) progress in the past has had consequences for the available jobs pool. With the arrival of AI, employees may be concerned that their jobs are now at stake. It is more likely however, that man will have to work together with machines, whereby the strengths of the people are combined with those of the machines. This is also known as augmentation or collaborative intelligence. It is therefore important to include the human aspect in the process in order to experience development as positive and thus not to see development as a threat, but as an opportunity. The research by Wilson and Daugherty (2018) shows that greatest performance gains come when humans and smart machines work together, reinforcing each other's strengths. As a result, collaborative intelligence is optimally applied.

The human aspect, the culture is a factor to take into account, as is also recognized by Serrurier Schepper and Hiddink (2019). When implementing AI applications, there should be a collaboration at all levels of the organization, involving stakeholders from different disciplines and domains in order to achieve the best result. Collaboration is a key success factor. By involving the employee in the process, giving responsibility and a task, uncertainty can be removed, and the employee also sees the opportunities that this development entails.

Governance is much broader and includes other aspects, namely in the field of compliance, legal, and the human aspect. These other aspects of compliance and legal aspects are less relevant for the control of an algorithm. These aspects play a role in the creation of an algorithm, so they are not discussed in more detail here. For the sake of completeness, I would like to point out that if part of the process surrounding the control of an algorithm is outsourced (outsourcing), the organization remains responsible for the associated risks.

7 Role Auditor

The primary responsibility for quality and trust lies with the organization that develops the algorithm. An algorithm can sometimes become very complex, and as a result no one can fully understand how it exactly works. Sometimes it is possible that the algorithms start working in such a way that even its creators no longer understand why certain decisions are made, let alone that any of the end user can. This requires the auditor to adopt a proactive attitude by looking at the risk assessment, the design and implementation of controls aimed at controlling the algorithm early in the implementation process. Specific knowledge about the chosen application and the underlying programmed code is required, but also knowledge of the process concerned. This therefore requires a joint approach from the business and IT organization, but also from the auditor. Once deployed the algorithm can then be considered as a "black box," whereby it is not always clear which data a system contains and how algorithms work. As a result, it is not always possible to understand exactly how the output is created. Yet, transparency, comprehensibility, verifiability, and explainability are essential and one should always be able to see through afterwards or find out how certain decisions came about. To ensure transparency, comprehensibility, verifiability, and explainability, it is important to be able to answer the following questions when it comes to an algorithm:

- What rules has the model learned?
- How does the model think or reason?
- Who controls the algorithm?
- Who understands the algorithm (and the code)?
- What assumptions and choices were made when training?

To be able to make a well-founded statement about the reliability of an algorithm, an auditor will not be able to suffice with the traditional approach. The assumptions and/or choices made in the development of the algorithm are just as important. For example, about the data with which the algorithm is fed and whether it is sufficient for the purpose of the algorithm, the choice of the algorithm itself, and the methods used to test and optimize the correct operation of the result.

7.1 What Requirements Must an Algorithm Meet?

When auditing we test the performance of a system against a standard. This seems logical, but what is the standard against which to test? There is a certain fault margin that we can tolerate for an ML algorithm. However, this fault tolerance is arbitrary and needs to be put in perspective of a certain context. For instance, if a human life depends on the decision of the algorithm, we would tolerate less faults as when the decision would be for administrative purposes only. As Mona de Boer (2019) in her article it is people who devise, train, and feed algorithms with data. However, the involvement of humans in the design and creation process of an AI also introduces potential risks. The image that must be avoided is that supervision (and/or an audit) of algorithms offers 100% certainty. Just as the (human) civil servant was not flawless, an algorithm will not lead us into a flawless dream world.

European privacy legislation has been tightened further with the arrival of the GDPR. Among other things, the law requires that every decision made by a computer can be explained. This also sets requirements from European privacy legislation in the field of data and algorithms, where integrity and traceability are of great importance. However, the more systems become self-learning, start to feed themselves with data and select their algorithms themselves, the closer the moment comes that their functioning can no longer be understood by humans.

As indicated earlier, the actual use of AI for business processes takes place by means of an application. Just like other applications, these AI supported business processes also need to be adequately controlled. Likewise, for IT-components IT Governance controls should be implemented to ensure the continuous and proper working of the automated processes and to safeguard these processes against unauthorized changes or that hackers procure unauthorized access to the algorithm. The framework of standards is broader than just the IT perspective and will also address the management aspects of control, process, and content surrounding the control of algorithms.

7.2 Systems-Oriented Versus Data-Oriented Auditing

An audit of a ML algorithm can be both system- and data-oriented. Several sequential steps can be followed to audit the algorithm. The audit starts with a risk-based audit approach during which the auditor analyzes the risk that the financial statements are materially misstated. The auditor then adapts the approach to the outcomes of the analyses by planning system- and data-oriented activities. Using the system-oriented procedures, the auditor determines to what extent use can be made of the measures that the organizations themselves have put in place to prevent or discover a material misstatement in the financial statements. Depending on the outcome of this first step, substantive procedures are performed to obtain sufficient certainty about the quality of the accounting. The expectation is that in a mature organization in the

IT field you should be able to audit system-oriented, on realization where you focus on the process and not on the input/output. You also include other aspects and signals from other angles in this assessment, such as management information about complaints. Are there any signs that could indicate that the algorithm is not working properly? However, there is no standardized approach to address this question as it depends strongly on the context.

In the approach to assessing the mastery of an algorithm, a measurement moment will be: Can the process approach be applied, or will the data-oriented approach have to be applied? This also depends on the maturity level of the organization and the way in which the algorithm was created. The process approach will be chosen for an organization with a solid maturity level. Before the start of the research, this consideration must first be made, which options are available and on that basis the choice for a process- or data-oriented approach can be made.

7.3 Conclusion Role of the Auditor

As indicated, the primary responsibility for quality and trust in the control of the algorithm lies with the organization that develops the algorithm. Auditors can further strengthen this trust by checking whether the algorithm is doing what it is supposed to do and by asking critical questions that are in the public interest. The assessment of the (IT) organization and associated (IT) control measures has remained unchanged in all those years: there is always a person behind the (development of) systems and the auditor therefore focuses strongly on this. In a sense, you could say that AI—with a permanent feedback loop that provides learning capacity—is an extremely fast form of change management. In essence, algorithms are mainly about applying calculation rules yourself in order to also be able to make changes in order to make decisions. However, it is not just about checking the algorithm itself with the organization and the management measures surrounding it, but also paying attention to the data used, the methods used in the development and (continuous) optimization of the algorithm. These aspects of management, process, and content should therefore also be part of the assessment framework and thus the audit approach.

8 Conclusions

How do we make an algorithm reliable? This sounds difficult and complex. Control is part of one of the tools out there to manage the adverse effects of algorithms. These adverse effects are often reflected in the media, but of course many good things are also done with the help of algorithms. What risks do we see and how can we ensure that the AI application is created in a controlled manner and works reliably. This starts with having sufficient competences to understand how this works, both when it was created and afterwards how it should be investigated. Relevant control aspects

that are presented in this chapter are the minimum aspects that can be expected in the assessment framework aimed at assessing the control of an algorithm. This concerns aspects aimed at control, process (including the feedback loop), and content, but also aimed at preconditional aspects. Summarizing from my research, the following aspects are important that must be addressed in a testing framework aimed at controlling an algorithm. Control aspects aimed at:

- Control
- Process (including feedback loop)
- Contents
- Outsourcing

Preconditions aspects:

- ITGCs
- Governance

Other management aspects aimed at:

- Culture/human aspect
- Compliance aspect[1]
- Legal aspect (see Footnote 1)

It appears that the aspects discussed just now are most affected. As far as we are aware of, we did not identify any other control aspects that should be added to the testing framework aimed at assessing the control of an algorithm.

From the case study and other works it seems that enhancing knowledge within an organization about the inner workings of the algorithms is important. Therefore, a multidisciplinary approach is also important as it combines the knowledge of several disciplines (e.g., business and IT). Another finding is that it makes no sense to make a checklist and go through it in order to have an overview of all the risks. The risks associated with an algorithm depend on the context in which it is used. It is far more important that within the organization there is awareness and a basic level of knowledge about the algorithm. Knowing everything about the algorithm is virtually impossible, but organizations must be able to recognize the aspects, the level of consciousness, in order to hook up the right people from their specialism to the controlling process. These capabilities are required to ultimately be able to conclude that the application has been carefully developed, whereby the identified risks in the process have been thoroughly controlled in order to arrive at an algorithm that works sufficiently reliably. This is not only relevant for the organization itself, but also for supervisors of the algorithm, for example. It is therefore recommended to carry out the entire process from a multidisciplinary point of view, including drawing up the risk analysis. In this way there is timely insight into the risks in the various specialisms and this can be considered during the process.

[1] Not discussed in this chapter.

The study published by the Netherlands Court of Audit (2021) offers good frameworks for general control, reliability and safety, as well as model quality, data quality, and ethics, which are integrally interwoven with it. However, it can be noted that the assessment framework is generic in nature. It is a good solid foundation to be aware of the risks associated with an algorithm. The context must be leading for the interpretation of the assessment framework in practice, the general questions that must be answered prior to the application of the assessment framework help with this. The framework focuses on accountability afterwards, but also offers guidelines in advance in the field of quality aspects that are already relevant during the development and realization of the algorithms. Some of these have been identified separately, some of them have been included in the elaboration of the five perspectives. I have already noted that "outsourcing" is not specifically mentioned separately but is briefly mentioned under the perspective of management and accountability and does not appear explicitly in the other perspectives. Here too, the organization bears responsibility for the risks. In my opinion, this element could have been worked out emphatically in the assessment framework.

The "People/Culture" element is also not specifically mentioned in the assessment framework, but this can also partly be seen in conjunction with the multidisciplinary approach. However, the case study points out that involving people and assessing the culture is an important aspect. The cultural aspect should certainly not be underestimated. Ultimately people have to implement the algorithm and that is why it is important to involve them early in the development. Doing so will ensure that employees within the organization are not surprised by the change during the implementation, and consequently will resist it less.

We conclude that the assessment framework provides a broad basis for an audit. It is a generic framework that must be tailored to the situation and context of the algorithm. As mentioned, the testing framework serves as a practical instrument for the auditor and is a means of control afterwards. Of course, this framework can also be of great value and input at the front end for the quality requirements surrounding the creation and use of algorithms, at the front end of the process. It is important to involve the "People/Culture" element, not only in the development, but also the people in the organization who will be involved in the implementation, so that they are included in the change and are involved in the implementation. Don't be surprised by. This is partly reflected in the multidisciplinary teams. There is some overlap in this but is not mentioned separately as an aspect.

It is up to the organizations themselves to gain insight into the algorithms and their use and to realize how powerful and important the role of algorithms in a certain process can be. To subsequently deal with this in a good and controlled manner, focus should not only be on the opportunities and on the effectiveness and efficiency of the process, but also on the awareness and importance of the creation, implementation, and control of the process. Is the algorithm able to be 'accountable'? The management aspects recognized from my research offer the auditor guidelines for assessing the reliable operation of an algorithm that is relevant to the audit object of the IT auditor. These management aspects partly overlap in the available assessment framework published by the Court of Audit, which has been elaborated based on perspectives.

References

de Boer, M. (2019). Vertrouwen in een algoritmiserende samenleving: Hoog tijd om algorithm assurance op te pakken. *De IT-Auditor*. Retrieved from https://www.deitauditor.nl/wp-content/uploads/2019/04/Hoog-tijd-om-algorithm-assurance-op-te-pakken.pdf

Kohnfelder, L., & Garg, P. (1999). The threats to our products (Vol. 33). Microsoft Interface, Microsoft Corporation.

Liebchen, G. A., & Shepperd, M. (2008, May). Data sets and data quality in software engineering. In *Proceedings of the 4th International Workshop on Predictor Models in Software Engineering* (pp. 39–44).

NBA. (2020). *What if Wat als auditors een rol gaan spelen bij het temmen van algoritmes?*

Netherlands Court of Audit. (2021, January). *Understanding algorithms*. Retrieved from https://english.rekenkamer.nl/publications/reports/2021/01/26/understanding-algorithms#:~:text=The%20Court%20of%20Audit%20found,use%20and%20operation%20of%20algorithms

Serrurier Schepper, M. S., & Hiddink, T. (2019). *Artificial Intelligence in actie* (1st ed). van Duuren Management.

Siau, K., & Wang, W. (2018). Building trust in artificial intelligence, machine learning, and robotics. *Cutter Business Technology Journal, 31*(2), 47–53.

Wilson, H. J., & Daugherty, P. R. (2018). Collaborative intelligence: Humans and AI are joining forces. *Harvard Business Review, 96*(4), 114–123.

Algorithm Assurance: Auditing Applications of Artificial Intelligence

Alexander Boer, Léon de Beer, and Frank van Praat

1 Introduction

Algorithm assurance is a specific form of IT assurance that supports risk management and control on applications of risky algorithms in products and in organizations. These algorithms will often be characterized in organizations as applications of Artificial Intelligence (AI), as advanced analytics, or—simply—as predictive models. The aim of this chapter is to introduce the concept of algorithm assurance, to give some background on the relevance and importance of algorithm assurance, and to prepare the auditor for the basic skills needed to organize and execute an algorithm audit.

An algorithm is essentially a recipe to solve a specific class of problems using a finite sequence of well-defined instructions. Starting in an initial state with input data that characterizes the problem, execution of the algorithm proceeds through a finite number of successor states, terminating in a final state with output data that solves the problem.

The concept of an algorithm is an important vehicle for communication of scientific results between computer scientists, and mathematically proving desirable properties of algorithms is an important part of those scientific results. Those desirable properties may for instance be related to the worst-case running time of the algorithm, the characterization of the specific class of problems it solves, or the qualities of the solutions it comes up with. In practice, this allows programmers to apply routine algorithms without further research if they can ascertain that (1) the problems they want to solve belong to the class of problems that can be solved by the algorithm, and (2) the desirable properties of the algorithm match the task at hand.

A. Boer · L. de Beer · F. van Praat (✉)
KPMG, Amstelveen, The Netherlands
e-mail: Boer.Alexander@kpmg.nl; deBeer.Leon@kpmg.nl; vanPraat.Frank@kpmg.nl

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_7

Algorithms are in common parlance specifically associated with the field of AI (see Sect. 3 in chapter "Introduction to Advanced Information Technology" of this book), because that field aims to build computer programs that can perform tasks that would otherwise have to be performed by a skilled human being. The field of Artificial Intelligence pushes the envelope, looking to expand the class of problems to which algorithms can be applied. Sometimes with spectacular results, but also with considerable risk. AI uses of computer programs often introduce considerable risk, and this risk can be attributed to risky applications of algorithms to real-world problems that can have a profound impact for those involved. Applications of algorithms are, in essence, always fundamentally questionable given the nature of the problems to be solved. If the application of the algorithm to the class of real-world problems is sufficiently well-understood and becomes routine, it stops being of interest to Artificial Intelligence. Or the media, for that matter.

An important tool in the toolbox of AI is the machine learning algorithm, which is capable of adapting to the problems it is exposed to by learning. An ML algorithm only has a capability to learn to a certain extent, and that extent is often not well-understood. This type of algorithm is trained by exposure to data reflecting the class of problems it is supposed to solve. In chapter "Introduction to Advanced Information Technology," Sect. 3.1 of this book a distinction was made between three different modes of learning: supervised learning, unsupervised learning, and reinforcement learning. This distinction is going to be important for understanding this chapter.

Algorithm assurance is not about the properties of the algorithm itself, but about its implementation in a computer program and about its application to real-world problems. The object of assurance is never the algorithm itself. It is a computer program, or component of a computer program, containing implementations of a risky algorithm or algorithms, to be reviewed in the context of a task in which it is applied or a prospective class of tasks in which it may be applied (in case of for instance admissibility in a market).[1]

In this chapter we will introduce the algorithm assurance engagement as a specific type of IT audit. After a general discussion of the background of algorithm assurance and the type of IT applications we are concerned with in this type of engagement, we will extensively discuss the scope of an algorithm assurance engagement, how to approach the risk assessment that should take place initially, how to set up and audit plan, and the audit techniques and tools that play a role in an audit plan. In Sect. 7 we discuss some examples of development skills that may be called on by the audit team during an engagement to help it judge risk and find problems. Throughout the chapter we use a running example—introduced in Sect. 3—and discuss the various sections in context of that running example throughout the chapter.

[1] Because the term algorithm in this context has become equated to implementations and applications of algorithms, we will indiscriminately use the term algorithm wherever we mean implementation or application of the algorithm.

2 Background

We are increasingly surrounded by, and dependent on, applications of AI technology. And its potential dangers are increasingly worrying us. Dystopian perspectives of the future in literature, film, and games demonstrate the potential ramifications of decision-making computers using data about us. Basically, these dystopian perspectives have been introduced since the idea of general purpose computers started gaining traction.

Over the last decade these worries have led to terms like AI, algorithm risk, and algorithmic bias entering common parlance in the context of *burning platform*[2] situations and in broad and general discussions about the risks and ethics of application of AI. These discussions have led to new legislation focusing on the uses of data and the uses of algorithms. For instance, the General Data Protection Regulation, which limits the uses to which data about people can be put in automated decision-making. Another example is the Artificial Intelligence Act, which addresses various forms of manipulation and harm caused by AI. The Digital Services Act and Digital Markets Act address unfair competitive advantages caused by data collection and manipulation through recommendation algorithms. These discussions have also brought the topic of accountability for harms caused by algorithms to the attention of organizations.

The implementation and application of algorithms has therefore also become a Governance, Risk, and Compliance topic. As a consequence, there is a growing call for algorithm assurance services. But not every algorithm—in the computer science sense—is an object of concern. Only algorithms that create unchecked risks, and only if their implementation, or application to a problem, may cause harm. In general, these criteria touch upon the characteristics of AI applications. For effective Governance, Risk, and Compliance over algorithms, these risky algorithms need to be identified and tracked first.

2.1 Common Risk Factors

The identification of key risks the algorithm poses to the company is a critical step in effective risk management. This step needs to be comprehensive. If a potential risk is not identified at this stage, it may be omitted from further analysis. This may result in material risks being given insufficient attention at a later stage. In algorithm assurance, material risks are often hard to pinpoint, as these often originate from the *blackboxness* or lack of transparency of the technology itself, but materialize as risks

[2] A risk management term referring to the explosion of the Piper Alpha oil platform in 1988, due to a small risk ignored by the entire industry sector. The burning platform situation creates a sense of serious urgency absent before.

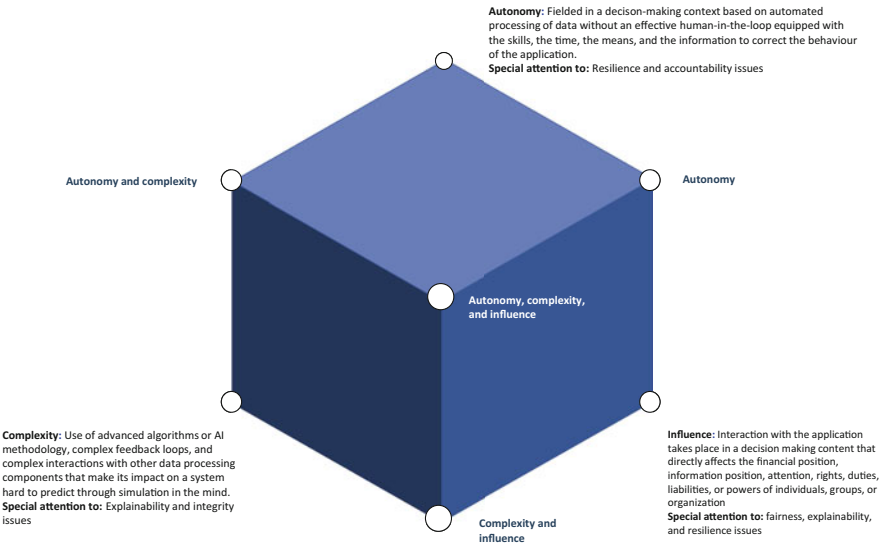


Fig. 1 Dimensions of risk and points of attention

in all kinds of other contexts. Common risk factors that relate to the deployment of algorithms may, roughly speaking, be grouped into three dimensions:

- Complexity
- Autonomy
- Impact

If the algorithm has a presence on all three dimensions, and on one of these dimensions can be considered high risk, it is likely to become a target for review or audit at some point for some reason. In Fig. 1, we show the three dimensions in the form of a cube. An easy way to convey risk profiles is scoring the application on each of the three dimensions and drawing a plane through the cube connecting the three selected points. At the axes we directly relate these risk dimensions to the five *control objectives* we use for our work: integrity, resilience, explainability, fairness, and accountability.

The first of these three dimensions is the *complexity* of the technology, of the task, and of the information ecosystem it operates in. In essence it relates to what is in the media often called *blackboxness*: the technology or information ecosystem is complex if it is hard to imagine simulating what it does in your mind and—importantly—if it is hard to recognize errors and hard to understand the cause of the errors it makes through simulation in the mind. Complexity can in this sense be seen as a dual of *explainability*, a concept that has been gaining in popularity in AI literature.

Complexity need not be directly related to the computational complexity class of the calculations made by the algorithm, or the complexity of the input data structure. These do definitely contribute to complexity: a deep learning-based algorithm will typically be considered more complex than a linear regression, and a linear

regression on many input parameters is more complex than a regression of a few parameters. But it is more often than not rather the complexity of the task to which they are set which is at issue. Facial recognition is for instance undoubtedly both computationally complex and based on a complex input data structure, but is often not seen as problematically complex. This is because the task—recognizing a face based on examples of that face—is not one we as humans usually consider complex. We appear to have an inborn talent for it, and we can often easily judge errors.

The algorithm may however still make errors that we would never make. Face recognition systems are for instance commonly fooled by holding a photo in front of your face, and that may be a fundamental flaw for the execution of the task to which they are set. For instance, if the face recognition unlocks a phone. The algorithm does what it was built to do: It recognizes the face. It is just not suitable for the complex task to which it was set. The task in this case turns out to be just a tiny bit more complex than the algorithm can reliably handle.

The second dimension is its *autonomy* in decision-making. The algorithm operates autonomously if it essentially functions without effective human oversight and its errors are likely to go undetected, unexplained, and unremedied. The face recognition phone lock scores high on these aspects of autonomy as well. Its user will be aware of false negative errors, when the phone does not unlock in the user's presence. The false positive error, unlocking without the user's presence, will go unnoticed. A last important aspect of autonomy is the algorithm's ability to autonomously adapt its behavior during its operational life by learning from its experiences without expert supervision. In general, this is a rare ability, but the face recognition phone lock has this ability as well. It learns to recognize its user without oversight by an expert, and without a formal validation process.

The third dimension is *impact*. Impact is determined by the characteristics of the task it performs. Impact is what is determined in an impact assessment, and is usually closely related to the motive for requesting an audit. It is material risk in the narrow sense: For instance, does the algorithm affect people's legal positions (it changes or establishes rights, duties, liabilities, etc.)? Does it handle money or valuable, private, or confidential information? Does it affect many people? Is it capable of abusing market power? The face recognition phone lock scores high on this dimension as well, because it may after all give access to all functions the user is authorized to access using that phone, including for instance banking and other functions based on authentication by phone.

Algorithm assurance differs from many other forms of assurance mainly on the impact dimension. A cybersecurity audit or an IT audit in the context of a financial statement audit is clearly scoped by a category of impacts on which the audit is focused. Algorithm assurance on the other hand focuses on the entity to be audited itself, and may cover a wide variety of impacts. Because algorithms may be set to any task, identifying its impacts requires some creativity from the auditor.

For governance functions scores on the three dimensions gain quick insight in the degree of attention an algorithm deserves, and what kind of risk mitigation needs extra attention. Complexity requires transparency and explainability, autonomy requires oversight, and impact requires explainability—because important decisions

must be justifiable—and impact-mitigating measures. As usual, everything starts and ends with the integrity of the implementation and application. If the algorithm doesn't effectively do what it is claimed to do, risk mitigation will not save us.

2.2 Algorithm Task Environments

Algorithms may be set to any task, and equally important, in any task environment. To get an overview of the field, we list some examples of categories of algorithms one may encounter in an algorithm audit.

A variety of algorithms are used for *financial prediction models*. These are commonly encountered in support of the financial statement audit, as they often have a direct effect on the financial statement. Technology used may vary from supervised machine learning to rule-based prediction models based on expert opinions, and hybrids of these. Typical issues are integrity and performance optimism, and less often gaming-the-system risks. The risk these algorithms pose mainly derives from complexity and impact on the financial statement. Compliance concerns relate to financial reporting regulations.

Supervised machine learning algorithms are typically used for *prognostic and diagnostic medical devices*. Applications range from prognosis of aggression by mental health patients based on non-invasive monitoring of vital signs to diagnosis of diseases of the retina using a high-quality camera. Typical issues are privacy and medical ethics concerns about data collection for training and testing the algorithm, equal performance on ethnic groups and genders, and presence of effective monitoring to check that actual use follows intended use. Compliance concerns relate to medical device regulation and regulation on medical ethics research involving human beings. Because decision-making is usually left to medical professionals, complexity of the algorithm is usually more of a concern than autonomy.

A variety of algorithms are used for *risk-based selection on applications* or claims to select suspicious applications for in-depth manual processing. Non-suspicious cases are then handled automatically. Technology used may vary from supervised machine learning, unsupervised machine learning (outlier detection or clustering when accurate training data for supervised learning is scarce), or rule-based prediction models based on expert opinions. Typical issues are differential treatment of groups based on static descriptors (profiling or discrimination), indirectly leaking sensitive data about individuals, and gaming-the-system risks because customers have reasons to game on ending up in the automatically processed or "happy" flow. Applications are for instance found in insurance, banking, policing, and taxation, and compliance concerns are often related to privacy and human rights. When operating on very large data streams, autonomy of the algorithm is a serious concern.

A variety of algorithms are used for *automated trading systems*, varying from basic robotic process automations for handling simple purchases or payments to high frequency, high volume flash trading of derivatives, to bidding agents for ad space. Technology used may vary from supervised machine learning to rule-based

prediction models based on expert opinions, and hybrids of these. Typical issues relate to intended use, oversight, and gaming-the-system risks. It is mainly the autonomy of the algorithm that is at stake. These systems may come into scope of the financial statement audit. More rarely compliance concerns related to for instance market manipulation (MIFID II) play an important role.

Unsupervised algorithms are often used for *clustering unstructured text into topics* to improve access to large corpuses of text. These texts are sometimes anonymized. A typical issue in this type of application is re-identification risk in anonymized corpuses based on the propensity of algorithms to cluster texts written by the same author together. Gaming-the-system issues may play a role as well. The leading compliance concern is generally privacy. The algorithms involved are usually just complex.

Recommendation algorithms for products, music, films, etc. usually involve a hybrid of reinforcement and unsupervised learning technology. Typical issues are differential treatment of groups based on static descriptors (profiling or discrimination) and gaming-the-system risks because suppliers of the products being recommended have reasons to game on ending up in recommendations. A less common compliance concern is self-preferencing by the organization running the algorithm if it acts as a supplier itself, which can be seen as an anti-competitive behavior by its business clients. Recommendation algorithms tend to be sensitive to *cold start* problems and *popularity bias.* Extra care needs to be taken when they are first deployed to mitigate these risks. These algorithms score high on autonomy.

A variety of algorithms are used for *profiling* and *ad targeting*. Hybrids of supervised, unsupervised, and reinforcement learning are used. Common issues in ad targeting is differential treatment of groups based on static descriptors (profiling) and indirectly leaking sensitive data about individuals. Compliance concerns are generally privacy and differential treatment of groups based on static descriptors (profiling or discrimination). Ad targeting business often also includes automated trading for advertising space.

The list of example task environments provides context to the rest of the chapter, but in the rest of the chapter we will limit ourselves to a single example task.

3 Running Example for This Chapter

As a detailed running example for this chapter to illustrate choices made in the audit, we introduce a public body that processes applications for child benefits. The public body does not have the manual processing capacity to investigate every application. Ninety-five percent of applications are processed automatically, following the claims made on the application form. In the vast majority of cases, this leads to an acceptance. In some cases, applications are directly rejected on formal grounds. Five percent are processed manually and claims are investigated in detail. Discretionary manual investigation can take anywhere from 5 min to many hours, often weeks in real time, leading to a final accept or reject decision. Manual investigation

can involve contacts with the applicant and third parties to collect additional information. If intentional noncompliance is suspected, the case may be handed over to a special investigation unit that will decide whether a report should be filed with the police.

The public body has a policy of picking applications for manual processing based in noncompliance risk. To help with this risk assessment it has introduced a supervised learning algorithm in the category of *risk-based selection on applications*, that selects risky applications based on historical information from applications manually processed in the past. The risky applications are automatically sidelined for manual processing. The algorithm will be retrained yearly, suing the new data generation by manual processing.

Processing takes place in the context of the GDPR. Based on specific administrative law about child benefits, the public body does however have special permission to process sensitive information about natural persons if this data is required for making decisions, and to collect additional information from third parties like banks, townships, or schools. The public body does however feel very vulnerable to scandals about unfair treatment based on sensitive attributes and has therefore decided to have the risk-based selection algorithm regularly audited so that it will be in control if a scandal would develop.

Because benefits will only be awarded if the parent takes care of children the majority of the time, child benefits usually go to the household where the mother is present (English, 2021). This leads to an increased likelihood that the historical data may be biased against single fathers and that this affects the algorithm. In addition, the rules about what is and what is not allowed have regularly changed over the last decade. Because it is clear that the historical data has been collected over a period in which the rules regularly changed, and presumably will keep changing, there is a risk that the algorithm is not as accurate and reliable as performance measures may suggest for the groups affected by the changes.

4 Scoping an Algorithm Assurance Engagement

In the previous section, we have introduced a model (see Fig. 1) with the three dimensions *complexity, autonomy, and impact* to determine if an algorithm is likely to become a target for review or audit. Especially if an algorithm is in its context perceived as impactful, the need to be assured of its reliability grows. In this section, we will discuss how to scope an algorithm assurance engagement by understanding the algorithm's context and the audit objectives, and how the context and audit objectives set the criteria that form the basis for the risk assessment.

4.1 The Importance of Understanding an Algorithm's Context

In any larger, more complex, social setting, algorithm assurance should not only focus on the (technical) properties of the algorithm itself, but also on its purpose as a problem solver in the real world. A standalone algorithm without task environment is not useful, but as soon as it is put into a complex task environment to perform highly impactful tasks, the things that can go wrong are countless. For the auditor, to comprehensively understand an algorithm in its context is crucial in order to start scoping an algorithm assurance engagement. The definition of an algorithm's success is in the end whether it is fit for purpose in the task environment in which it is embedded as a decision maker or decision support system. This purpose and the required skill level determine the technical requirements on the solution. In many cases, a traditional IT system will suffice, because most problems are relatively easy to solve. Only when the definition of success requires a more advanced type of solution due to the complexity of a real-world problem, the implementation of an AI algorithm should be considered. A computer program, or component of a computer program, that contains implementations of a risky algorithm or algorithms, is to be reviewed in the context of a task in which it is applied or a prospective class of tasks in which it may be applied. Figure 2 shows how traditional IT systems and advanced algorithms are often combined to work towards a single decision. In such situations, solely auditing an algorithm itself would make no sense.

Understanding the context of an algorithm requires an assessment and detailed understanding of a range of broader social and political facts about its stated definition of success. Typically, the context of an algorithm includes the process

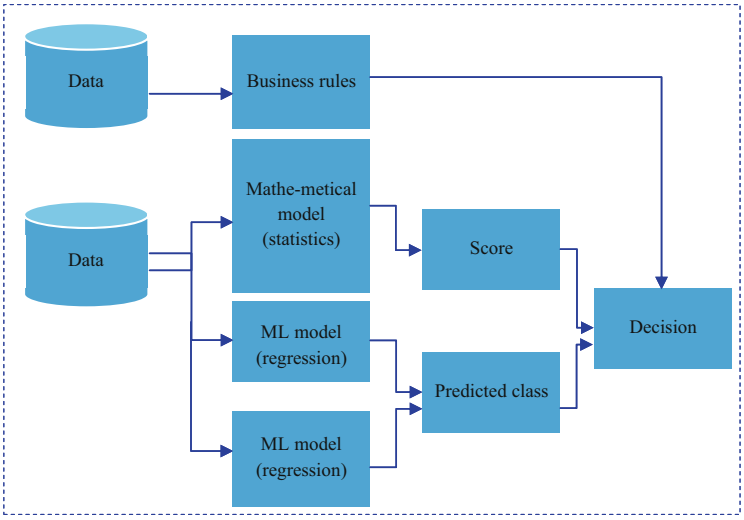


Fig. 2 Algorithm-based decision-making

of development of the algorithm, the process of preparing the data for training and testing the algorithm, the process of delivering an algorithm to its primary user, and often, most importantly, the setting within which it is used (Brown et al., 2021). To understand the algorithm's context and to take a first step in reviewing the algorithm itself, an important distinction needs to be made between a claimed skill and a claimed capability. Capability reflects the general problem-solving potential of the algorithm itself centered on accuracy and reliability claims, for a variety of tasks for which it could be fielded as a solution. Skill reflects the actual performance on a task in a specific task environment, including impact and autonomy aspects, and including risk-mitigating measures taken to control the task environment. An algorithm that works well in the Amsterdam office may not work in the Rotterdam office if the Rotterdam office lacks certain risk-mitigating mechanisms.

When we consider our running example again, the algorithms' definition of success is simple: detecting noncompliance. Incorrect applications are considered as a given, and the goal is to determine whether these applications are incorrect by accident or deliberate. The difference between accidentally or deliberately incorrect applications is of crucial importance in the context of this algorithm, because for mistakes made by accident the algorithm has no reason to create a signal. As a system for—essentially—fraud detection, compliance criteria and fairness criteria as typical issues for this type of fraud detection algorithm are differential treatment of groups based on static descriptors (profiling or discrimination). Consider how different it would be when a same type of algorithm is used with the purpose to identify incorrect applications to help citizens to better apply for subsidies? In that case, the definition of success would be entirely different and so are the relevant criteria to review.

4.2 Assurance Criteria

Over the past few years, many non-commercial and commercial organizations have issued principles for trustworthy AI. The EU High Level expert group for example, put forward a set of seven key principles that AI systems should follow in order to be deemed trustworthy (European Commission, 2019). Google as well introduced seven principles, and a complete audit framework for algorithms (Raji et al., 2020). Although these principles are to a certain extent similar, there are some notable differences. The EU stresses the importance of privacy and human oversight, while Google also finds it important to use AI only in alignment with scientific evidence.

If we consider how assurance engagements on other types of IT systems are currently carried out, the concept of overarching principles applies as well. The so-called *trust services criteria* (Ewals et al., 2019) are used as means to assess the extent to which an organization has controls in place to let IT systems operate in correspondence with the criteria.

Table 1 Overview of SOC2 trust principles, EU working groups, and coherent audit research questions

SOC2 trust principle	EU working group	Audit research question
Security	– Technical robustness and safety	– Can the data used by the algorithm be accessed by unauthorized individuals?
		– Are there risks of gaming the algorithm?
Availability	– Technical robustness and safety	– If the algorithm is business critical: how is its availability and contingency managed?
Confidentiality	– Privacy and data governance	– May the output of the algorithm lead to the identification of (protected) subgroups?
	Transparency	
Processing integrity	– Human agency and oversight	– Does the algorithm perform in line with its definition of success?
	– Accountability	– Is the algorithm fair and unbiased in its specific context?
	– Diversity, non-discrimination, and fairness	
	– Societal and environmental well-being	
Privacy	– Privacy and data governance	– Are there sufficient legal grounds to use the algorithm?
	– Diversity, non-discrimination, and fairness	

From an algorithm audit perspective, there are reasons to argue that such trustworthy AI principles are a good basis to scope an algorithm audit. This is because these principles provide a specific perspective, a set of control objectives appropriate for AI assurance, for an auditor to focus on. There is also reason to argue that the already existing trust services criteria are insufficient, because algorithm assurance should not only focus on the algorithm itself but also on the context in which it is being used. If you try to map the SOC2 trust services criteria to the AI principles of the EU working group, no exceptional creativity is required to successfully make it fit.

In an algorithm assurance engagement, the auditor should combine the auditees requirements with the context of the algorithm to select the appropriate criteria. We also provide some example audit questions that should be answered satisfactorily depending on the selected criteria (Table 1).

The auditee, or the client authorized to request the audit, may have its set of control objectives to be audited. The audit report should be relevant to its audience, after all. Business sectors moreover usually operate within a framework furnishing relevant assurance criteria as well. Various high-risk sectors, ranging from the financial, automotive, and health sector to the trade in children's toys, have, or will develop, guidance for using AI for high-risk functionality. If you are auditing a

medical diagnostic or prognostic application, for instance, there will be guidance that can be followed interpreting Medical Device Regulation regulations (e.g., there is a guidance for medical diagnosis in the Netherlands (Van Smeden et al., 2021)). Besides that, there will usually be a number of ISO/IEC standards to take into account. Sector-specific jargon and perspectives cannot be avoided, and over time algorithm assurance will require the development of a certain amount of sector specialization guided by scientific contributions (e.g., Wirtz et al., 2022).

Coming back to our running example of our algorithm to select applications for child benefits for manual processing, we argue that *diversity, non-discrimination, and fairness* would be the most relevant audit criteria. In this case, it would mean that the audit team will for example need to determine that the algorithm is unbiased against all protected groups. In addition, fairness is also about weighing the legitimacy of the task the system executes, how well it does at performing that task, its use of personal and sensitive data, and the quality and representativeness of that data for the task it performs. Assurance on diversity, non-discrimination, and fairness is therefore based on presumptions about technical robustness and safety and accountability. These should also be part of the audit team's investigations. Moreover, the targeted readers of the audit report are clearly citizens, politicians, journalists, and potentially a court of law. Having a good explanation of what the algorithm does is essential to risk mitigation. Investigating transparency is therefore unavoidable as well, even if the reported findings are about diversity, non-discrimination, and fairness.

There are two key differences between SOC2 assurance and algorithm assurance. Firstly, SOC2 criteria are formulated in a very generic manner, while in algorithm audits specific controls aligned with the algorithm's context and associated risks are crucial. Secondly, SOC2 follows the COSO-framework, which is extensive but in practical terms leads to audits that are fully focused on control testing only. In an algorithm assurance engagement, we argue that control testing only would fall short to be able to provide enough comfort about the algorithm working in alignment with the selected criteria. A typical audit approach for control testing is required to be augmented with other types of audit approaches such as testing the model itself or a form of substantive procedures. In the last section of this chapter, we will propose four of such approaches.

4.3 What Do the Trust Services Criteria Apply to?

In regular IT audits, one or a combination of the following components are assessed against the Trust Service Criteria during a SOC2 examination: Infrastructure, Software, People, Procedures, Data. In algorithm assurance, we argue that the scoping exercise in terms of (technical) components is subordinate to the importance of how an algorithm has been implemented in its context. Typically, we believe that the audit or review of an algorithm would focus for a large part on the steps that were carried out by the team that builds the model, instead of all the individual

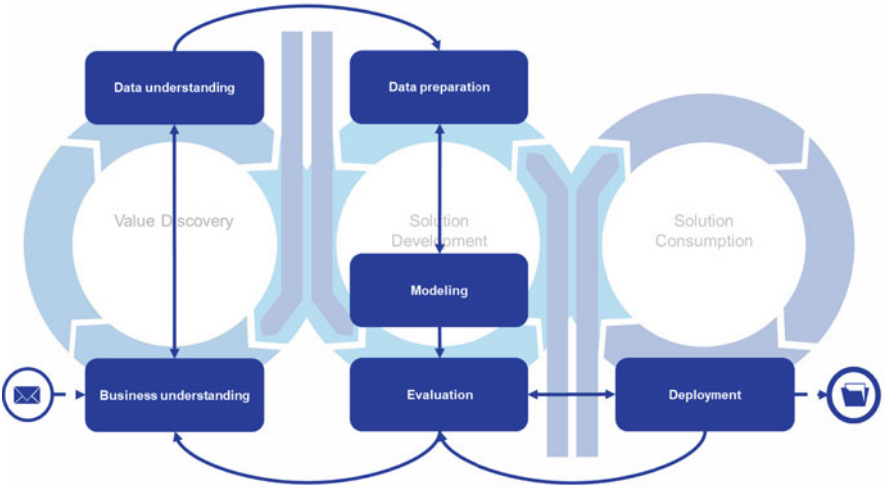


Fig. 3 Spheres of activity where risk and control play different roles

components of an algorithm and how they exactly operate. As described in Sect. 4.1, next to the setting in which an algorithm is used, it would also include the process of development of the algorithm, the process of preparing the data for training and for the process evaluating the algorithm, and the process of deploying an algorithm in its task environment. And finally, the central issue of developing a good problem conceptualization, which should be based on a realistic data understanding and business understanding. Generally speaking, we distinguish three different spheres of activity in the life of an algorithm (see Fig. 3). Each phase requires a different perspective on dealing with risk and control.

To further illustrate how the process of developing an AI algorithm is important, we return to our running example. When building a supervised learning algorithm that is aimed at identifying noncompliance, a common issue is the number of false negatives. As many noncompliant transactions will go unnoticed, the labeled data that is required to build a supervised learning algorithm is going to be extremely biased towards learning about true and false positives. It doesn't come as a surprise that in banks for example, unsupervised learning systems are favored for fraud detection over supervised learning algorithms to tackle this problem. Assuming that the developer in our example is aware of this general issue with fraud detection algorithms, there must be reasons why supervised learning was still preferred over other type of models. The relevant question to ask as an auditor is: How did the developer come to this decision, and what steps were taken in order to discover the false negatives for which no outcome of manual processing is available. How the developer has coded its model and what frameworks were used is considerably less important.

4.3.1 An AI Model's Technical Architecture

AI algorithms are often hidden behind user interfaces, web services or in software components. There is no one typical AI architecture that is common across all AI capabilities. If we browse online through the setups that are disclosed by companies or third-party vendors, we mostly come across an overview of relevant platforms, frameworks, and supporting tools during the development and deployment cycles of algorithms only. Each year Firstmark[3] publishes an overview of all relevant vendors in the ML and AI business in the so-called Machine Learning, Artificial Intelligence, and Data (MAD) Landscape. The overview distinguishes high-level categories to show what is available in the marketplace. The MAD Landscape shows a myriad of vendors arranged by type of services, ranging from infrastructure and data (re)-sources, to analytics and machine learning/AI platforms. For an auditor, it would never be possible to build the required expertise to appropriately assess all the hundreds of different products available on the marketplace.

The audit team should limit itself to the development process instead of the specific platforms, frameworks, and tools to perform AI and Machine Learning tasks. Uber, the taxi and food delivery company that is well-known for its advanced AI deployments, provides some guidance in this regard. The description of Michelangelo, their Machine Learning platform, is based on the steps taken in the machine learning lifecycle[4] instead of the technical architecture: manage data, train models, evaluate models, deploy models, make predictions, monitor predictions. Another common model that is used to lay out the AI development lifecycle is the CRoss Industry Standard Process for Data Mining (CRISP-DM),[5] which also forms the basis for our previously presented Fig. 3 on spheres of activity where risk and control plays different roles.

4.4 Stakeholders in the Audit and Accountability

As part of the criteria, we identified accountability as one of the key aspects to look into. The assurance engagement should be *scoped* towards the risks that matter to the client, depending on the purpose of the engagement and the algorithm's context.

An algorithm assurance engagement may be motivated by internal risk management needs of the engagement client, reporting obligations to supervisory authorities, the risk management needs of one or more third-party stakeholders in the decisions the algorithm takes or supports, or a general need for transparency towards society. The risks that need to be focused on are determined by the motive for the engagement.

[3] https://mattturck.com/data2021/

[4] https://eng.uber.com/michelangelo-machine-learning-platform/

[5] https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

An important aspect to scoping the problem is whether the assurance client is a provider of the algorithm, a user of the algorithm, or both—in case an algorithm developed in-house is used. This is an important question from an accountability point of view since the provider and user have different responsibilities. The provider needs to provide something that will work well *if* the manual is followed. Assurance is in this case mainly about consistency between claims about the algorithm and their substantiation by the algorithm *if* it is used correctly. The user needs to follow the manual: any deviation from intended usage is a relevant finding, and potentially a source of additional risk.

4.4.1 Accountability of Cloud Providers

Most companies these days use some sort of cloud computing to reap the benefits of AI. For many companies Uber's approach to set up an end-to-end platform from scratch is unrealistic, because of the required investments and the scarce knowledge that is required to set up such a platform. Therefore, most companies turn to the larger cloud vendors such as Microsoft's Azure, Amazon's Web Services, and Google's Google Cloud to work with off-the-shelf learning algorithms. For the auditor these larger vendors remain an almost insurmountable obstacle, as they typically try to avoid to contractually agree on a right to audits. In these situations, the process approach helps to limit the reliance on the work done by the cloud providers. It is increasingly common to depend on ISO/IEC 27001 and 27018 certifications from cloud service providers.

5 Risk Assessment

In Sect. 2.1 of this chapter, we introduced a simple three-dimensional risk model and classification method for determining whether an algorithm is a suitable candidate for algorithm assurance. In practice, the algorithm rarely scores as high risk on all three dimensions of the risk model, because the presence of clear risks on two of these dimensions typically leads to lower risk choices on the third dimension. The risk classification method does not replace a true risk assessment. It selects candidates for a risk assessment. In this section we introduce a risk assessment method based on identifying risk likelihood drivers and impact drivers in the task environment. We also discuss the need for a diverse audit team composition.

5.1 Drivers for Likelihood and Impact

Identifying the key risks an algorithm poses to the company is a critical step in effective risk management. This step needs to be comprehensive. If a potential risk is

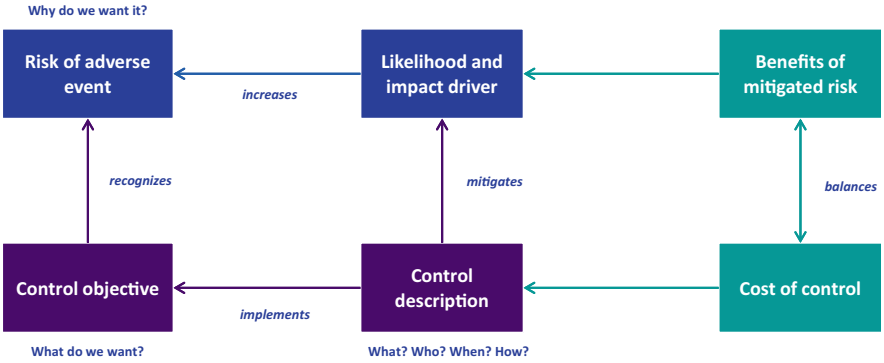


Fig. 4 How control objectives, risks, and likelihood and impact drivers relate to each other

not identified at this stage, it may be overlooked during further analysis. This may result in material risks being given insufficient attention at a later stage. In algorithm assurance, material risks are often hard to pinpoint, as these often originate from the *blackboxness* or lack of transparency of the technology itself, but materialize as risks in other places.

In Fig. 4, we relate the ingredients of our approach to AI Assurance to each other. The *risk* you take with an algorithm is your exposure to loss or damage caused by *adverse events* involving the algorithm. Which events you consider adverse events is determined by your *control objectives* (like the aforementioned seven AI Ethics principles). A *likelihood driver* is a circumstance (in the task environment, or during the conceptualization of development phases in Fig. 3) that increases the probability of adverse events happening to the algorithm. An *impact driver* is a circumstance that increases the impact of adverse events, usually by enabling additional adverse events to happen to people, processes, data, etc. *Controls* mitigate for the circumstance that increases the probability or impact of the adverse event happening to the algorithm. Generally, the point of risk mitigation processes is:

- To create awareness of likelihood and impact drivers present in the environment of the algorithm
- To select and implement controls that reduce the total amount of risk to an acceptable proportion
- To periodically check the continued presence and operation of the controls

For most auditors, likelihood drivers and impact drivers will sound new. Typically, a risk assessment is carried out in terms of likelihood and impact only. In algorithm auditing specifically, likelihood is often replaced by complexity, suggesting that if a model is more complex automatically its risk profile rises. We argue that this equivocation is far too broad and simple. An algorithm's context is much more decisive for its risk profile than its complexity, and combinations of factors constitute risk. A three- or five-point scale from low to high is used to build a risk profile. We believe a solid risk assessment should take it a step deeper

considering factors contributing to the likelihood or impact of adverse events. Since risks factors involving algorithmic bias often form mechanisms that can be expressed in the form of causal loops, we recommend to, where appropriate, assess the drivers in the form of causal loop diagrams or a similar diagramming technique.

The context of the algorithm, in combination with the control objectives you committed to, determines what the relevant adverse events are. When doing the risk assessment, the auditor should hypothesize what outcomes are to be considered as irregular in relation to the algorithm's normal performance and behavior. In general audit terms, these adverse events are often referred to as what-could-go-wrongs. These must be reduced to acceptable proportions using controls. Acceptable risk relates to the cost of control: Controls usually have a cost, and that cost has to be balanced in practice against the risk mitigation benefits of the control mechanism.

In our running example of the public service organization selecting applications for manual processing, we can also make a distinction between likelihood and impact drivers. As mentioned earlier in this chapter, supervised learning algorithms used in fraud detection are typically known to be very susceptible for their lack of ground truth. Because typically only the fraud that meets human expectations is discovered, other types of fraud are not identified and therefore the data only shows parts of the truth. This ground truth issue clearly classifies as a driver on likelihood: lack of representativeness of the available training and testing data for the data that the algorithm receives as input (including all the false negatives) directly contributes to the risk that the algorithm, and its evaluation, will be inaccurate. The organization *could* control for that risk through random sampling for manual processing, but searching manually for the false negatives is going to be very costly in man hours and this cost of control may be at odds with the business case for the algorithm.

The purpose of the algorithm is an impact driver: because the outcome of the process directly affects the legal and financial position of citizens, and citizens do not usually participate in that process for fun only. Even the delay caused by selection for manual processing may be considered unfair.

The possibility of bias against single father household applicants is a typical adverse outcome in the fairness category. Because benefits will only be awarded if the parent takes care of children the majority of the time, child benefits usually go to the household where the mother is present (English, 2021). There is a clear likelihood factor present: likelihood that the historical data may be biased against single fathers. This may affect the algorithm. From a risk assessment perspective, the auditor (and public body) should take into account that the impact of an accusation of algorithmic unfairness may be considerable. Single fathers may generate a lot of attention and sympathy in the media, and differential treatment without a good justification may be considered a human rights violation in court. Impact may be considerably reduced by having a good explanation ready at hand for the media for any apparent differential treatment.

5.2 A Standard Set of Likelihood and Impact Drivers

A comprehensive risk assessment of an algorithm highly depends on the context and the real-world problem. AI algorithms are associated with risks that capture the public imagination, and stir the interests of regulators: deanonymization, profiling, unfairness to protected groups (discrimination), surveillance, restriction of freedom of speech, gaming the system, hampering competition, disturbing public order, abuse of markets, and abuse of information position. Financial risks often relate to the costs of reparations: manually re-doing processed cases, litigation costs, fines, damage, loss of reputation.

In the overview below, we present some examples of likelihood and impact drivers including a short description from our own risk identification inventory. By no means this should be perceived as an extensive list of algorithm risks, but it helps the auditor in the line of thinking to objectify the likelihood and impact of algorithms not operating in line with their definition of success (Tables 2 and 3).

5.3 Who to Involve in the Risk Assessment?

There is increasing consensus (Shen et al., 2021) on the relevance of involving a heterogeneous group of people in terms of cultural background, technical expertise, and domain expertise in the development teams of AI algorithms. By making people with a pluriform background part of a development team, the integrated team will be better at conceptualizing a real-world problem from different perspectives. Consequently, pluriform teams develop better AI algorithms with and diminish the likelihood of undetected risks. In the same vein, we argue that this line of reasoning also holds for algorithm auditing, and carrying out the risk assessment as part of it (Shen et al., 2021). Making sure that an audit team that performs a risk assessment represents the cultural and gender demographics of the stakeholders in the algorithms that they are auditing, major blind spots on stakeholder impact with potentially critical issues surfacing only post-deployment can be already identified during a risk assessment. Composing a heterogenous team is not always achievable, but making sure the audit team has a certain level of heterogeneity will actually help to assess an algorithm in its broad context.

6 The Audit Plan

In this section, we will discuss how to formulate an audit plan, how traditional tools and techniques from the auditor can be leveraged during an algorithm audit, and how AI-related skills play a crucial role to perform successful algorithm audits.

Table 2 Overview of impact drivers and rationale thereof

Likelihood drivers	Explanation
The predictions of the AI application cannot be adequately or timely verified by observation to measure performance.	For an AI application, you would like to know whether your prediction also came true. In some cases, this is not possible. For example, when the AI application predicted when something would break, but it is repaired before that specific date. Or whether a mortgage loan will be paid off, which is known only after 30 years.
All training and evaluation data originates from one specific task environment.	In case an AI application is designed in a specific environment, but is executed in a different environment, the outcomes might not be correct. For example, predicting what EU citizens would like to pay for a hotel based on Europe, but erroneously assuming this model will predict correctly for South America. Since the EU cannot be compared to South America, the model will likely not be generalizable.
Experts making the same decision with the same information report complex and diverse reasoning patterns for different cases that are hard to capture by the machine learning technology applied from a learning capacity perspective.	The complexity of the task environment is beyond the learning capacity of the algorithm employed. For instance, if you train an application to predict whether someone is ill, completely ignoring the fact that doctors distinguish a lot of different diseases with different underlying mechanisms. Better to train an algorithm per disease category, and combine these in a hybrid system. This type of application will moreover create huge explainability problems.
The risks involved in wrong predictions made by the AI application for downstream tasks are not adequately distinguished from the accuracy of predictions in performance measurement, leading to a conflation of accuracy and utility of the AI application.	Any abductive argument is uncertain, in the sense that you jump to a conclusion knowing you may be wrong. How tolerant you are of making mistakes depends on the value of the conclusion in tasks that functionally depend on it. This risk tolerance needs to play a role in the measurement of performance, but should not be implicitly mixed in with accuracy. Conflation means treating two distinct concepts—in this case accuracy and utility—as if they were one, which produces errors or misunderstandings as a fusion of distinct subjects tends to obscure analysis of relationships which are emphasized by contrasts. Very common mistake, for instance if the *F*-value statistic is used for performance measurement without consideration of risk appetite for false positives and false negatives, which is important for determining the utility of an algorithm.
The AI application operates in a task environment that requires complex interactions with	Certain failure modes may be easy to prevent for an individual agent, but may arise for a

(continued)

Table 2 (continued)

Likelihood drivers	Explanation
other software agents, consists of a complex combination of AI techniques or models, or is input to, or dependent on the output of, other AI applications.	combination of agents. Typical examples are market abuse (MIFID II rules) or algorithmic price cartels. Even though each individual trading agent keeps to MIFID II rules, all agents in the organization taken together may violate them. Similarly, one agent may simply be following market prices, a cluster of agents may form a cartel setting prices.

6.1 Audit Approaches

The aim of the audit plan is to formulate the required steps to perform the audit based on the approach that is the most feasible. We present four high-level approaches an AI auditor could follow to structure the audit plan. These approaches have a different area of focus and in practice will often be combined into an audit plan tailored to the case at hand (Table 4).

6.1.1 Approach 1: Evaluation of Algorithm Entity Level Controls

As part of this approach, the auditor shall evaluate at enterprise level whether sufficient entity level controls are in place to ensure algorithms are built and managed in a controlled environment. Controls in the area of AI strategy and policies, data governance, technology and platforms, skills and awareness, and development methodology should be part of the review. When only assessing a company's entity level controls, no direct assurance regarding the outcomes of an individual algorithm would be possible, but in general it may help to identify and assess overarching risks.

Algorithm entity level controls generally reduce the *risk of failure* for the algorithm and its outcomes, allowing for reduction of depth of testing (model test) or sample size (substantive procedures). An advantage to this approach is its feasibility. Testing entity level controls would only require traditional control evaluation procedures such as inquiry, inspection, and reperformance.

6.1.2 Approach 2: Testing the Model

As part of this approach, the auditor shall perform an in-depth assessment to determine if the algorithm performs in line with relevant audit criteria and whether the identified risks are properly mitigated. The approach to test an algorithm itself is generally speaking not too different from testing an automated control, because the initial focus would also lie on design and implementation. Still, for machine learning and AI algorithms (i.e., not rule-based models), the test of design is fundamentally

Table 3 Overview of impact drivers and rationale thereof

Impact drivers	Explanation
The decision made by the AI application significantly or irreversibly affects the interests or legal position of people.	The decision the system takes can affect legal position, financial position, or emotional interests. For example, rejecting to pay out a claim or to give the person a mortgage, award custody over children, infringe on people's privacy, stigmatize them, etc. Basically, anything that may drive people to court, causing damage to the organization.
The AI application takes decisions fully autonomously, without or only with pro forma supervision by people.	The decision made by the AI application is final and in practice not reviewed by a human. Adverse events may go unnoticed for some time, causing damage. This is most common for system that takes decision with a high frequency, like trading and recommendation systems.
Unfairness extends specifically to a subpopulation defined by a legally protected attribute (like ethnicity, gender, religion, etc.) that is required to be protected in that task environment.	AI application outcomes could be unfair to a subpopulation defined by a legally protected attribute. For example, the outcome could be unfair to women, giving them a lower chance of getting invited for a job interview. Presence of this driver increases the chances of other damage, and the organization may violate its own ethical principles.
The adverse outcome causes significant reputation damage.	The use of the AI application can cause significant reputation damage when certain adverse events happen. This depends on the presence of other impact factors, but also significantly on how visible the functioning of the system is to the outside world. A system that is open to outsiders for probing may for instance easily be tested for manipulations, or unfairness, and this increases the chance of reputation damage. We recommend carefully checking each adverse outcome individually!
The AI application handles or informs decisions about large amounts of money, or involves significant financial exposure.	The algorithm handles for the company a significant amount of money, for example a pricing algorithm for a significant account or revenue stream, for an online web shop or trading algorithm. Failure of the algorithm may lead to losses for the organization or other stakeholders. Note that this circumstance is relevant for financial assurance.

different from testing regular IT functionality. The key difference is that the logic captured in the algorithm is not specified up-front but is discovered from the training data during model training. Furthermore, the logic may evolve through time as a result of offline or online retraining and automated feedback loops. The assessment should therefore focus on the assumptions and design decisions that were made by

Table 4 A matrix of audit approaches with coherent focus area, the difficulty and feasibility of the audit

Audit approach	Focus area	Level of comfort	Feasibility
Evaluation of algorithm entity level controls	Overall algorithm control environment	Low	High
Testing the model	Algorithm design and maintenance	Medium to high	Medium to low
Testing monitoring controls	Algorithm output	High	Low
Substantive testing	Algorithm output	High	Low

the algorithm developers in conceptualizing the initial business problem into a formalized AI problem. Of course, the quality of the data and data preparation activities should also be in scope of these audit procedures. To test an algorithm's implementation the same types of test procedures as in regular IT audits can be used as a starting point, although some types of procedures may be less applicable or feasible, depending on the characteristics of the algorithm. In the subsection on tools and techniques, we will go in more detail.

Testing the model can provide a high level of comfort, depending on the detail of testing. If for example advanced techniques such as algorithm replication are used, the level of assurance on the quality of the algorithm will increase, because it requires the auditor to independently reperform (part of) the algorithm's development process.

The feasibility of this approach depends heavily on the complexity of the algorithm and availability of data sets. For rule-based algorithms, feasibility is much higher as explicit business rules provide clear criteria to test.

6.1.3 Approach 3: Testing Monitoring Controls

As part of this approach, the auditor should test if the enterprise put internal controls in place to monitor the transactions performed by the algorithm and mitigate the risks of algorithm failure. Essentially, this is a sort of black box approach focusing on the output of the model instead of its inner workings. Testing monitoring controls might be a preferred approach as it circumvents the complexity of testing the algorithm itself. However, this approach also has some drawbacks. Firstly, the implementation of algorithms may render traditional monitoring controls obsolete (e.g., controls involving comparison of employee performance are not possible if all employees are replaced by a single algorithm). The auditor should carefully assess if the monitoring controls are sufficient to mitigate the relevant algorithm risks. Secondly, monitoring if individual algorithm outcomes are correct is often not possible or feasible (unless for some rule-based applications or very trivial classification tasks like image recognition). We notice that controls aimed at directly assessing the quality of algorithm output are still rare today. Controls are more likely to monitor if data

distributions in transactions stay between predefined boundaries and identify outliers for manual follow-up.

The level of comfort provided by this approach depends on the type of controls and their goal. In case monitoring controls directly assess the quality of the individual algorithm transactions, high levels of comfort can be achieved. In all other cases, for example when monitoring is only done on aggregated figures, the level of comfort is much lower.

6.1.4 Approach 4: Substantive Testing

As part of this approach, the auditor should test if (a sample of) transactions were processed by the algorithm in line with relevant criteria. Similar to testing monitoring controls, substantive testing should be considered as a black box approach potentially leading to high levels of comfort. But potential issues are also to be considered. Firstly, it cannot easily be determined if algorithm output was correct or incorrect (or such information may only become available with a significant time lag). If such information was readily available, the algorithm would not be required in the first place. This severely limits the applicability of testing the reliability of algorithms through transaction analysis (in fact a form of black box testing). For example, for mortgage loans it takes 30 years before the predicted probability of default can be validated. Or for recruitment algorithms, the actual job performance of rejected candidates will never be known (setting aside practical problems related to object job performance evaluation). Secondly, depending on transaction volume a key issue with substantive procedures is that testing a significant number of transactions may be very time consuming. After all, algorithms are used to automate complex decisions not easily captured in simple business rules. And thirdly, due to opaqueness of the input-output relationships it is hard to determine if a sample of transactions provides sufficient evidence for the entire population (representativeness issue).

This approach provides a high level of comfort, as long as the sample that is tested is sufficiently large to properly represent the algorithm's performance. In that case, substantive testing gives high levels of comfort as the outcomes are directly tested per transaction.

6.2 Tools and Techniques

When the auditor has selected the most feasible approach, or a combination of them, there are multiple tools and techniques in the standard auditor's toolbox that can be used to perform the algorithm audit. In principle, the same types of test procedures can be used as in regular IT audits. Some types of procedures may be less applicable or feasible, depending on the characteristics of the algorithm. We discuss five types

of test procedures, which can be used in combination, to test the design and implementation of an algorithm.

Inspection Similar to regular IT audits, all the relevant documentation as output of the steps followed during development is reviewed. In case of an algorithm audit, the documentation should at least provide detailed information about the algorithms' definition of success and how it aligns with the problem conceptualization, the ways data exploration was done, how feature engineering was performed and how feature importance was measured, the configuration of hyperparameters, how overall testing and validation has been done, etc. Of course, this type of test procedure can only be used if the algorithm development and maintenance processes of the organization are sufficiently mature.

Reperformance On top of inspection, the auditor can also choose to reperform certain activities executed by the development team. For example, in case of supervised learning, the training phase can be reperformed using the same training/test dataset and the same parameters as the algorithm's developers to establish if this results in the same algorithm with the same performance (small differences may occur due to different random seeds). This type of test procedure requires specific expertise on part of the auditor and the auditee must be willing to provide the auditor access to the original data and an environment to train the algorithm.

Code review A code review on itself would never be sufficient to get the required comfort for algorithm assurance. Code reviews should therefore always to be used in combination with other testing procedures. The added value of code reviews is sometimes a topic of discussion, as in most algorithmic solutions the machine learning algorithm itself is not really implemented in readable code itself, but rather an off-the-shelf asset. Code reviews are especially relevant for custom code or scripts or if uncommon libraries are used.

Independent testing This type of procedure involves testing the algorithm using an independent dataset developed by the auditor. Independently testing an algorithm would require deep expertise about the specific technological details of the algorithm under review. The data set should be representative for the dataset that was used to build the algorithm, which can be a great challenge. But in scenarios where the impact of the algorithm is great, and the auditee demands a great amount of comfort, there just might be sufficient justification to use this type of approach.

Replicating functionality Just like for independent testing, replicating an existing algorithm's functionalities also requires deep expertise of data science and modeling. With this approach, a similar or more simple reference algorithm may be developed in order to compare the performance of the reference algorithm to the actual algorithm being audited. It highly depends on the type and complexity of the algorithm that is audited whether this approach is feasible. In addition, it requires the dataset for training/testing from the client to be available.

7 AI Skills and Expertise in the Audit

When the audit plan and specific procedures have been considered and planned, an assessment should be made what skills and expertise are required in order to successfully complete the audit. And although the depth of the audit may vary greatly and may even be very limited, it is important to have, next to a certain level of diversity, the right AI-specific skills and expertise in the audit team to spot and investigate potential problems. The audit team should be able to:

- Recognize unrealistic problem specifications that are not likely to result in safe algorithm use.
- Investigate the origins of the data to spot bias and quality problems in the data.
- Interpret and criticize the metrics used to justify the reliability of the algorithm.
- Perform an exploratory data analysis and interpret the output of common explainable AI (XAI) algorithms.
- Pick and use the right metrics for measuring fairness, and give the measurements a reasonable explanation.

7.1 Realistic Problem Specification

A key skill, maybe even the defining skill, of AI as a discipline is translating real-world problems into problem specifications solvable in information space using an algorithm for that class of information space problems. Bad quality algorithmic solutions generally start with a bad problem conceptualization. Starting from a good business case for an algorithm, a good problem specification operationalizes business performance in such a way that it can be measured and optimized, and clearly outlines the intended use of the algorithmic solution by setting out the conditions that must be met before it can be safely assumed to perform as claimed. The translation of key performance indicators that are relevant to business into measurable indicators for performance is an important source of error.

The auditor judges the documented problem specification for risks and for gaps—important criteria that remain unmeasured and unaddressed. A large part of the review of the solution itself can be interpreted as a comparison between what was specified and what actually happened during development and what actually happens in use. If the problem conceptualization is good, and the algorithmic solution is an optimal solution to the specified problem, and it is used as advertised, the algorithm will generally score well on the integrity pillar.

Let us at this point return to our running example and apply the measures of recall, precision, and *F*-score that were introduced in chapter "Introduction to Advanced Information Technology," Sect. 3.3 of this book. The public body uses precisely these measures to quantify performance and has trained the algorithm to optimize *F*1-score. The public body has decided before development of the algorithm, without argumentation, that an *F*1-score of 0.9 seems acceptable for

performance based on a quick search of $F1$-scores of some other projects, and the algorithm clearly exceeds that benchmark.

There are two fundamental problems here. The first one is the arbitrary benchmark. One should always use a benchmark that is relevant for the task environment. There is no objective answer to what is a good $F1$-score. It depends on the alternatives methods available for making a risk-based selection of applications. The F-score is moreover sensitive to class imbalance, or differences in ratio between the two outcomes in the historical data. Class imbalances vary over projects.

When you are developing a medical diagnostic algorithm, you can often uncover an appropriate benchmark for roughly the same task environment through study of scientific literature. There are after all many hospitals doing roughly the same things. The public body executes a unique task, and has no such option. It has two directions to move in to produce an empirically grounded benchmark:

- Try to create a golden standard dataset of correctly processed application forms and measure the performance of the manual processing department compared to this golden standard dataset. To produce this dataset usually involves assigning multiple employees to the same applications, and spending far more time on it. This may be prohibitively expensive. On the other hand, this golden standard dataset is also useful for researching bias in the historical data.
- Play structured games with employees of the manual processing department or decision makers to determine what distribution of true positives, true negatives, false positives, and false negatives they tolerate. This approach leverages expert knowledge effectively, assuming the employees involved do understand their business well.

The second problem is that $F1$-score as a balanced score of precision and recall weighs false positive selection and false negative non-selections equally heavily as errors. It is a harmonic mean, after all. This is very unlikely to reflect the actual business objectives of the public body. As noted, when we introduced the running example manual processing capacity is scarce, and selecting applications for processing needlessly is a waste of effort. Besides that the organization specifically fears unfairly selecting people for manual processing, and this risk only relates to false positives. It should therefore be concerned with precision much more than recall when measuring performance. Fortunately, it is quite easy to modify the F-score to take a certain exchange rate between recall and precision, to reflect that employees would trade for instance five false negatives for one false positive in a structured gaming situation.

$$F\frac{1}{5} = 1 + \left(\frac{1}{5} \cdot \frac{1}{5}\right)^2 \frac{\text{precision.}\ \ \text{recall}}{\left(\frac{1}{5} \cdot \frac{1}{5}.\ \ \text{precision}\right) + \text{recall}}$$

This generalized F-score can be used for plotting precision against recall for an algorithm's performance to gain insight into what task performances are feasible depending on a chosen exchange rate between precision and recall. For a given task

environment, with an already determined exchange rate, only one point on the curve is important.[6] But the developers of the algorithm often do their work not knowing what that point is going to be.

7.2 Data Lineage

Whether a machine learning solution may be expected to do what it is claimed to do depends considerably on the fidelity with which the training and test data used for its construction reflects the task environment in which it is fielded. When we are forming an opinion about the usefulness of training and test data for an algorithm, we are looking for signs of lack of representativeness of the dataset for the task environment, and for signs of systematic misrepresentation of what actually happened in the task environment in the dataset. The first type of problem is an (inductive) bias problem. The second type is a data quality problem.

The concept of bias is widely applied, to describe (1) lack of representativeness of datasets for an environment, (2) the causes of that lack of representativeness (reporting bias, survivorship bias), and (3) the consequences of that lack of representativeness for decision-making based on the algorithm's output (popularity bias, algorithmic bias, and—as a convenience label—for any unfair decisions caused by bias). Here we limit ourselves to bias as a property of a dataset in a task environment.

If the algorithm used belongs to the class of supervised algorithms, it is trained and tested with data labeled with the (putatively) correct answer. The most obvious technique for researching bias is to compare data used for training and testing with the remaining unlabeled data, for which no correct answer has been determined, in an *exploratory data analysis* or EDA. Judging and performing an EDA is therefore part of the desk research skills one would expect of an audit team. Systematic differences found are in need of an explanation.

The auditor will in addition investigate and sometimes test the processes that created the data to gain insight in bias and quality problems and their causes. Part of these processes—from the master datasets that were sourced for the development process to the datasets that are fed into the algorithm—are under direct control of the developers of the algorithm. This is the data preparation pipeline. The pipeline should be documented well enough to allow for reperformance by an audit team. Bias and quality problems are however often already present in the master datasets that were sourced for development. At some point the audit team will be investigating where this master data came from.

At this point we run into an important scoping question. There are basically two ways in which the lineage of these datasets may be proven (Cheney et al., 2009). In *eager lineage* settings, the data is well-governed and the characteristics of the

[6] A very similar curve, containing similar information, is the ROC curve which plots recall against the true negative rate. This type of curve is more often encountered in documentation.

processes that created it are already routinely well-documented by the data controller. One may for instance expect this in medical settings. Data gathering is supervised by a medical-ethical authority, data management plans will be in place before gathering starts, and the process will be subject to an audit regime. In this case we would have an independent party assuring us of the quality and representativeness of the data. In *lazy lineage* cases research into business practices generating data had to take place within the context of the development of the algorithm because no such assurance already existed. In this case lineage should be fully documented as part of the development process and is clearly subject to investigation by the auditor in an algorithm assurance engagement.

7.3 Reliability of Trained Models

The auditor should understand empirical approaches to determining the reliability of a predictive model through resampling methods, and if necessary, should be able to apply them to the data. The most basic method for estimating performance is a train-test split. This gives us performance statistics, but no insight into how robust that statistic is going to be on new data. Validation of performance should take place on holdout data that was not available to the developers. Ideally the holdout data would be produced in an empirical impact study that is an exact simile of the prospective task environment.

Without access to new data, robustness of the algorithm can still be estimated by the developers and serves an important purpose in itself. The standard approach to showing reliability is to essentially make a lot of randomized train-test splits (cf. resampling methods like cross-validation; Kohavi, 1995). The average and variance of the performance statistics collected in train-test splits gives insight into the reliability of the model—assuming that the data reflects the task environment in which the algorithm will be used.

In addition, it is good policy to test any hypotheses one has about groups or time frames that can be found in the training and testing data in which the predictive model may perform less well to validate the problem specification, to ascertain there are no resilience problems to be expected (cf. so-called *underspecification* problems; D'Amour et al., 2020). One doesn't want to depend on an algorithm that doesn't work in winter, or doesn't work in Amsterdam. Measuring unfairness based on hypotheses about groups that may be treated differently is essentially a special case of this type of hypothesis testing.

7.4 Exploratory Data Analysis and the Use of Explainable AI (XAI) Techniques

While explainability can be considered a core goal of algorithm assurance, and we therefore favor transparent and self-explanatory algorithms, there are cases where either an alternative form of analysis is called for to uncover what the algorithm does, or where a parallel, more explainable algorithm with less performance is built to gain insight into the relation between inputs and outputs of a black box algorithm. The audit team is expected to understand exploratory data analysis and the use of common Explainable AI (XAI) techniques to uncover what the algorithm does. See for an overview of XAI techniques that can be used Linardatos et al. (2020) and for an understanding of the limitations of these as a tool for explainability cf. Lipton (2018). These techniques will occasionally be used by the audit team to gain the necessary insights and to explain its findings. Specifically, the audit team should be able to:

- Compare datasets collected from the same task environment.
- Apply feature selection and extraction methods to gain insight in the relevance of the data to the problem solved by the algorithm.
- Apply XAI methods for gaining insight into what role features play in how the algorithm solves the problem.

7.5 Measuring Fairness

Algorithm fairness is a hot topic, and for clients often a gateway into requesting algorithm assurance. It is moreover a central topic in our running example for this chapter. Making a judgment about fairness starts with identifying which groups or individuals may be differentially treated by an algorithm based on static descriptors. In a well-managed development process, these groups or individuals have been identified with the help of stakeholders during a prospective risk identification, and precautions have been taken to prevent differential treatment of the identified groups or individuals—including a requirement to measure whether the groups or individuals are indeed treated differently by the algorithm.

Identifying unfairness risks with stakeholders starts involves looking at how the output of the algorithm is used in decision-making, and how it affects stakeholders that may be unfairly treated. In a simple binary decision, it is usually simply a matter of deciding which of the four possible outcomes—true or false positive and true or false negative—are usually considered good or bad from the perspective of the stakeholder. If the decision is for instance a medical diagnosis the stakeholder wants the outcome to be true, regardless of whether it is positive or negative. If it is an accept-reject decision the stakeholder wants to be accepted, and will often be

happy to be a false positive. In some cases, both ground truths and outcomes are important.

Usually, we are looking at group fairness for specific, identified vulnerable groups. In rare cases, we may be concerned with unfairness towards individuals. If doors for instance don't open for someone whose face cannot be recognized by an algorithm (yes, this happens), this (1) is unfair, and (2) implicitly characterizes a new vulnerable group of people whose face was not learned by the algorithm. Although we are dealing with individuals, we can find those individuals in the data as a group of successive inputs relating to the same individual, and we can apply the same measurement tools to detect this unfairness to individuals. Fairness risks relating to individuals are usually characterized as *social exclusion* risks.

If the algorithm treats a group or groups of people differently, it is apparently capable of picking the members, or successive inputs relating to members, of the unfairly treated group based on the input data of the algorithm. This input data may contain *proxies* that function as static descriptors of group membership.

Assuming the risk identification is adequate, and static descriptors potentially identifying groups have been identified, measurements should be made to quantify the difference in performance or outcome for these groups. These measurements can be made using hypotheses about what the proxies in the data are for group membership, or by using an external data source not used by the algorithm that directly identifies group membership. If the organization has this external data for measurement of unfairness, it is usually personally identifiable data or sensitive data. Permission for its use will be required.

Although a large number of different measures have been proposed in the literature (Verma & Rubin, 2018), the problem in essence boils down to a simple choice between two approaches. We are either comparing the relative *outcomes* for a pair of groups to see whether the difference is within the organization's tolerance margins for outcome inequality, or we are comparing the relative performance of the algorithm for a pair of groups. Regardless of which choice we make, we do often encounter some difference. It is up to the client to decide whether this difference is tolerable, and what it means.

Let's reconsider our running example again. Using the AI application, the public body wants to know whether bias is present in the algorithm against single father household applicants because the benefits will only be awarded if the parent takes care of children the majority of the time.

As pointed out earlier, in the public body example case the two possible outcomes—being manually or automatically processed—are perceived as a punishment vs. reward scenario. Where earlier we addressed making a smart choice in which performance statistic to look at, we now address a similar problem with fairness statistics: which one is meaningful for the problem at hand.

The comparison that matters in this case is mainly the outcome: if it is fairly equal for both groups, there is little risk that fairness issues will be raised. The measure of choice will therefore be *statistical parity* (or *group fairness*; cf. Verma & Rubin, 2018): the probability of being manually processed is equal for both groups:

Table 5 Confusion matrix for the running example

		Predicted outcome of manual processing	
Actual outcome	Total population	**Predicted positive**	**Predicted negative**
	Applicants: 100		
	Single fathers: 10		
	Positive	True positive	Applicants: 80
		Applicants: 10	Single fathers: 6
		Single fathers: 3	
	Negative	False positive	(Distribution between false negatives and true negatives is unknown)
		Applicants: 10	
		Single fathers: 1	

$$\frac{\text{True positive} + \text{False positive}}{\text{Total Population}}$$

This measure is crude, but also one likely to be used by the media to support an accusation of unfairness. The algorithm does not use the gender of the applicant, but the public body does have access to data about the gender of the applicant and household composition from a third party. We can therefore set up *confusion matrixes* for the single father household vs. the rest to gain insight (see Table 5). Ideally, we would like to be able to fill in all four conditions, including the distinction between true negatives and false negatives, but for the negative predictions we don't have information about what the outcome of manual processing would have been.

A quick calculation shows that there is indeed a sizable outcome inequality as expected:

$$\frac{3+1}{10} = 0.4 \text{ vs.} \frac{10+10}{100} = 0.2$$

To justify that difference, it remains relevant to assess the relative accuracies for both groups. Only when the algorithm performs equally well for both groups, the difference can be accepted as a matter of fact. Although it is in principle possible to calculate and compare the weighted *F*-scores, it is more common to compare the precision scores (explained in chapter "Introduction to Advanced Information Technology," Sect. 3.3 of this book). We don't know the distribution between true and false negatives after all. In the context of assessing the problem specification we made the same choice. In the context of fairness, this comparison is labeled

predictive parity (Verma & Rubin, 2018). A quick calculation shows that precision for the group of single father households appears to be even higher than for the total population of applicants, assuring that the root cause of the difference is most likely in the datasets used for training and testing.

$$\frac{3}{3+1} = 0.75 \text{vs.} \frac{10}{10+10} = 0.5$$

Since the number of applicants in the single father household is rather low, we don't have reason to be confident about that conclusion. Ideally one would advise to gather some more data about the group of single father households, but that is obviously going to be difficult: only time will tell. In any case, the audit team neutrally reports differences, possible root causes of those differences it uncovered, and possible ways of removing or reducing those differences, for instance with the help of debiasing algorithms to reduce outcome inequalities (Agrawal et al., 2020). Debiasing should only be used in the understanding that optimizing equality for one type of measure usually worsens the other given the same, unchanged training and test datasets. The bias that caused the unfairness is still embedded in the data in some way. Besides that, if used unwisely, debiasing algorithms may introduce unfairness towards other groups, and may in certain cases be judged unlawful (Xiang & Raji, 2019). The reason for this is simple: giving a specific group a push in the back by definition disadvantages everybody else.

8 Discussion

In this chapter, we have presented a structured approach to define an audit plan for algorithm assurance, based on knowledge from scientific and popular literature and practical experience. Despite our aim to be as comprehensive and detailed as possible, the fact remains that this chapter is fully based on our knowledge and experience as assurance providers in a newly developing field. In this section, we discuss three critical pointers in order for algorithm assurance to mature.

8.1 Transparency and Standardization

Algorithm auditing as a profession is still young. In order for it to become mature profession, it needs, besides more scientific research, shared practical experiences from the field. This calls for a shared learning environment to everyone's benefit. The time of practitioners re-inventing their own wheel is over, especially because the increasing impact of algorithms requires systemic oversight, and governments increasingly realize that it does. Auditors can play a significant role in creating trust, but only if they agree on how algorithm auditing should work.

Standardization would be a logical next move up in the algorithm auditing maturity curve. Firstly, this will help the auditee to understand what is being audited. Even more importantly: one auditor's outcome would be the same as the outcome of another auditor, because the same methodology is followed. Secondly, it also helps to put expectation management in place. What may an auditee, or the receiver of the algorithm assurance report, expect from the auditor and what degree of assurance can the receiver get from the audit report? We truly believe that existing professional associations such as the International Auditing and Assurance Standards Board (IAASB)[7] of auditors have to play a crucial role. But auditors themselves should be open to the approach they follow as well.

The main complication is the diversity of task environment algorithms operate in. One size fits all solutions may impose a cost of control on developers and operators of algorithms that exceeds the business value of many trivial algorithm applications. It is likely that auditor specializations will develop over time for specific high-risk areas governed by different areas of law (medical device safety, consumer rights and legal liability for harm, financial reporting, privacy law, etc.) if standardization is to go deeper than the level of principles.

8.2 Skills and Expertise

In Sect. 7 of this chapter, we have described the specific skills that are required to successfully perform an algorithm audit with the required level of depth. We believe that existing (IT) auditors today do not have this skill set. Yet using the same criteria is just one aspect. Spotting the same risks is an entirely different one. It might be worth a discussion whether specific individual accreditation is required in order to perform algorithm audits.

8.3 Auditing AI with AI

A topic that we didn't discuss in the chapter is how AI technology can also help to perform AI audits. Although this is a fairly new topic, it is worth exploring. The use of AI technology to mitigate risk or exercise control on AI is a lively field. When talking about explainability, or fairness, many in the field of AI immediately think of the research into how to do these things automatically. Obviously. We have looked at a standard audit approach, including all the relevant methodological aspects that are part of it. This approach will not go away: behind any important automated control solution there will be auditor signing off on it. But it is possible to look

[7] https://www.iaasb.org/

beyond control automation and think of AI solutions to general purpose adversarial testing of algorithms in specific domains, for instance vision.

9 Conclusions

Based mainly on the professional experiences of the authors, we introduced the field of Algorithm Assurance in the audit practice. In the context of algorithm assurance, we use a non-standard meaning of the concept of an algorithm: The object of the audit is a computer program, or component of a computer program, containing implementations of a risky AI algorithm or algorithms, to be reviewed in the context of a task in which it is applied *or* a prospective class of tasks in which it may be applied. We distinguished a number of task environment types in which such computer programs may be encountered in an audit context, and the reasons why they may be subject to an audit.

After that we have successively laid the scope of an assurance engagement, the control objectives or principles that guide the assurance engagement, the risk assessment, audit strategy and action plan, and the typical AI-related skills and expertise required of the auditor to do an in-depth investigation of an algorithm.

The main area in which algorithm assurance is still under development is in standardization of what is being tested and how. Standardization is essential for the development of trust in algorithm assurance. The main problem in this area is the diversity of task environments to take into account, which may lead to the development of specializations in the field.

References

Agrawal, A., Pfisterer, F., Bischl, B., Chen, J., Sood, S., Shah, S., Buet-Golfouse, F., Mateen, B. A., & Vollmer, S. J. (2020). *Debiasing classifiers: Is reality at variance with expectation?* Retrieved from https://ssrn.com/abstract=3711681 or https://doi.org/10.2139/ssrn.3711681

Brown, S., Davidovic, J., & Hasan, A. (2021). The algorithm audit: Scoring the algorithms that score us. *Big Data & Society, 8*(1), 205395172098386. https://doi.org/10.1177/2053951720983865

Cheney, J., Chiticariu, L., & Tan, W. C. (2009). *Provenance in databases: Why, how, and where*. Now Publishers.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., ... & Sculley, D. (2020). *Underspecification presents challenges for credibility in modern machine learning*. arXiv preprint arXiv:2011.03395.

English, R. (2021, July 26). Discriminatory basis of child tax credit is justified, rules supreme court. *UK Human Rights Blog*. Retrieved March 23, 2022, from https://ukhumanrightsblog.com/2012/05/17/discriminatory-basis-of-child-tax-credit-is-justified-rules-supreme-court/

European Commission. (2019, December). *The assessment list for trustworthy artificial intelligence*. Retrieved from https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Ewals, R., Francot, J., Frins, C., Houtekamer, D., Van Helden, M., Matto, J., Boon, R., Meulendijks, J., & Bruggeman, A. (2019, December). Handreiking voor SOC 2® en SOC 3® op basis van ISAE3000 / richtlijn 3000A. NOREA. Retrieved from https://www.norea.nl/nieuws/6509/nieuwe-handreiking-voor-soc2-en-soc3-rapporten

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (Vol. 2, no. 12, pp. 1137–1143). Morgan Kaufmann. CiteSeerX 10.1.1.48.529.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A review of machine learning interpretability methods. *Entropy, 23*(1), 18.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue, 16*(3), 31–57. https://doi.org/10.1145/3236386.3241340. ISSN 1542-7730.

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D. & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 33–44).

Shen, H., DeVos, A., Eslami, M., & Holstein, K. (2021). Everyday algorithm auditing: Understanding the power of everyday users in surfacing harmful algorithmic behaviors. *Proceedings of the ACM on Human-Computer Interaction, 5*(CSCW2), 1–29.

Van Smeden, M., Moons, C., Hooft, L., Kant, I., Van Os, H., & Chavannes, N. (2021, December). *Guideline for high-quality diagnostic and prognostic applications of AI in healthcare*. Ministry of Health, Welfare and Sport. Retrieved from https://www.datavoorgezondheid.nl/documenten/publicaties/2021/12/17/guideline-for-high-quality-diagnostic-and-prognostic-applications-of-ai-in-healthcare

Verma, S., & Rubin, J. (2018). Fairness definitions explained. In 2018 *IEEE/ACM International Workshop on Software Fairness (Fairware)* (pp. 1–7). IEEE.

Wirtz, B. W., Weyerer, J. C., & Kehl, I. (2022). Governance of artificial intelligence: A risk and guideline-based integrative framework. *Government Information Quarterly*, 101685.

Xiang, A., & Raji, I. D. (2019). *On the legal compatibility of fairness definitions*. arXiv preprint arXiv:1912.00761.

Demystifying Public Cloud Auditing for IT Auditors

Jacques Putters, Jalal Bani Hashemi, and Ayhan Yavuz

1 Introduction

Over the course of the past decade, cloud computing has become the underpinning infrastructure that supports trends such as the Internet of Things, data analytics, and artificial intelligence. It is giving organisations a competitive advantage in digital transformation in terms of innovation, agility, resilience, and skills. As more organisations become more aware of these prospects, adoption of public cloud is taking place at a fast pace. In addition, 'The economic, organizational and societal impact of the pandemic will continue to serve as a catalyst for digital innovation and adoption of cloud services', said Henrique Cecci, senior research director at Gartner. 'This is especially true for use cases such as collaboration, remote work, and new digital services to support a hybrid workforce'. As a result, global cloud adoption will continue to expand rapidly. Gartner forecasts end-user spending on public cloud services to grow from $396 billion in 2021 to reach $482 billion in 2022 (Gartner, 2021). Additionally, by 2026, Gartner predicts public cloud spending will exceed 45% of all enterprise IT spending, up from less than 17% in 2021.

The financial services industry was initially hesitant to adopt public cloud technology. Primarily security and compliance concerns in addition to an unclear regulatory position prevented them from migrating regulated workloads into the public cloud and made many of them instead choose for private cloud implementations (Association for Financial Markets in Europe, 2019). These concerns usually pertained to data compromise or exploitation by Cloud Service Providers (CSPs), other CSP clients or law enforcements offices, vendor lock-in, inability to perform control and audit activities, and the loss of physical control. In

J. Putters · J. B. Hashemi · A. Yavuz (✉)
Group Audit, ABN AMRO Bank, Amsterdam, The Netherlands
e-mail: jacques.putters@nl.abnamro.com; ayhan.yavuz@nl.abnamro.com

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_8

addition, the financial services industry is heavily regulated, causing financial institutions to be very—sometimes overly—cautious.

As financial institutions have become increasingly aware that—to stay competitive—the adoption of public cloud technology is a bare necessity, they have been trying to address the aforementioned concerns. In parallel, several guidelines have been published by authorities such as the European Banking Authority (EBA) (2019) and the European Securities and Markets Authority (ESMA) (2020) to ensure that the financial services industry and its regulators have a clear set of standards that say how to address these concerns.

The purpose of this article is to provide a conceptual framework that can be used for auditing operational public cloud systems. We will do this by first describing some of the characteristics of cloud computing in Sect. 2. In Sect. 3 we will outline the journey we—as IT auditors—made to address the challenges that the introduction of public cloud technology brought about. This will include a description of the different frameworks and audit programs we used as a basis for our audit activities. Section 4 contains the case description: It outlines the IT/Cloud transformation that our organisation—ABN AMRO Bank—has been going through since 2012. This transformation initially related to the implementation of a private cloud and was later followed by both the rollout of DevOps and the large-scale migration of applications to Microsoft Azure, but we will limit ourselves to the audit on the operational Microsoft Azure environment and exclude the change program and the DevOps transformation from the scope. Section 5 contains a description of the several audits we have done on public cloud since the start of the bank's journey to the cloud. Based on the knowledge and experience that were gained during the execution of the different audits, we designed a conceptual framework that can be used to organise and define IT audits for public cloud systems. This framework is described in Sect. 6. It does not specifically address the audit of Software as a Service, although some elements of the framework apply to SaaS as well. In Sect. 7, we outline a few discussions regarding auditing public cloud systems and the presented framework. This article ends with our conclusions in Sect. 8.

2 Cloud Computing

There are various definitions of cloud computing. Amazon Web Services[1] (Amazon. (n.d.)) defines cloud computing as 'The on-demand delivery of IT resources over the Internet with pay-as-you-go pricing'. A frequently used definition has been published by the National Institute of Standards and Technology (NIST) (Mell & Grance, 2011): 'Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly

[1] https://aws.amazon.com/what-is-cloud-computing/

provisioned and released with minimal management effort or service provider interaction'. While these definitions give a first idea of what cloud computing is, it can be further illustrated by its characteristics as defined by NIST (Mell & Grance, 2011). These have been universally adopted/accepted and are being referred to in many publications.

The first characteristic is the *on-demand delivery of services*. This means that cloud-based services/resources are provisioned without any human interaction with the cloud service provider. They are delivered automatically whenever and wherever they are needed. These services could be virtual machines, databases, storage, etc. The second characteristic is *broad network access*. It means that access is enabled to whatever resource you want, whenever needed, from any location, if you have (Internet) network access. Network bandwidth and latency are key factors to consider because they will determine the quality of service. The third characteristic is *multi-tenancy/resource pooling*. Multi-tenancy means that software and the associated infrastructure serves multiple customers (tenants), at the same time ensuring data privacy and security/isolation on a logical level. In addition, resource pooling means that different physical and virtual resources are dynamically assigned and reassigned according to customer demand across the client base. The fourth characteristic is *scalability and elasticity*: having the ability to quickly provision/scale up or decommission/scale down resources in the cloud whenever required. The fifth characteristic—*measured service*—means that the usage of resources is measured and reported by the cloud service provider and that clients pay in line with resource usage.

There are four main types of cloud deployment models, i.e. public, private, hybrid, and community cloud computing. Cloud deployment models classify cloud environments based on several criteria, such as ownership, purpose, and scale. As every model has its benefits and disadvantages, companies should choose a model—or a combination of models—that best meets their needs.

Public clouds are owned, managed, and controlled by cloud service providers. They are aimed at making cloud services available to multiple customers (tenants) and they offer extremely high scalability, performance, and throughput thanks to the enormous size of public cloud technology. Some examples of large public cloud service providers are Amazon Web Services, Microsoft Azure, and Google.

A *private cloud* is usually owned, controlled, and used by one single company, but it might be managed/operated by a third party. It offers the owner a much higher level of control as compared to a public cloud, but it comes at a price: considerably higher costs are incurred, as they include the costs of traditional data centre ownership as well as the costs of managing the related infrastructure. In addition, although the technical differences between a public and private cloud are small, private clouds will usually be much smaller in size than their public counterparts. Many public cloud service providers also offer solutions that can be used to implement and support a private cloud environment for customers that need to control their whole IT infrastructure.

Community clouds are quite similar to private clouds. The main difference is that—instead of one company—several companies will own, control, and share the

infrastructure and related community cloud resources. Usually, these companies have similar backgrounds and shared interests and—consequently—similar requirements, e.g. in the areas of compliance, privacy, or security.

A *hybrid cloud* consists of a combination of two or more interconnected cloud deployment models (public, private, or community) that allows companies to choose the cloud environment that best meets the needs of the applications and associated data, even on a per case basis. It offers companies a good compromise between costs and control.

In addition to the distinction between the various cloud deployment models, a distinction can also be made between the three distinct types of cloud computing (Jones, E. (2021): (1) Infrastructure as a Service (IaaS), (2) Platform as a Service (PaaS), and (3) Software as a Service (SaaS). Just like with the cloud deployment models, companies should choose the types of cloud computing services that best meet their requirements as to the level of control, management effort, flexibility, and costs.

IaaS gives companies internet access to processing power, storage, and network facilities, which they can use to deploy and run all kinds of software. IaaS offers the highest level of control and flexibility, but it also requires the most management effort. The cloud service provider manages and controls the underlying cloud infrastructure, but the companies using IaaS services have control over the software deployed on top of the IaaS services (e.g. operating systems, applications).

PaaS provides companies with access to all cloud services that are needed to manage the entire lifecycle of applications, without the burden of having to manage the underlying infrastructure. It includes all software needed to design, program, test, deploy, and run applications. The service provider controls and manages the cloud infrastructure, operating systems, middleware, storage, etc., and the company controls and manages the applications.

SaaS provides companies with internet access to applications that are controlled and managed by the service provider (although some application settings might be configurable). SaaS offers the lowest level of control to the companies using these applications, but also the lowest management effort. In many cases, the SaaS provider will use PaaS or IaaS services from other service providers. The following figure (Fig. 1) shows how the three cloud service types compare as to the level of control, flexibility, management effort, and costs/efficiency.

3 Audit Programs for Public Cloud Audits

Although virtualisation has been around since the 1960s and the first cloud (SaaS) applications became available in the late 1990s, public cloud only really took off when Amazon launched Amazon Web Services in 2006. And although public cloud is being adopted at an accelerating pace, most IT auditors are still quite unfamiliar

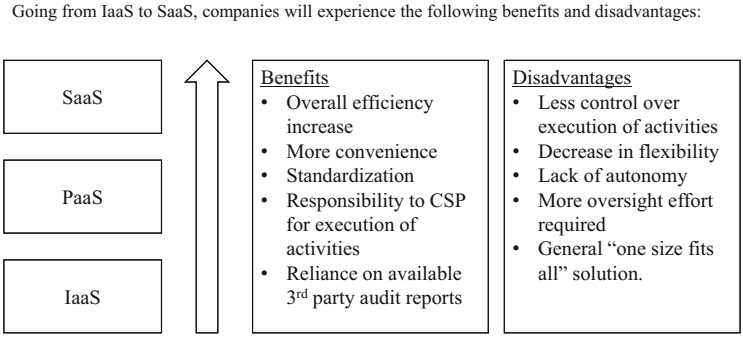


Fig. 1 Comparison of the benefits and disadvantages of the different cloud service types

with the subject matter. Over the course of the past years, some auditors who were being confronted with public cloud on a professional level have resorted to the Certified Cloud Security Professional certification from ISC2 to obtain the required knowledge to audit public cloud developments and systems. Only recently the 'Certificate of Cloud Auditing Knowledge™' (CCAK™) was introduced by the Cloud Security Alliance® (CSA), a global leader in cloud security research, training, and credentialing and ISACA® a global leader in training, education, and certification for IS/IT professionals.

In addition, although there are some audit programs available to help IT auditors figure out how to audit public cloud system(s), processes, and organisation that are included in the scope of their engagement, a holistic view on auditing public cloud subjects is still missing. This is exacerbated by the fact that in a public cloud world there are many variables to consider when defining the scope and objectives of the engagement. What type of cloud computing deployment model is the object of our audit (e.g. public cloud, private cloud, hybrid cloud)? And, which type of service model (e.g. IaaS, PaaS or SaaS or a combination of these)? To what extent are outsourcing controls relevant? Can we rely on available assurance reports? Does the audit relate to a BAU (Business as Usual) system, or do we need to take the migration of a system and associated data to the public cloud into account?

Regardless of the type/subject of audit, it is crucial to plan ahead. Audit programs or plans can be helpful and having them before starting an audit engagement is in most cases mandatory. The audit procedures included in audit programs are to ensure that auditors meet the specific criteria for an audit assignment. Furthermore, the audit program looks to create a framework that can provide auditors with guidelines. The following sections give a brief overview of (1) the shared responsibility model, (2) frameworks/sets of best practices, and (3) work programs that are currently available and that can help auditors design a suitable audit program, including the required audit procedures.

3.1 Shared Responsibility Model

Cloud service providers explicitly communicate the shared responsibility model to their clients. It explains their view on the responsibilities of management and security of their cloud services as managed by the organisation/consumer as it deploys workloads in the cloud, versus those managed by them as the provider of those services. The line between the organisation's responsibilities and those of the provider is also the demarcation between the assets, processes, functions, and associated controls that the provider owns and is responsible for, and the ones of the organisation/consumer. Please note that the views on the shared responsibility model vary between the different CSPs. The following tables show the shared responsibility model as used/published by Microsoft and Amazon Web Services, respectively (Tables 1 and 2):

In a traditional data centre/on-premises model, the organisation/consumer is responsible for management and security across its entire operating environment, including applications, physical servers/hardware, network configuration, user controls, and even physical and environmental security/control. In a cloud environment, the service provider takes on a share of the operational burden. By working together with the CSP and by sharing portions of the security responsibilities, it is possible to maintain a secure environment with less operational costs. When CSPs speak of 'shared responsibility', it is important to understand that the user and CSP never

Table 1 The shared responsibility model according to Microsoft (2022c)

	Responsibility	SaaS	PaaS	IaaS	On-Prem
Responsibility always retained by the customer	Information and data	■	■	■	■
	Devices (Mobile and PCs)	■	■	■	■
	Accounts and identities	■	■	■	■
Responsibility varies by type	Identity and directory infrastructure	■	■	■	■
	Applications		■	■	■
	Network controls		■	■	■
	Operating system			■	■
Responsibility transfers to cloud provider	Physical hosts				■
	Physical network				■
	Physical datacenter				■

■ Microsoft, ■ Customer, ■ Shared

Table 2 The shared responsibility model according to Amazon Web Services (Amazon, 2021)

CUSTOMER responsibility for security 'in' the cloud	Customer Data			
	Platform, applications, Identity & Access Management			
	Operating system, Network & Firewall configuration			
	Client-side data	Server-side encryption (file system and/or data)	Networking traffic protection (encryption, integrity, identity)	
	Encryption & Data Integrity			
	Authentication			
AWS responsibility for security 'of' the cloud	SOFTWARE			
	Compute	Storage	Database	Networking
	HARDWARE/AWS GLOBAL INFRASTRUCTURE			
	Regions	Availability zones	Edge locations	

really share responsibility for a single aspect of operations. The parts of the application and infrastructure stack that a consumer can configure, are solely managed by the consumer of the services, and the CSP does not dictate how the service consumer should secure his parts. Likewise, the user/consumer has no control over how the CSP secures their portions of the application and infrastructure stack. The user/consumer usually has the ability and right to access the CSP's certifications and related reports (e.g. SOCI, SOCII, SOCIII, FedRamp, ISO) to verify that their systems are secure and that they are adhering to the agreed terms and conditions. CSPs publish these reports regularly and freely, and the most current reports are always accessible to their clients. Please note that not all CSPs offer one or more of these reports as it can be costly to produce them/obtain these certifications.

In our cloud audits, we have used the Microsoft Azure Shared Responsibility Model to make clear demarcations of the in-scope and out-of-scope elements in our audit engagements. Moreover, we have also used the model in our audit planning process to find gaps in our audit coverage.

3.2 Frameworks

Some existing frameworks give a solid foundation for the creation of work programs for audits on public cloud systems, these are described below:

- *ISACA:* COBIT (Control Objectives for Information and Related Technologies) (Haes et al. (2015))—framework for enterprise governance of IT. The framework defines a set of generic processes for the management of IT, with each process defined together with process inputs and outputs, key process-activities, process objectives, performance measures, and an elementary maturity model.
- *AXELOS:* ITIL (Information Technology Infrastructure Library) (Axelos, (2020))—a library of best practices for managing IT services and improving IT support and service levels. One of the most essential parts of ITIL is the

configuration management database (CMDB), which provides the central authority for all components—including services, software, IT components, documents, users, and hardware—that must be managed to deliver an IT service.

- *The National Institute of Standards and Technology* (NIST) Information Technology Laboratory regularly publishes research, standards, and guidelines on information systems and security. NIST Special publication SP 800-53 outlines the standards and guidelines for Security and Privacy Controls for Information Systems and Organizations. This publication lists the controls that will enable companies to protect themselves against a diverse set of threats and risks. The controls cover 20 areas, including access control, awareness and training, audit and accountability, contingency planning, and incident response. The classification of the information system (low, medium, or high) will determine the controls that must be implemented and monitored. SP 800-53 is widely used by cloud service providers as the set of reference controls that they have their audit or compliance teams audit them against.
- *ABN AMRO IT Organisation: Standards for Cloud Risk Control*. In the proof-of-concept phase of the Microsoft Azure and the Amazon Web Services platforms, the ABN AMRO IT organisation defined the Standards for Cloud Risk Control to guide the implementation of workloads on these platforms. The standards were created in close collaboration with cloud specialists from Azure and AWS and representatives from both the Corporate Information Security Office and the Corporate Technology Office. These standards define which controls need to be implemented to adopt and use the platform services securely. The focus of these requirements is on which controls teams must implement. The standards apply to IaaS, PaaS and SaaS services.

3.3 Audit Programs

There are several audit programs that can be used for auditing cloud service providers or implementations of public cloud in organisations.

First, there is the Cloud Controls Matrix (CCM) of the Cloud Security Alliance. CCM is composed of 197 control objectives that are structured in 17 domains (shown below in REF), covering key aspects of cloud technology. The controls in the Cloud Controls Matrix (CCM) are mapped against industry-accepted security standards, regulations, and control frameworks including but not limited to ISO 27001/27002/27017/27018, NIST SP 800-53, AICPA TSC, German BSI C5, PCI DSS, ISACA COBIT, NERC CIP, FedRamp, CIS, and many others (Table 3).

It can be used as a tool for the systematic assessment of a cloud implementation and provides guidance on which security controls should be implemented by which actor within the cloud supply chain and is considered a de-facto standard for cloud security assurance and compliance (CSA, 2021).

Second, there is ISACA's Cloud Computing Management Audit Program (ISACA, 2020–2021), which focuses on the governance affecting cloud computing,

Table 3 Overview of 17 domains of the Cloud Controls Matrix (CCM)

Audit and Assurance	Identity and Access Management
Application and Interface Security	Interoperability and Portability
Business Continuity Management & Operational Resilience	Infrastructure & Virtualization Security
Change Control and Configuration Management	Logging and Monitoring
Cryptography, Encryption, and Key Management	Sec. Incident Management, E-discovery & Cloud Forensics
Datacentre Security	Supply Chain Management, Transparency, and Accountability
Data Security and Privacy	Threat and Vulnerability Management
Governance, Risk Management, and Compliance	Universal Endpoint Management
Human Resources Security	

Table 4 Processes of ISACA's Cloud Computing Management Audit Program

Governance of Cloud Computing Services	Incident response, notification, and remediation
Enterprise Risk Management	Application Security
Information Risk Management	Compliance
Third-party Management	Tools and Services
Legal and Electronic Discovery	Application Functionality
Legal Compliance	Data Security and Integrity
Right to Audit	Key Management
Auditability	Identity and Access Management
Compliance Scope	Virtualization
Certifications	Standards and Best Practices
Service Transition Planning	

Table 5 Areas of ISACA's Azure Audit Program

Governance	Logging and monitoring
Network configuration and management	Security incident response
Identity and access management	Data encryption controls
Resource security	

contractual compliance between the service provider and customer, and control issues specific to cloud computing. Controls and test-steps are included (with mapping to COBIT5) and cover the following processes (Table 4):

Third, there is ISACA's Azure Audit Program (ISACA, 2020–2022), which helps auditors in their assessments of whether the enterprise's use of Azure services supports achievement of strategic goals through covering the following areas (Table 5):

The Cloud Computing Management Audit Program is agnostic to the cloud platform being used, while the Azure Audit Program holds specific details and is

tailored towards the Azure environment. For our specific use case, these two programs were complementary to each other.

3.4 Suitability of the Available Frameworks and Work Programs

The shared responsibility model and the available frameworks and work programs will be of added value for IT auditors when deciding how to audit public cloud implementations, system(s), processes, and organisation(s). The shared responsibility model provides IT auditors with a reference for deciding what to expect from the user organisation versus what to expect from the service provider per type of cloud computing. This is especially important for scoping purposes. In addition, the frameworks and work programs mentioned in the previous paragraphs give a basis for auditing specific aspects of public cloud implementations such as encryption & key management, governance & risk management, infrastructure & virtualisation, third party (risk) management, etc. However, there are three disadvantages to the use of these work programs. First, they lack the holistic perspective, as they do not show or explain the relative importance and interrelationships between the individual components of the work programs. Second, although the level of detail differs between these work programs, none of them are sufficiently specific and give the required guidance for more experienced IT auditors if they aim to do a more in-depth audit of public cloud implementations. And third, these frameworks and work programs do not distinguish between the platform and the workloads running on the platform, although this is a relevant distinction when auditing public cloud implementations.

4 Case Description: The ABN AMRO IT/Cloud Transformation

4.1 ABN AMRO Bank

Headquartered in Amsterdam and employing some 18,000 people, ABN AMRO is the third largest bank in the Netherlands. The foundation of the current bank was laid when the ‘Algemene Bank Nederland’ (ABN) merged with the ‘AMRO Bank’ in 1991, thereby creating the largest bank in the Netherlands, the 6th bank in Europe and the 16th bank worldwide. A period of domestic and international mergers and acquisitions followed. By 2007, ABN AMRO was the second-largest bank in the Netherlands and the eighth largest in Europe by assets. It had operations in 63 countries, with over 110,000 employees.

In 2007, a consortium that consisted of the Royal Bank of Scotland (RBS), Fortis, and Banco Santander under the name RFS Holdings, made an offer on the shares of ABN AMRO. This offer was accepted by the shareholders in September 2007 and ABN AMRO was split up by its new owners.

When in 2008 the global financial crisis hit the financial service industry, the Belgian-Dutch Fortis Group that had taken over the ABN AMRO Business Units Netherlands, Asset Management and Private Banking had to be bailed out by the Dutch and Belgian governments. The Dutch government bought the Dutch activities of Fortis Bank, Fortis' insurance activities, and Fortis' share in the ABN AMRO Bank. The Dutch government later decided that these parts would be integrated in the new ABN AMRO Bank which eventually took place on 1 July 2010.

The current ABN AMRO Bank is aiming to become a personal bank in the digital age. This strategy rests on three pillars:

1. Reinvent the customer experience: Getting closer to clients and offering them a fully digital experience with best-in-class services and products.
2. Support our clients' transition to sustainability.
3. Building a future-proof bank.

Information Technology is at the heart of the bank's strategic goals. To improve the productivity and lower the costs of IT, senior IT management decided to transform from an agile into a DevOps organisation. This transformation would be strengthened by also moving to the public cloud and away from a managed IT service provider. The bank applies a cloud-first strategy and has chosen Azure as its strategic cloud platform and AWS as its challenger cloud platform (Monterie, 2020).

4.2 IT Within ABN AMRO Bank

IT within ABN AMRO has its foundation in the IBM Mainframe systems that have been used since the 1960s. But over the course of the last 30 years, a wide variety of platforms had been added to the environment, especially at the end of the 1990s when large-scale client-server system implementations took place. This resulted in an overly complex, expensive, and difficult-to-control situation. A large variety of platforms was used: Open VMS, HP Unix, AIX, Linux, Solaris, AS400, Windows server, Tandem, etc. As the need for a reduction of complexity and cost grew, a virtualised platform was identified as a means to accomplish this. ABN AMRO decided to select one of the private cloud offerings from IBM to become the platform of choice for the years to come. It also enabled the organisation to explore cloud technology and to experience how to make applications cloud-ready. In 2016, this on-premises dedicated cloud platform went live, and a program was started to migrate hundreds of applications from the legacy/midrange platforms to this private cloud environment.

It was already clear at the time of implementation that the functionality offered by the private cloud would not be able to compete with the ones from large cloud

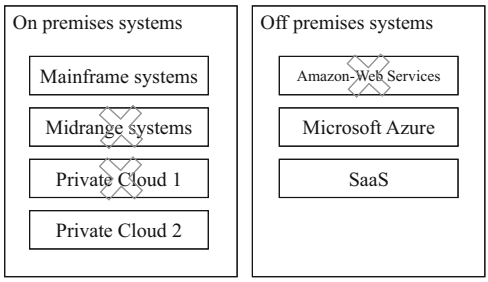


Fig. 2 ABN AMRO platform landscape after the cloud transformation

service providers such as Amazon Web Services, Microsoft Azure, and Google. Consequently, in 2017 two proofs of concept were started to experiment with Microsoft Azure as well as with Amazon Web Services. Secondly, an alternative private cloud solution was explored and implemented. The proof of concept of the two public cloud platforms was so successful that a multi-platform strategy was finally adopted where both Microsoft Azure and Amazon Web Services had their place. The two private cloud solutions were maintained, next to the traditional Mainframe environment.

In 2019 IT began to realise that—even although steps forward were being made—a drastic strategic shift was needed for the bank to become more efficient. Although the many midrange systems had now to a substantial extent been migrated to the private cloud environment, a further reduction of complexity had to take place. Based on the experiences with the two public cloud platforms (AWS and Microsoft Azure), it was decided to use public cloud as a strategic platform next to the Mainframe that would remain to run many core systems. The choice was made to select one public cloud platform—Microsoft Azure—instead of using the two platforms available and to migrate all private cloud and AWS hosted workloads to Microsoft Azure (Rosa, J., & Dee, M. (2020)). This will result in the platform landscape as depicted in Fig. 2.

To achieve the IT transformation, a program organisation was put in place that had three main aims. Migrating all private cloud systems and AWS workloads to Microsoft was one of them. The other two pertained to the introduction and rollout of DevOps and the optimisation and consolidation of vendor relations.

5 Internal Audit Activities on Public Cloud

In this section, a description is given of the activities that enabled the internal audit department of ABN AMRO to opine on the public cloud environments. These activities were primarily focused on educating the auditors, creating the audit universe for cloud, and keeping a close eye on the transformation process.

5.1 Bringing the IT Auditors Up to Speed

One of the best practices of the IT Audit section of Group Audit ABN AMRO has always been the pro-active involvement in projects and programs. As Benjamin Franklin said: 'An ounce of prevention is worth a pound of cure'. This saying has driven IT auditors to get involved in programs/projects as early as possible to provide the program/project with audit feedback at a moment that fixing shortcomings is possible without affecting timelines and budget too much.

So, as soon as it became clear that IT was going to run a program aimed at exploring public cloud as a potential replacement or alternative for on-premise systems, the decision was made to dedicate one full time IT Auditor to the program. He had to get acquainted with the program, but also with the subject matter. A basic understanding of cloud computing was needed, so the Certified Cloud Security Professional (CCSP) course was done. Initially, the IT auditor closely monitored the program while gaining knowledge on cloud computing. One of the first things that needed his attention were the new cloud policies and standards that had to be put in place. In addition, the program was audited, covering both program governance and its deliverables. After that first period, the IT auditor made sure that public cloud was included in the multi-year audit plan and that specific audit activities were planned for the next year.

Gradually, the IT Audit department started to realise that public cloud was here to stay and would gain relevance in the years to come. With increasing cloud adoption, the audit workload would also increase. Sharing the acquired cloud knowledge and experience was needed for the other IT auditors to become proficient and remain relevant in an organisation with a substantial number of workloads running in the public cloud. A start was made by organising and providing internal training to the rest of the IT auditing community. This proved to be a very cost-effective method.

Given that CCSP certification is cloud-agnostic, the training material does not include the details that are specific for a cloud service provider. Once it became clear that the bank would be focused on Azure, the choice was made for the Azure Fundamentals course. As there was sufficient online training material on Azure, in-house training was needed, and staff was able to follow this course online at their own pace.

5.2 Audits Performed

The audits that were done on the implementation of public cloud, took place during three distinctive phases. The first phase was characterised by the ad hoc nature of the audits. Through continuous business monitoring, it became clear that the IT organisation intended to put in place two public cloud platforms. From the moment the two proofs of concept were started to experiment with Microsoft Azure as well as with

Amazon Web Services, until the time that the IT organisation decided that both Azure and AWS would be strategic platforms, the following audits were done:

- *Initial program and cloud platform set-up:* An audit was done from the very start of the two proofs of concept to ensure that no critical mistakes were made, and that cloud computing was being used in a controlled fashion. This included auditing the program organisation, but also whether the products being used were secure enough and compliant with the bank's policies and standards. One of the focus areas of the audit was the set of standards that was drawn up by the IT organisation to act as a basis for the configuration of cloud services and associated workloads. This all resulted in two audit reports: One for each platform.
- *Cloud Service Provider audit:* At the end of 2017, ABN AMRO Group Audit started taking part in the Collaborative Cloud Audit Group (CCAG) and carried out several pooled audits on cloud service providers as part of this group. As one of the early members of the CCAG, we have been actively involved in setting up, organising, and executing these audits. We have shared our knowledge, journey, and experience regarding CCAG in two articles mentioned under References (Pooled audits on cloud service providers—Parts 1 and 2).
- *Cloud Maturity Assessment:* The IT organisation found shortcomings in several areas that impeded an accelerated adoption of public cloud. Consequently, a high priority initiative was launched across the organisation to improve public cloud maturity. The audit aimed at assessing to what extent the initiative resulted in the required improvements to support a controlled acceleration in public cloud adoption. The areas covered included governance, security, architecture, operations, financials, cloud native development, and technical skills.

With the decision to use Microsoft Azure as a strategic platform next to the Mainframe, and to migrate all private cloud and AWS hosted workloads to Microsoft Azure, also the next audit phase started. This audit phase was characterised by the efforts to audit the IT transformation program organisation, the Azure migration organisation, and the Azure platform in a more detailed manner. This resulted in the following engagements:

- *Audit on the IT Transformation program:* This audit was aimed at assessing the readiness of the organisation for a large-scale DevOps implementation and migration to Azure. This audit covered the five principal areas of the program: (1) governance, (2) organisational design, (3) execution and migration organisation, (4) strategic sourcing, and (5) the Azure foundation design and delivery.
- *Azure migration organisation and deliverables:* The program responsible for the migrations of workloads to Microsoft Azure was audited to assess whether it could migrate existing workloads to the Azure cloud environment in a safe and prompt fashion.
- *Migration factory and tooling:* This audit covered the migration workflow 'factory' and associated tooling being used for the migration of workloads to ensure a standardised, controlled approach when migrations take place. Tooling was of special interest because of the prominent level of automation involved.

– *Cloud landing zone:* The cloud shell chosen as the private space of the bank was audited to ensure isolation from the public space, isolation between different workloads, and for the separation of development and production environments.

As more workloads had been migrated to Azure, the emphasis of the audit activities shifted to auditing applications running on Azure in addition to the ongoing audits on the Azure platform, now being a more balanced mix in the audit plan. This characterised the third and final audit phase. During this phase, the following audits were done:

– *Cloud platform products:* Platform products/services can be used as building blocks to set up the infrastructure for applications: e.g. Windows/Linux Virtual Machines, Storage Accounts, SQL databases. Using a risk-based approach we selected and audited the most critical components to ensure that these building blocks are designed and implemented correctly/securely.
– *Cloud security/directory services:* Our cloud platform relies on Azure AD for Identity and Access Management, and Azure Sentinel for security analytics and threat intelligence. We have performed audits on these crucial components as many products and services are depending on them.
– *Cloud applications:* Using a risk-based approach, we selected applications running on the Azure platform and performed an examination of all underlying cloud products and services to assess whether control processes were suitably designed and operating effectively.
– *Deployment pipelines:* By auditing pipelines we wanted to verify whether these essential components were (technically) sufficiently secured to ensure separation of environments and segregation of duties.
– *Software-As-A-Service (SAAS):* Next to the platform and application audits, we also performed audits on the usage of the riskiest SaaS applications. The focus here was primarily on the user-organisation controls and the integration with the banks internal processes (e.g. incident/problem/change management) and shared services (e.g. IAM, SIEM, CMDB).

A key element in all our internal engagements was the arrangement of read-only access to resources (e.g. products, groups, subscriptions) in the cloud environments. Our Infrastructure Managed Services department (refer to Sect. 6.2) created a new role for us with almost tenant-wide access (excluding the LogAnalytics workspaces) in Azure and as eligible role ‘security reader’ for AAD analysis. This provided us with uninterrupted access to (technical) audit evidence and streamlined our audit work in terms of efficiency.

6 Conceptual Framework

The audits discussed in Sect. 5.2 served as the inspiration for the design of the framework that we will present in this section. This conceptual framework can be used for planning and executing audits on public cloud implementations. While elaborating on the various components of this framework, we will refer to several products and services that are used by ABN AMRO and/or that are offered by Microsoft Azure. Figure 3 depicts the complete conceptual framework to audit public cloud implementations. However, most of the concepts that will be elaborated on will apply to other public cloud implementations as well. We will start our outline with the section on Cloud Service Providers (1. Cloud Service Provider) and work upwards to the Governance section (6. Governance).

The focus of our explanation will be on Infrastructure Managed Services (component 2 of the framework) and the Services and Workloads (component 3 of the framework). In our opinion, this has the most added value, given the fact that there currently is hardly any concrete guidance for IT auditors available on these two topics. We will supply both the contextual information and risk-control descriptions that will help IT auditors gain a better understanding of the subject matter and that will aid them in designing the audit programs they can use to audit public cloud implementations. As there are several publications and work programs that adequately cover the other components (Cloud Service Provider, Processes, Policies & Standards, and Governance), we will only explain what the specific attention points for these topics are in the context of public cloud. We will refer to relevant articles and audit programs when covering these topics.

6.1 Cloud Service Provider

Cloud service providers supply the basic services that their customers can use to build, run, and support their applications. These services pertain at the very least to

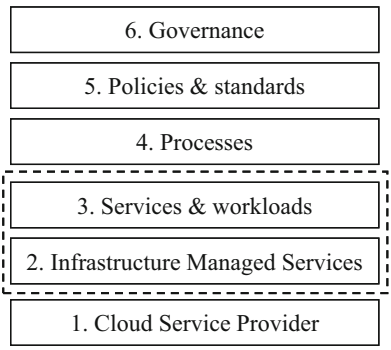


Fig. 3 Conceptual framework for auditing public cloud implementations

the physical data centres, physical networks, and physical hosts, on top of which virtualisation software runs. Customers could use a wide range of added services, depending on their specific needs.

To get assurance on the services that are outsourced to the CSP, there are three complementary approaches: The first approach is aimed at assessing the way the retained organisation manages the outsourcing arrangement. This is usually done by collecting information as to the outsourced services, inspecting this information to verify that performance is in line with expectations and contractual obligations, and finally to—whenever applicable—contact the CSP to request for corrections. The auditor will need to assess these processes to come to an overall conclusion on the level of control over the outsourced services. With the second approach, the retained organisation uses the available assurance reports and certifications made available by the CSP to get the assurance that is needed. The scope and applicability of the assurance reports will need to be assessed in addition to the proficiency of the external auditor, the quality of the report, etc. The third approach is aimed at carrying out audits at the CSP, possibly in collaboration with other clients. By carrying out these audits, auditors will be able to provide assurance on the scope of the audit. A combination of these three approaches is highly encouraged as they are complementary.

As regulatory guidelines (European Banking Authority, 2019; European Securities and Markets Authority, 2020) and other publications (e.g. Institute of Internal Auditors, 2018) are already available on how to audit outsourced activities or how to use a pooled audit approach to audit cloud service providers (Akdeniz et al., 2020; Bani Hashemi et al., 2020), we will refrain from elaborating on these topics in this section.

6.2 Infrastructure Managed Services

Although cloud service providers such as Microsoft Azure and Amazon Web Services offer many possibilities for DevOps teams to utilise ready to use (or customise) services, several aspects will be mostly the same for all DevOps teams within a company. For a start, there is one Active Directory for Identity Management for on-premise usage with centralised management and there will be only one enterprise Azure Active Directory (AAD) for usage in Azure. The same holds for Azure policy management: At the highest level (tenant management group), the Azure policies will be managed by a central department. At lower levels, DevOps teams may specify their own specific policies, if they do not contradict the central policies. At ABN AMRO, we decided to centralise functions that should be the same for all DevOps teams into Infrastructure Managed Services. These are:

1. Identity management
2. Policy management
3. Product development

4. Subscription management/Secure landing zones
5. Network management
6. Support for implementing security event monitoring

These functions will be discussed into more detail below.

6.2.1 Identity Management

Identity management (IdM), also known as identity and access management (IAM) ensures that only authorised people have access to the technology resources they need to perform their job functions. It includes policies and technologies that encompass an organisation-wide process to properly identify, authenticate, and authorise people, groups of people, or software applications through attributes including user access rights and restrictions based on their identities.[2] Most companies with a large IT landscape will recognise the need for central administration of identities and access rights: it makes it easier to block all access to systems at once of staff leaving the company and role-based access can be implemented across different IT systems. The ideal situation would be that all local user administrations of IT systems (such as e.g. Linux, Oracle and Windows Active Directory) and applications are/can be onboarded to central solutions like Ping Identity or SailPoint.

When an organisation starts using Azure services, it can only manage identities and access rights by using the Azure Active Directory (AAD) SaaS service. There are four offerings: Free, Office 365 apps, Premium P1, and Premium P2. The free version has an object limitation, and the Office version comes with added features to work with the functionality on the Microsoft collaboration platform. The premium editions offer more advanced access control capabilities and for heavily regulated industries like government and finance P2 is recommended. Obviously, making changes to the Azure AD directly/interactively must be restricted to a limited set of (tier 0) administrators. Analogous to Microsoft administrators who under specific conditions can make changes to production systems, they must use dedicated hardware to maintain the AAD. Complementary to their laptop for day-to-day tasks they use a separate specifically hardened device that can be used for AAD maintenance only.

As a detective measure, all actions on the AAD must be logged and monitored. We refer to the Monitoring section for further detail.

By using Azure in addition to your traditional IT landscape (hybrid situation), a new identity and access management system is introduced. Luckily, using the AAD Connect sync service, information held in the on-premises Active Directory (e.g. users/identities and user groups) can be synchronised towards Azure AD (one way), the central IAM system is so to say, ‘in the lead’.

[2] https://www.vmware.com/topics/glossary/content/identity-management.html

For a new application or supporting DevOps team in Azure, the following steps for onboarding are typical: A few generic (company/organisation) roles typically apply to almost every application and Azure service, such as administrator, developer, and operator. For each (new) team these roles can be defined centrally and the DevOps team members are added/assigned to the appropriate role(s) centrally as well. The identities of the team members can be added to the on-premises AD the moment they join the company. So only their group membership needs to be synchronised from the central IAM solution to the on-prem AD. Next, via AAD Connect, group memberships are synchronised to the Azure AD. The synchronisations are of course automated, i.e. they do not require administrator intervention.

When performing an audit, it is important to understand the structure of access rights assignments in Azure to identify anomalies or undesired implementations.

After the relevant AD groups have been defined, within Azure the access rights per (organisation) role need to be determined. An important feature of Azure is inheritance. Access rights in Azure can be granted at the following levels: management group, subscription, and resource group. They are in hierarchical order, which means that rights granted at management group level are inherited to all lower-level subscriptions and resource groups. Access rights granted at subscription level are inherited to all lower-level resource groups. When a subscription is shared between several teams, one would expect limitation of access to their specific resources/resource groups. When the subscription is for one team, then assignment of access rights at subscription level can be expected.

For a central team that manages Azure at enterprise/company level, access rights are expected to be granted/assigned at tenant/management group level. In practice, it is possible to have different management groups in a hierarchical relation and access right inheritance will follow that order.

At practically all levels combinations of built-in and custom Azure roles are to be expected. These Azure roles contain the permissions. Built-in roles like owner, contributor, reader, and user access administrator are widely used and set at management group level. On subscription level, the built-in role Support Request Contributor is assigned additionally.

In addition, custom roles can be built regarding e.g. role management, cost management, and policy management. These Azure roles are assigned at management group level to members of central teams managing Azure. Roles can be assigned to groups which in our case correspond to groups in the AAD and ultimately to the groups in the central identity and access management system. But Azure roles can also be assigned to applications and users which only exist in Azure and AAD. Now is a good moment to pause at the question which Azure roles should be assigned to which generic organisation roles. The actual assignments depend on the Azure service, but as a rule of thumb regarding the built-in roles the following can be configured (Table 6):

The owner role is very powerful and should therefore not be assigned permanently to members of the DevOps team. It is good practice to assign the owner role to a non-personal account like the service principal which is created (in our case) when the subscription or resource group is created. The service principal account is used

Table 6 Assignment of Azure built-in roles. Adopted from Microsoft (2022d)[a]

Azure built-in role	Permissions	Assign to
Owner	Full access to all resources	Service Principal
	Delegate access to others	
Contributor	Create and manage all types of Azure resources	Developer
	Cannot grant access to others	
Reader	View Azure resources	Central team and Auditor
User access administrator	Manage user access to Azure resources	Central team and Privileged Identity Manager (PIM)

[a] The table was extended with column 'Assign to'

by the CI/CD pipeline and used to deploy services and changes to the appropriate environments via a service connection. The contributor role is better suited to grant to developers on a permanent basis. The other roles are assigned outside the DevOps team. In roles mentioned above, no built-in roles are assigned to administrator and operator. In certain environments, administrators typically have the highest access rights, which in this case would be the built-in owner role. That is however undesired from a control perspective and in practice we do not expect to see many assignments of built-in roles to the generic administrator role. Only for a part of the Azure services does a built-in operator role apply (e.g. backup operator, Cosmos database operator, and site recovery operator). Obviously, when these services are not used, no assignment to the generic operator role is required.

The Azure AD P2 edition contains the Privileged Identity Manager (PIM) that can be used to assign/elevate privileges of Azure identities temporarily. For example, when the contributor role of a specific service is assigned to be eligible for a developer in the production environment, then he can only obtain that role after approval of a peer (i.e. someone in the group of identities that are also eligible for that role). PIM makes sure that the access rights are withdrawn after the pre-defined time window for usage has elapsed. The performed actions are logged in an activity log file.

There is more to say about non-personal accounts like the service principal account and managed identities, and their relation to Azure KeyVault and the fact that not all Azure services support Azure AD authentication. The auditor is recommended to be aware of the additional details while performing an audit. Microsoft does have online documentation that can be consulted.

6.2.2 Policy Management

Policy management is the process of creating, implementing/enforcing and maintaining policies within an organisation. Enterprise-wide policies could apply to all business processes, and some could be IT related. The IT-related policies can be generic in order to apply to different environments. However, regarding public

cloud environments a dedicated framework can be useful where the company policies are translated into cloud security controls that are in principle cloud-agnostic. The challenge is how to enforce these controls?

Azure Policy is a service in Azure which allows organisations to create policy definitions which enforce and control the properties of a resource. Requests to create or update a resource are evaluated by Azure Policy (it is a little bit more complex). Each policy definition has a single effect (e.g. audit, deny, disabled). That effect determines what happens when the policy rule is evaluated to match.

Azure has many built-in policies that can be viewed via the Azure portal. Most are disabled or in audit mode by default. By putting them in deny mode, they are enforced. Additional considerations regarding migration to enforced policies will follow below.

Industry experts are divided on the topic of using built-in policies: the majority is not in favour of using built-in policies because Microsoft can change these definitions at any time, which can lead to operational problems. For example, when you enforce geo-redundant backups to be enabled and Microsoft changes the default to disabled, then from the time the change is active there will be no more backups. Therefore, companies that use built-in policy definitions, need to closely monitor policy definition changes made by the cloud service provider, and apply timely life cycle management and testing when changes occur to guarantee continuity of service delivery (or in this case to guarantee the ability to restore data when required). Given data privacy restrictions, an obvious built-in policy to use for EU based companies is Allowed Locations. You can restrict the locations to which resources are deployed to e.g. North Europe (Dublin) or West Europe (Amsterdam) by adding a custom policy. Policies can be assigned at distinct levels: management group, subscription, resource group, and individual resource.

Another attention point is the exceptional case that a specific resource is not available in the desired locations but still necessary for (specific) application teams. To enable this, the policy will typically not be enforced, giving all the other application teams within that subscription too many choices. As a compensating control an alert can be triggered when undesired locations are configured. The risk can be remediated by migrating the application requiring the geographic location to a separate subscription and enforcing the policy on management group level with an exemption for the separate subscription.

In the first stages of cloud adoption, it can be expected that by default practically all policies are in ‘audit’ mode, which means that they are evaluated, but not enforced. For the DevOps teams this may look convenient, but from control perspective it is far from ideal. Putting the policies in ‘deny’ mode would enforce them, but the policies out-of-the-box are not customised to the organisation’ needs. There are over 600 built-in policies in Azure. These are grouped into categories that are partly Azure service specific (e.g. compute, Cosmos DB, and Data Factory) and more generic (like general policies, tags policy, backup, and monitoring). The built-in policies are developed by Microsoft, based on their worldwide experience. But that does not mean that they are good enough or directly applicable for organisations.

Of course, Microsoft realised that and offers the possibility of defining custom policies.

In practice it is not unusual that Azure policies are maintained by different parties within the same organisation. For example, product teams (refer to next section) that are responsible for providing customised versions of the Azure services by using custom policies at resource level. And a central department that maintains the not-service specific policies like the General policies (including Allowed Locations) and the Tags policy. To distinguish the custom policies from the built-in policies a naming convention can help.

Most organisation will start in a situation where only a few policies are enforced. For reasons described in the Subscriptions/Secure landing zones paragraph below, gradually more policies can be enforced as the environment matures. It is hard to over-estimate the effort that is required to determine which Azure built-in policies are wanted/needed to be enforced. The cloud controls of our framework range from generic requirements regarding data leakage prevention to specific product/service settings regarding TLS. The built-in policies can be used to enforce part of the controls, but more than likely additional custom policies need to be designed.

Enforcing the policies is another step that should not be taken lightly: the impact will depend on the number of applications/subscriptions, and the maturity of the DevOps teams. When the applications remain in their same subscriptions/resource groups, then obviously a phased approach, starting in audit mode and resolving all non-compliance before turning to deny mode is the best practice. An alternative approach would be to migrate the applications to other/separate subscriptions instead. Organisations needs to consider whether the same policies should apply to the development and test environment as to the acceptance and production environments. And tempting as it may seem to apply a different set of policies to development and test, one should keep in mind that as of consequence the changes required before going to acceptance would be bigger. In our opinion it is better to apply the same policies, albeit that in development and test most policies remain in audit mode. Policy maintenance will be an ongoing effort since it is expected that new Azure services will become available in the future. When DevOps teams require new services, it needs to be determined whether policies need to be changed and/or new policies to be added.

An auditor would expect that all policies are enforced; however, this might not always prove to be practical. First probably not all built-in policies are necessarily relevant (and would hamper application execution when enforced) and second a balance should be made between security and risk appetite. Secure landing zones where more policies are enforced than in the shared subscription and the individual subscription model can help in striking the balance. By no means is reviewing the policies (e.g. what is (not) enforced) going to be an easy task for the auditor. However, he can benefit from using automation in this area, e.g. by using Azure Governance Visualizer. This is a is a PowerShell based script that iterates your Azure Tenant's Management Group hierarchy down to Subscription level. It captures most

relevant Azure governance capabilities such as Azure Policy, RBAC and Blueprints and a lot more.[3]

Also important to consider is life cycle management per policy (incl. implementation and compliance) and a rationale for either enforcing the policy, or not. In the case of non-compliance, follow-up depends on the type of non-compliance. When policies that should be in deny mode do not apply or can be circumvented, this should be known to the DevOps team and preferably also with the central oversight department. In addition, there should be an approved policy deviation and a planning/path to compliance. Policy rationales can be reviewed by the auditor for plausibility with support from the DevOps teams and Azure experts. The roles and responsibilities of a likely to be implemented Azure Policy Board can be assessed as well taking into account its composition. Furthermore, the auditor can consider verifying whether the applicable company policies and derived cloud security controls have all been covered effectively by the enforced policies. When company policies and cloud security controls have not been (completely) covered by the policies enforced in Azure, then the gap needs to be determined and compensating controls need to be assessed.

6.2.3 Product Development

Cloud service providers manage a large set of services that cloud customers can deploy in their subscriptions. These services are cloud-based products that include compute, storage, networking, databases, development tools, and management tools. Product development is the process to customise native cloud services (by the cloud customer) to ensure that they meet the organisations security standards.

When an organisation starts its cloud journey with inexperienced DevOps teams, it can be considered necessary to protect the teams against themselves and let them use only customised/approved Azure services that could be deployed from a separate repository (the product catalogue), so not directly from Microsoft. Complaints from the DevOps teams are to be expected from this restrictive approach, as teams somehow always need ‘more exotic’ services. The customisation depends on the service. An easy-to-understand example is the requirement for TLS 1.2 for secure communication to services like Azure SQL server and Azure Data Factory. Another example is selection of encryption at rest for storage and databases like SQL and Cosmos DB.

The preference to protect the teams comes with a price. The Azure services of Microsoft are updated and patched regularly. Using a customised version means that the organisation will have to perform life cycle management and maybe patch management on these services itself. It is not unimaginable to have 3–4 versions per service in the product catalogue which all need to be maintained. So besides customising the services, it is important to manage timely upgrades.

[3] https://github.com/JulianHayward/Azure-MG-Sub-Governance-Reporting

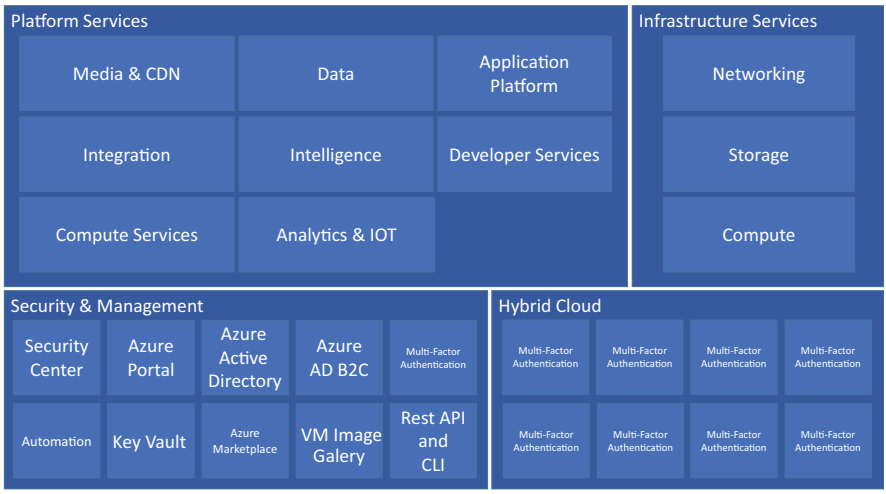


Fig. 4 Overview of Azure services according to Microsoft (n.d.)

Regarding the virtual machines (VMs) of Linux and Windows, an organisation can choose to implement/use CIS (globally recognised standard for secure baselines) hardened images. Attention point is the fact that Microsoft changes the VMs more frequent than CIS changes the hardened image. In other words, there is a delay in CIS hardened images becoming available. In order to keep up the pace with Microsoft, product groups can better build the images themselves, making sure they have the latest copies and all required patches. The images can then be replaced at a higher rate in the product catalogue. Mandatory automated update of deployed VMs should also be considered.

As soon as a DevOps team deploys a service (e.g. from the product catalogue shown below, Fig. 4) into their resource groups, it becomes their responsibility to maintain the lifecycle and to perform vulnerability and patch management, at least for the IaaS services. Of course, tooling (e.g. Microsoft Defender for cloud) is available in Azure to monitor resources, but teams need to be aware of their responsibilities first. When teams are accustomed to support traditional applications where development and operations responsibilities are split, teams do not automatically get the mindset required to maintain public cloud applications.

6.2.4 Subscription Management/Secure Landing Zones

With the on-premises data centres, developers had to request a server to deploy their applications to. Not long ago, these were physical machines, and the ordering process could take months. By keeping servers in stock, the process could be accelerated and by using virtualisation the process could be accelerated even further.

But still it would take several days to configure the (virtual) server before it was ready to use by the developers.

In cloud environments, subscriptions and the associated resource groups can be considered the equivalent of the physical environment. DevOps teams can deploy Azure services into resource groups which are logical containers. Resource groups are part of subscriptions which have limits or quotas on the number of resources you can create and use. Organisations can use subscriptions to manage costs. As part of subscription management, we consider the design and implementation of Azure management groups, subscriptions and resource groups and their (hierarchical) relations. The design is important because the hierarchical relations determine how certain characteristics are inherited. These characteristics include policies and access rights. For example, applying a certain policy at management group level that restricts the configuration of a service to a specific value will result in all underlying subscriptions and resources experiencing that same restriction.

Proper subscription/management group design can facilitate cloud adoption when it meets the organisation's requirements. The structure will depend on the nature of the organisation's activities, the geographical set-up, the types (and variety) of applications/workloads, the number of workloads/applications, etc. Azure management groups are designed to be flexible so they can be used to design a management group structure that reflects the expected organisational needs. The following example in Fig. 5 illustrates how different strategies of organising subscriptions can be combined:

A minimal design would be one management group under the root management group. Under the management group one shared subscription group is made for the developers and one subscription group for the Azure support team. Later a separate management group can be added for Information Security Officers and the Security Operations Centre where the activity log files can be stored and evaluated. The applications within the shared subscription could still be separated by using different resource groups.

When a DevOps team requests their first environment (e.g. development) in Azure, at least the following is required: one resource group, one service principal account, one AAD group (to add the DevOps team members to), a DevOps project, and a pipeline to deploy services/applications in the resource group. The pipeline is connected to the resource group via a service connection. Once teams need to be onboarded to the shared subscription, soon the subscription limit of 980 resource groups will be reached. Considering the environment types per application (development, test, acceptance, and production) only 245 applications can be hosted, which might be enough for small companies but certainly not enough for large companies. From cost management perspective, organisations may also want to implement more than one subscription. As time goes by, the DevOps teams became more experienced, and their demand to be more autonomous will increase. By monitoring adherence to Azure policies, the (central) department responsible for Azure policy management over time will gain more insight into which policies need to be enforced at which level. These developments can trigger organisations towards the decision to adopt secure landing zones: these are environments where

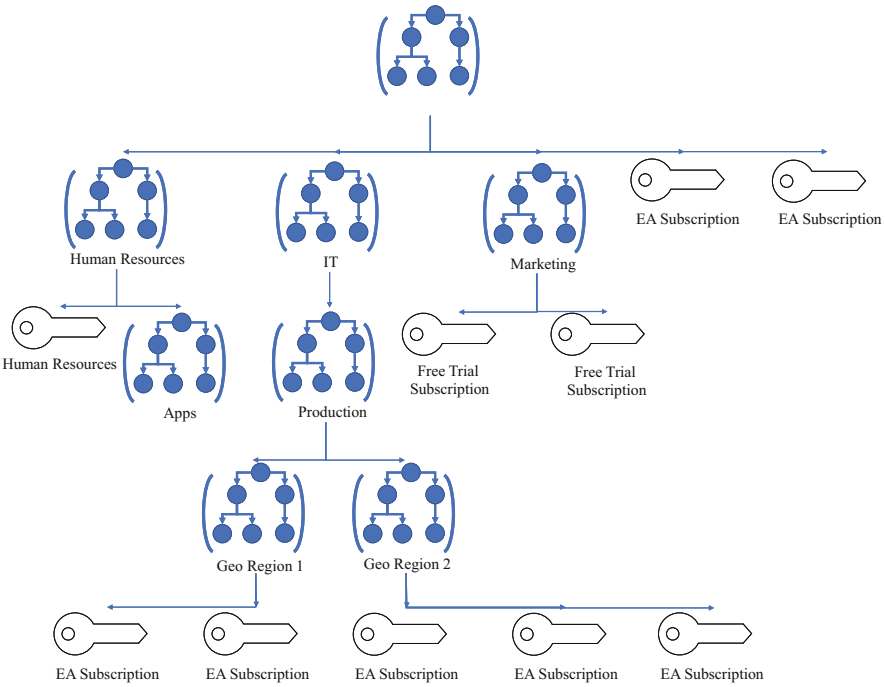


Fig. 5 Mix design strategy: departmental hierarchy, followed by geographic distinction for the Production department within IT, adopted from Microsoft (2022e)

non-customised services can be configured/deployed while the cloud controls/policies are determined at management group level (and then via inheritance will be enforced/in deny mode on lower levels).

To enable enforcement of different policies in different environments, an organisation can choose to implement a separate management group for the development and test environments and another for the acceptance and production environments. When an organisation introduces secure landing zones at a later phase (i.e. not from the start), it should account for future changes regarding management groups, subscriptions, and resource groups. Teams that already have applications in the shared subscription or in their individual subscriptions need to migrate their applications to the secure landing zone environments. Migration can only be done after all applicable policies have been implemented. Therefore, a planning for application migration needs to be made. Migration of applications to secure landing zones is not done overnight and organisation may want to avoid changes in the management group-subscription-resource group structure that require application migrations. To mitigate that risk, an IT auditor could review the design process and assess whether sufficient expertise was involved. Not only to speed up the subscription/resource group deployment process, but also to ensure that every DevOps team starts in the same position (with the same controls) the deployment process should be automated.

The IT auditor should verify that changes to the deployment process are detected and that appropriate follow-up actions are taken.

6.2.5 Network Management

Network management covers design, implementation, and maintenance of logical network configurations to enable secure communication. Connectivity, in general the ability to communicate between two points, is provided by networks. In addition, network components provide many more services like IP address resolving by Domain Name System (DNS), DDoS protection, load balancing, filtering by firewalls, intrusion detection/prevention (not necessarily all provided by Azure). From a different perspective one could say that networks provide isolation. Within Azure isolation is provided on many levels, starting with/at the tenant level. When an Azure subscription has been agreed upon and a customer/billing relationship has been established, then this Azure subscription is associated with one unique Azure Active Directory (Azure AD). By using a dedicated instance of the Azure AD tenant isolation is established; members of other tenants do not have access unless the tenant admin grants it through federation or by provisioning user accounts. Between the DevOps team subscriptions or secure landing zones there is also isolation and each must have their own dedicated range of (internal) IP addresses. Within the subscriptions virtual networks can be defined and finally there is isolation at Azure PaaS services level by firewall rules.

Typically network management responsibilities are divided into two parts. Centrally managed (in the management subscription) is the infrastructure that is shared, like circuits connecting Azure with the ABN AMRO Data Centres, more in specific Azure ExpressRoute. Also Azure Firewall and the DNS are centrally managed. Lastly, the assignment of IP-address ranges (IPAM: IP-Address Management) is centrally done. The DevOps teams usually manage local network configurations in their subscriptions like PaaS firewalls (e.g. in Azure SQL server or Azure Data Factory) and local virtual networks (VNets). Depending on the services that are used within a subscription, a VNet is required. VNets are required for among others virtual machines (VM), VM scale sets, and Kubernetes. A network security framework (NSF) that describes which traffic flows require which network controls can help organisations secure their (Azure) networks. The NSF model could utilise data classification regarding confidentiality and integrity. Roughly speaking, the higher the confidentiality and/or integrity rating, the more network controls need to be applied on specific dataflows.

To access confidential data (like account balance), besides Multi-Factor Authentication (MFA) measures, also the following measures are required: encryption, specific firewall rules, logging and monitoring, intrusion detection, data leakage prevention, etc. To access public data from the website, only a subset of these measures is required.

From an auditor's perspective, the possibility of DevOps teams to configure public endpoints and to allow access from public internet deserves attention. Besides

authentication, access control lists should be implemented to restrict access appropriately. In addition, adherence to the NSF needs to be monitored and non-compliance requires follow-up.

On some Azure services a parameter can be specified that enables bypassing all network separation rules. In practice, setting this parameter allows other tenants of the Azure cloud to access these services regardless of other firewall rules or deny public internet access settings. The parameter is 'allow Azure services' and can be specified on services like Azure SQL Server, Azure Data Factory, and Azure Synapse. The parameter is a feature and at the same time a huge security risk because it allows all users of Azure to attempt to access your service and data within that service. Valid credentials (user id and password) are the only thing that now stand between the other tenant/hacker and your data. When the parameter is misused in the development environment to circumvent connectivity problems, it will do the same in the production environment. Of course, this is highly undesired and should be prevented. Auditors should be aware of these types of parameters and verify whether the organisation has processes in place to identify and mitigate these types of vulnerabilities.

6.2.6 Security Event Monitoring

In a world of cyber threats, it is extremely important to detect/monitor events that could indicate compromise. In addition, in a highly regulated industry it is important to demonstrate compliance with regulatory demands regarding e.g. highly privileged access. In both cases, you need to record/log certain events, process/evaluate them, and take appropriate action. This is what we call security event monitoring. In principle, the DevOps teams are responsible also for arranging security monitoring, but identifying security relevant events and turning them into analytic rules in Sentinel requires a certain risk mindset/type of security awareness and specific skills that may not always be present in DevOps teams. Besides, as many teams use the same Azure services, they are likely to run the same risks. A central application security monitoring team can prove to be beneficial in developing several generic use cases that are likely to be expanded as more workloads are hosted in the cloud environment.

With powerful resources like activity logs from Azure services and tools like LogAnalytics and Azure Sentinel, one would expect the capability to correlate and monitor almost anything. And that may be true, however not out-of-the-box. With specific logging and monitoring requirements, additional measures need to be taken in order to fulfil those requirements. Logging of events may be required to fulfil regulatory requirements; however, the focus should be on events that threaten business processes, operations, and confidentiality and integrity of data. It may also be necessary to distinguish between changes made via pipelines/Everything as Code versus manual changes in troubleshoot situations.

In order to prevent alerting everything and flooding the Security Operations Centre (SOC) with false positives, in advance the organisation needs to determine

exactly what are the activities/events that need to be known and that require follow-up actions. These events may be generic (like disabling MFA for a user, elevation of privileges, making policy adjustments) and service specific (e.g. changing the access to AKV or changing TLS minimum level on SQL server). The set of events to monitor may grow over time, based on experience/new insights.

In our experience, identification of events to monitor is the trickiest part because maintenance and support staff are quite hesitant to identify security events in advance. Elevation of privileges by using Azure services like PIM is quite easy to monitor. Regarding subsequent actions, it is much harder to identify which pose a threat. Two arguments are often heard:

1. We do not know in advance which actions will be performed (using high privileges).
2. An action/event can in one situation/subscription be valid and required (e.g. viewing and updating data or configuration settings) and in another situation/subscription unauthorised. How to differentiate between those two?

The difference may be whether an incident was reported via another channel: when the changes made concern the incident, then they are probably fine. When there is no incident, then further investigation may be required.

Focusing on the riskiest events is a sensible approach. A few events may be identified in advance (like the ones mentioned above), while the rest may be based on new insights. This path is however still uncertain because it depends on vigilance of the maintenance staff to detect out of the ordinary actions. And it may not be the best way forward: e.g. for a single event it cannot be decided without additional information whether it was performed with malicious intent. From the on-premises IT landscape we already know that correlation of events is important to recognise patterns and compare them with attack tree scenarios. Within Azure, machine learning may be able to fulfil these requirements, an area definitely worth experimenting.

Once a security relevant event has been identified, it has to be figured out into which activity/log file (or a combination of log files) the event is recorded and whether the recorded data is sufficient to generate a useful/actionable alert. Next, a so-called analytic rule has to be set in Sentinel and the follow-up action needs to be determined/specified. Let us say an alert has to be sent to the SOC. The staff at SOC need to have instructions how to act on different alerts. Probably not only the SOC needs to be alerted, but also the product/application owner needs to be informed. For analysis of trends, the alerts may be aggregated.

From an audit perspective, we would expect that every Sentinel analytic rule has an owner and that a life cycle process applies to them all. As security events should occur exceptionally, their relevance needs to be determined periodically. The rules must have a documented rationale and follow-up actions must have been described. False positives need to be eliminated or at least minimised during development. The use of Sentinel comes with a bonus (in addition to machine learning): the MITRE ATT&CK framework is used within Azure Sentinel to help classify threats to the organisation and to provide quicker understanding of the level where intrusion

exists. The MITRE ATT&CK framework is a curated knowledge base and model for cyber adversary behaviour, reflecting the various phases of an adversary's attack lifecycle and the platforms they are known to target. Being able to classify threat events into the framework is a major step in demonstrating coverage and control.

6.2.7 Summary of Key Risks and Controls for Infrastructure Managed Services

In Table 7 below the key risks and controls for Infrastructure Managed Services have been summarised.

6.3 Services and Workloads

Next to the centralised functions, generic services and boundaries described in the previous section (i.e. Landing zone), DevOps teams need to set up and maintain specific Azure services that provide compute, network, and storage functionality to host the actual workloads (i.e. business applications).

Azure provides more than 200 services, which are divided into 21 categories. These categories include computing, networking, storage, IoT, migration, mobile, analytics, containers, artificial intelligence, and other machine learning, integration, management tools, developer tools, security, databases, DevOps, media identity, and web services.[4] Via the Azure portal DevOps teams can use these services to create cloud-based resources, such as virtual machines (VM) and databases for their workloads. Depending on the services being used, controls need to be implemented to adopt and use these services securely. In this section, the key control domains are described that are applicable to all consumable services. Microsoft has extensive online documentation that can be reviewed for the specifics of each service.

In this section, we will refrain from giving guidance on the audit of functional application controls (e.g. input/processing/output controls) as the audit of these functional application controls is only marginally different from the audit of these in an on-premises environment. Existing literature can be reviewed on this subject.

6.3.1 Network Configuration & Management

Network configuration is the process of setting policies, flows, and controls for an organisation's network infrastructure. It's a critical step to ensure that the application network works properly and stays secure. Within Azure it's important to create and maintain network segmentation, control inbound/outbound communication, control

[4] https://azure.microsoft.com/en-us/services/

Table 7 Risks and controls for Infrastructure Managed Services

Risks	Controls
6.2.1 Identity Management	
Unauthorised changes to AAD, product catalogue (may be ABN AMRO specific), policies, central network configuration	The Azure Ownership role should only be assigned to non-personal accounts (like SPN) and temporarily to emergency/troubleshoot groups
	Azure roles are not granted (directly) to personal user accounts
Abuse of privileged access	Access rights can only be elevated using PIM, requiring approval by peers
	Logging and monitoring of changes and timely follow-up on incidents
Access cannot be denied timely	Only groups that correspond to groups in central IAM solution may be used
Creation of local accounts (not known to AAD) with weak authentication measures	Detection of these accounts can be implemented by a combination of policies and logging and monitoring measures
6.2.2 Policy Management	
Azure built-in policies are changed by Microsoft	Monitor changes made by the CSP, assess the impact of the change, and take corrective action when required
Enforced Azure policies do not meet company rules (or are inconsistent)	Life cycle management process is implemented, rationale needs to be plausible and supported by experts, policies are approved by policy board, policies are tested before enforcing them, the complete set of policies is evaluated periodically
Enforced Azure policies do not cover all threats	Coverage is monitored by policy board or equivalent
Deployed services do not fully comply to Azure policies	Central oversight function to monitor non-compliance and follow-up
6.2.3 Product Development	
Product catalogue contains old versions (that have vulnerabilities)	Life cycle management of products to ensure that only most recent versions are available/deployable
Product catalogue does not include all services required by DevOps teams	Keep a backlog and prioritisation mechanism
Baseline not defined	Central oversight function to monitor baseline adoption and adherence
Hardening reduction	Central oversight function to monitor baseline adoption and adherence
Lack of vulnerability management	Central oversight function to monitor vulnerability scan execution and vulnerability remediation
Lack of patch management	Central oversight function to monitor unpatched services/software and severity levels; escalate when remediation measures are delayed

(continued)

Table 7 (continued)

Risks	Controls
Product descriptions obscure, incomplete, and outdated	Periodically verify with consumers/DevOps teams whether the descriptions are comprehensible and adequate
Lack of scalability/products not timely available or incompletely customised	Extend the product development teams or refrain from customisation and compensate by means of policies (which also require maintenance)
6.2.4 Subscriptions/secure landing zones	
Structure of management groups, subscriptions, and resource groups does not fit business requirements (causing costly migrations when the structure changes)	Design review by experts, test the design in separate environment using a pre-defined/agreed upon requirement list
Deployment of subscriptions is not timely and repeatable (leading to different starting positions)	Automate and monitor subscription deployment
Unauthorised changes to deployment process (leading to subscriptions with e.g. less or no controls)	Monitor changes to automated deployment process
6.2.5 Network configuration	
Usage of public endpoints and allowing access from public internet/networks (applies to several Azure services)	Central oversight function to monitor and verify whether compensating controls have been implemented
Allow Azure services = yes (applies to several Azure services)	Implement custom policy
Measures do not correspond to data classification	Monitor compliance to network security framework and take appropriate action regarding non-compliance
6.2.6 Security event monitoring	
Not all security events are (timely) identified/lack of insight in coverage of use cases	Central monitoring of security event identification, supported by the MITRE ATT&CK framework mapping in Azure sentinel to periodically assess coverage
Not all services/components are monitored with developed/applicable use cases	Central monitoring of appropriate activity logs being loaded/processed
Inadequate follow-up actions defined	Life cycle management of rules/use cases—periodically review follow-up of alerts
Rules are outdated or will never trigger an alert	Life cycle management of rules/use cases—periodical review to verify effectiveness

communications between Azure resources, route and filter network traffic. Moreover, not only should the (virtual) network be well-architected, it should also adhere to well-established principles such as layering and tiering. Each Azure service has networking configuration items (e.g. VNETs, subnets, Firewalls, IP addresses) that should be taken into account as part of securing the network.

It is important to note that there is a difference between the overall network perimeter (i.e. Landing zone, discussed in Sect. 6.2.4), and the specific network configuration of a certain application. Auditors need to take into account both configurations, specifically for the application configuration careful attention needs to be given in the network rules and settings: do these adhere to standards, are these sufficiently hardened, and periodically reviewed. Moreover, it needs to be checked where sufficient isolation and tiering is in place between applications (i.e. sound architecture): point-to-point connections should receive extra attention on this matter in terms of potential security vulnerabilities. An example to consider is Network Security Groups which in essence are a basic, stateful, packet filtering firewall, that controls access based on the configuration of source IP address/port number, destination IP address/port number, and the protocol in use. Just as important is the implementation and configuration of Azure Firewall which is a fully stateful firewall service with built-in high availability and unrestricted cloud scalability. There are a lot more network measures that can be implemented depending on the requirements of the environment. It is important that the auditor first understands the network design (e.g. via documentation and flow-diagrams) and implementation and whether this is fit-for-purpose. The next step would be to check and verify each measure and solution.

6.3.2 Identity & Access Management

Identity & Access Management (IAM) ensures that the right users have the appropriate access to Azure services and resources. Azure has many capabilities that can help secure IAM, such as: Single sign-on, Multi-Factor Authentication (MFA), Azure role-based access control (Azure RBAC), Security monitoring, alerts, and machine learning-based reports, Privileged identity management, Identity protection, etc. (Microsoft, 2022d). Every service on Azure makes use of an identity alongside certain privileges that needs to be controlled.

The auditor should keep in mind that next to the centrally managed identities and accounts (i.e. Landing zone), certain Azure services and applications have their built-in accounts and identities. Similar to traditional audits, (non-personal)/(privileged) accounts should be reviewed and checked by the auditor against the principle of least privilege, adherence to periodic access reviews, and the implementation of strong authentication (e.g. enablement of MFA).

There are four fundamental built-in roles within Azure (Azure RBAC): Owner (full access to all resources), Contributor (create and change resources but can't grant access to others), Reader (read/view only), User Access Administrator (manages user access to Azure resources). The auditor needs to understand the use of each of these roles for the specific application and determine whether its use is controlled and appropriate. Another point of attention for the auditor could be the reports about administrator access history and changes in administrator assignments. The auditor can make use of a variety of reports within Azure to gain insight into the controls around IAM and how the organisation is operating: e.g. via sign-in anomaly reports,

user-specific reports which display device sign-in activity data for a specific user, activity logs containing audited events within certain timeframes (24 h, 7 or 30 days, etc.).

6.3.3 Resource Security

Azure services need to be secured just like any other resource. Depending on the type of service being consumed (e.g. IAAS/PAAS/SAAS), patching needs to be performed and endpoint protection (e.g. virus/malware protection) should be in place. Additional security measures include disk encryption, secure data transfer between resources, and adequate key management.

For the auditor it is important to note that the burden of maintaining resource security by the IT organisation is the most for IAAS (e.g. managing *all of the resources* within the Virtual Machine). For PAAS certain resources are taken care for by the cloud service provider and for SAAS this part is less applicable as the CSP is typically fully responsible for resource security. In the case of IAAS, the auditor should consider auditing the whole VM and all of its contents (as this is not managed by cloud service provider), this means general IT controls testing on the Operating System, Middleware, and database as all of these components are managed by the IT organisation. Key controls include: change management, lifecycle management, patch management, vulnerability management, system hardening management, etc.

For PAAS, the auditor needs to understand the PAAS-components that are managed by the IT organisation, typically this translates to configuration settings on networking (e.g. which components are allowed to communicate with each other?), admin access (e.g. who, what, when, and which conditions apply?), and hardening (e.g. legacy/weak protocols allowed?).

Depending on the Azure configuration policies set throughout the organisation, the auditor needs to perform more or in-depth testing of controls. This means that in the case that Azure policies are not globally applicable and enforced with no override possibility, the auditor needs to consider testing each Azure resource (e.g. product/service) relevant to a certain application as this could potentially deviate from security best practices. As mentioned in the previous section, DevOps teams may enjoy a certain degree of autonomy and freedom within their specific block and subscription which allows them to have less than optimal implementations.

6.3.4 Logging and Monitoring

Logs are event records where events related to the state of a specific Azure service are collected. There are a multitude of logs (e.g. performance, integrity, availability) for different Azure services. Selecting useful information to store and archive is key here: selecting metrics, rules, classification of alerts *for each service*. It's also important to ensure the security and confidentiality of stored logs, and control the quality of log data by analysing and adding missing information to logs.

Monitoring is also important to detect any lack of service performance and to detect attacks in real time. In order to detect these anomalies, Azure provides centralised supervision tooling to aggregate the different logs and to enable real-time monitoring (e.g. Microsoft Sentinel, Defender for Cloud, Azure Monitor, etc.). Of course, each service needs to be connected and configured to use the centralised tooling, and the tooling itself needs configuration and maintenance as well.

Although certain monitoring can be arranged centrally (refer to previous section), for each application and set of resources managed by DevOps teams, certain events can be logged and monitored. It is important that the auditor keeps in mind that both dimensions should be taken into account. For example, flow logs can provide insight in network traffic patterns. There are roughly three categories of logs within Azure: Control/management logs (e.g. create/update/delete of azure resources), Data plane logs (e.g. events raised as part of Azure resource usage for example via Windows event system, security, and application logs in a virtual machine), and Processed event logs (e.g. provide information about analysed events/alerts, examples of this type are Microsoft Defender for Cloud alerts). Finally, the auditor should also take into account that all of these logs need to be monitored in some shape or form via the monitoring solution. Next to performance, attention areas for the monitoring solution include the security posture of virtual machines, networks, storage and data services, and applications to discover and prioritise potential security issues. Azure provides extensive logging and monitoring capabilities for DevOps teams that can be utilised for each application and the resources involved.

Microsoft's Defender for Cloud continually assesses Azure resources for security issues and presents the results on a dashboard in the Azure portal. The recommendations vary from low to high severity and could be grouped into categories like: System updates should be installed, Log Analytics agent should be installed on virtual machine scale sets, Vulnerability assessment should be enabled on SQL servers, Authorised IP ranges should be defined on Kubernetes Services, etc. When security issues have been identified, Defender for Cloud gives recommendations how to improve and remediate these issues. IT auditors should be careful in interpreting these issues, especially concerning their validity: the tool makes no difference whether resources in a development or production environment are assessed, but for the risk profile this makes a difference. Furthermore, the tool may not 'see' compensating controls that are not based on Azure services/features, e.g. using Splunk instead of LogAnalytics or using DDoS protection from third parties. No doubt Defender for Cloud provides added value by identifying weaknesses in configurations, but the recommendations should be regarded with due caution. In addition to security issues, Defender for Cloud also provides statistics on regulatory compliance like ISO27001 and PCI/DSS. From the auditor's perspective it is worthwhile to retrieve this information to determine how DevOps teams are managing the environment.

Azure Monitor provides a comprehensive solution for collecting, analysing, and acting on telemetry from cloud environments which gives several 'insights' or views on the resources (metrics) on the one hand and operational alerts and access to LogAnalytics on the other hand. These insights regard applications, but also VMs,

containers, network, storage accounts, and a few others and can be tailored to specific needs. The DevOps teams need to determine which events require operational monitoring and how to respond to alerts and incidents. Just like vulnerability management operational monitoring may not be obvious to all DevOps teams. Especially when availability requirements are 7 × 24. A word of warning seems applicable when using Azure Monitor. Performance problems caused by badly written queries, or not timely reorganised (SQL) database indexes may be obscured or compensated by scalability measurements. Due to lack of production workload limitations, performance problems may not always directly surface. Also, performance problems can originate from Microsoft incidents as well. In February 2022, performance problems were encountered in Europe with the Azure DevOps service: Boards, Repos, Pipelines, and Test Plans were all affected.

From an audit perspective, availability of applications/business functionality is one of the key aspects. Typically, Azure Monitor is restricted to the Azure cloud environment and is therefore not implemented as an end-to-end monitoring solution. Therefore, additional measures should also be taken into account by the auditor. Depending on the application functionality and the Azure components used, a sensible selection of parameters to monitor have to be made. DevOps teams must be able to demonstrate their monitoring controls and explain their selection parameters.

6.3.5 Security Incident Response

Security Incident Response is about developing and implementing an incident response infrastructure (e.g. plans, defined roles, training, communications, management oversight) for quickly discovering an attack and then effectively containing the damage, eradicating the attacker's presence, and restoring the integrity of the network and systems.

It is important again to distinguish centralised operations (landing zone-level) as well as decentralised operations (application-level). The auditor needs to be aware that at both levels, runbooks need to be developed and periodically tested. Moreover, it is key to verify whether the involved teams have the right capabilities to handle incidents and how well the communication takes place between and across teams. Existing literature on this topic should provide sufficient guidance on how to assess this process.

6.3.6 Data Encryption

The main areas of encryption include encryption of data-at-rest, data-in-transit, and key management with Azure Key Vault. Data encryption at rest is available for most of the Azure services including file, disk, blob, and table storage. Microsoft also provides encryption to protect Azure SQL Database, Azure Cosmos DB, and Azure

Data Lake. The auditor should be aware of this option and depending of the services being used verify if this encryption is actually enabled.

Another point of attention is that Azure supports various encryption models, including server-side encryption that uses service-managed keys, customer-managed keys in Key Vault, or customer-managed keys on customer-controlled hardware. With client-side encryption, the customer manages and store keys on-premises or in another secure location (encryption performed outside of Azure). The three server-side encryption models offer different key management characteristics that the auditor should be aware of in order to assess the appropriateness of the implementation being used:

1. Service-managed keys (combination of control and convenience with low overhead).
2. Customer-managed keys (gives customer control over the keys, incl. Bring Your Own Keys (BYOK) support, or allows you to generate new ones).
3. Service-managed keys in customer-controlled hardware (customer manages keys in their repository, outside of Microsoft control, configuration is complex and most Azure services don't support this model).

The auditor should pay close attention to the Key Management process and Key Storage solution (e.g. Hardware Security Module). There are different options available for Key Storage and each solution has its certain pros and cons depending on the requirements of the organisation. Key requirements to check are tenancy (multi or single), integration possibilities (SAAS/PAAS/IAAS), supported key operations (public/private; key-lengths; ciphers), scalability/availability, FIPS-140 level support and certification, level of control over keys (full/partial/none) and compliance with regulations, and operational responsibilities (backup/restore, patching, upgrades, etc.).

Data-in-transit can be secured via various ways, some examples that the auditor could verify are whether the site-to-site VPNs are properly set up, SSH and RDP sessions are set up to use protocol encryption, REST API calls make use of HTTPS, and whether the TLS protocol is used to protect data between services.

6.3.7 Business Continuity and Disaster Recovery (BCDR)

Two factors are especially important for the resilience of an application: its availability (the proportion of time the application is functional) and recoverability (the ability to recover from failures). Although availability of Azure services is guaranteed for up to 99.95%, things can and will go wrong. The high availability of Azure services does not dismiss organisations of the responsibility to take measures to guarantee that applications (which most likely are supported by a combination of services) and data are safeguarded from outages. The measures consist either of implementing redundancy or the ability to quickly recover.

Azure services run on servers in datacentres across the globe. These datacentres are grouped into availability zones, and availability zones are grouped into regions

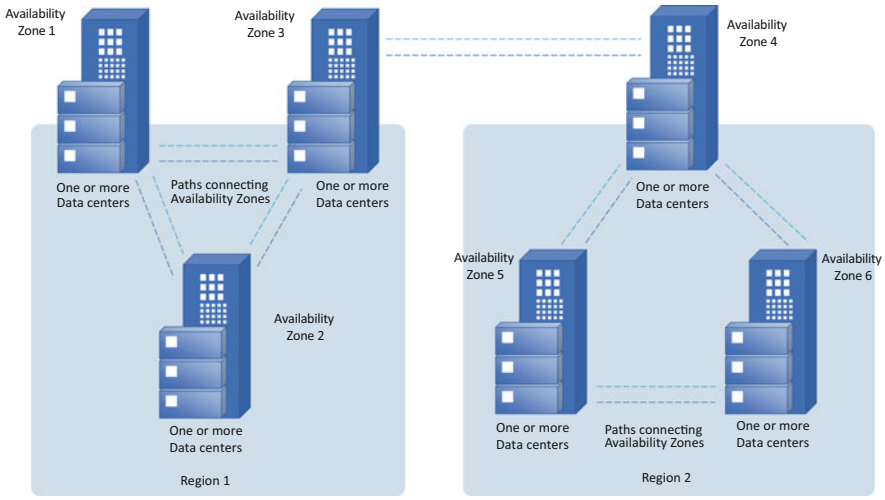


Fig. 6 Azure data centres, availability zones, and regions according to Microsoft (2022b)

such as North and West Europe. The datacentres are connected through a dedicated regional low-latency network. Azure Availability Zones are physically separate locations within each Azure Region that are tolerant to local failures. Failures can range from software and hardware failures to events such as earthquakes, floods, and fires. Tolerance to failures is achieved because of redundancy and logical isolation of Azure services. To ensure resiliency, a minimum of three separate availability zones are present in all availability zone-enabled regions. This design per region is outlined in Fig. 6.

Resilient solutions can be designed by using Azure services that use availability zones. The services can be divided into zonal, zone-redundant, and always-available services. The zonal services can be deployed to a specific, self-selected availability zone to achieve more stringent latency or performance requirements. Examples are Azure Backup, Azure Site Recovery, and Azure Virtual Machines. Resiliency is self-architected by replicating applications and data to one or more availability zones within the region.

With zone-redundant services, resources are replicated or distributed across zones automatically. For example, zone-redundant services replicate the data across three zones so that a failure in one zone doesn't affect the high availability of the data. Examples of services are Azure SQL, Azure Storage Account, Azure KeyVault, and Azure Data Factory.

Always-available services are always available across all Azure geographies and are resilient to zone-wide outages and region-wide outages. Examples are Azure Active Directory, Azure Policy, and Azure Portal. Always-available should be taken with a grain of salt, because in 2020 and 2021 Microsoft experienced several Azure AD outages.

Table 8 Risks and controls services and workloads[a]

Risks	Controls
Inappropriate access to data	– Identity & Access Management (e.g. multi-factor authentication, access reviews, segregation of duties, etc.)
	– Network configuration and management (e.g. VPNs, network segmentation, firewalls, etc.)
Exposure of confidential data in-transit and at-rest	– Secure key storage, adequate key management processes
Inability to mitigate and/or recover from data loss/exposure/manipulation	– Logging and monitoring (e.g. key events, alerts, follow-up procedures)
	– Security incident response (e.g. training, testing, documentation)
Compromised integrity of resource	– Resource security (e.g. patching, endpoint protections)
Unavailability of data and systems	– Appropriate zoning of data for compute and storage activities
Incomplete/inaccurate/invalid records	– Application controls e.g. (input/output/processing controls)

[a] This table is a high-level summary, refer to the above paragraphs for the key differentiating aspects related to the controls

It is evident that when an application consists of several Azure services, it is not so easy to achieve RTO and RPO values on the application level. When availability and performance are not critical, these measures probably are sufficient. However, for applications that are critical, e.g. financial transaction processing, performance objectives, and RPO = 0 will be difficult to be met and need careful consideration.

6.3.8 Summary Key Risks and Controls for Services and Workloads

In Table 8 below, the key risks and controls for services and workloads have been summarised.

6.4 Processes

Most processes that will be in scope of audits that relate to traditional (non-cloud) IT environments are also relevant in a public cloud context. Change Management, Problem Management, Deployment Management, and Capacity Management are just a few examples. As there are many sets of best practices, guidelines, and audit programs available for IT auditors to audit IT-related processes, we will refrain from covering this here extensively. However, there are a few subjects that are worth mentioning as they require specific attention from IT auditors.

Configuration management and the configuration management database (CMDB) that are at the core of most service management processes, must also cover the cloud environment. Usually, the public cloud in question will be able to deliver details on the configuration items, tags and dependencies by using the standard configuration management facilities offered by the CSP (e.g. Resource Graph, Azure Service Map, and Azure Application Map). Configuration management data can then be extracted and synchronised with the CMDB. In many cases, it will be possible to use native integration facilities, but...the configuration management data needs to fit the CMDB data model and that will usually require a lot of effort to make all the necessary adjustments. One of the recommendations therefore is to use the CMDB as the main tool for cross platform insights, but to use the CSP CMDB as centre of truth for deployed resources instead of fully syncing with the CMDB.

It is not the intention to cover the DevOps way of working as it represents a way of working that is closely related to public cloud, but it is a completely different subject nonetheless. However, there are two DevOps subjects that every IT auditor must be aware of when auditing a public cloud implementation in a DevOps context, as they will significantly impact the level of control over public cloud implementations.

The first subject pertains to the high level of autonomy and freedom of the DevOps teams that ideally work in a self-service model in the public cloud. This means that every process might be executed differently by different DevOps teams. This does raise the question: How does the organisation ensure that the DevOps teams work within the boundaries as set out in the corporate policies and standards. Based on our experiences, this can only be achieved if three interrelated conditions are met: First, there needs to be a set of goals for the DevOps teams that strikes the right balance between run and change responsibilities.[5] Second, (senior) management must direct and redirect the DevOps teams based on the actual performance on these balanced goals. This will support the culture that is needed to stimulate the right behaviour, i.e. that DevOps team members do not favour Development over Ops activities or vice versa. This also implies that senior management needs to be committed to these goals themselves. Although this sounds obvious, given the fact that many DevOps teams and their senior managers will have their roots in the Software Development area and that they are under pressure to deliver functionality for their (business) product owners, there is a risk that development activities get priority over operations and support activities. And third, this system can only work if the right management information regarding the goals and boundaries is available. This will require using reporting tools to frequently give insight in process performance for all processes and for all DevOps teams.

[5] According to the DevOps Research Assessment report 'State of DevOps 2021' by Google Cloud, there are four metrics of software delivery performance that can be considered in terms of throughput and stability. These metrics are the lead time of code changes (that is, time from code commit to release in production), the deployment frequency, the time to restore a service after an incident, and the change failure rate. According to that report, high performers score consistently higher on all four metrics.

The second item pertains to the fact that relying on the formal handover procedure between change and run teams is no longer possible when working in a public cloud/DevOps context. The so-called segregation of duties mechanism that relies on conflicting interests between run and change teams no longer exists if run and change activities are carried out in the same team. And it turns out to be difficult to ensure that mitigating controls operate effectively, as the privileges of DevOps team members enable them to bypass many of the theoretical controls. For example, the correct use of automated CI/CD pipelines could enable automated security tests, unit tests but also deployments under dual control. This would mitigate the lack of segregation of duties. However, it is inherent to the DevOps way of working that team members can adjust pipeline code/building blocks. In other words: without additional measures, DevOps teams could turn off dual control as part of the deployment process, security testing as part of the development process, etc. This is something IT auditors should be aware of and must consider when auditing the chain of change-related processes.

6.5 Policies and Standards

Policies hold sets of formalised rules, principles, and minimum control requirements that must be in place to direct behaviour, actions, and decisions in an organisation. Policies are generally based on laws and regulations or added requirements the organisation may be subject to or may subject itself to. They will generally be set in line with the organisation's risk appetite.

Standards are an extension to one or more specific policies and must always be consistent with these policies. Standards are used to describe detailed mandatory requirements, criteria, calculations, or methodologies associated with the implementation, enforcement, and support of the policies.

When auditing public cloud implementations, it is therefore necessary to assess the coverage and quality of the policies and standards that pertain to public cloud. Depending on the organisation's preferences, there might be cloud-specific policies and standards. Or the organisation might have decided to keep the policy framework more abstract and only outline high-level requirements that are applicable irrespective of the platforms in question. Nevertheless, the policies and standards should provide the IT organisation involved with clear direction and boundaries. They should make it clear what is acceptable use of public cloud technology and which controls must be implemented, depending on the specific situation. For example: A cloud policy might have rules that point out whether the use of public cloud is allowed for critical or regulated workloads—and if so—under which conditions. Cloud standards will give more detailed rules as to how the implementations must take place and which specific controls must be implemented.

6.6 Governance

There are several definitions for the term 'Governance'. In the ABN AMRO organisation, it is primarily defined as the activities that are aimed at providing direction (mission, vision, strategy, and goals), putting the organisation in place that will work to efficiently achieve the strategic goals, and that ensures that the organisation and its staff are held to account.

For audits on public cloud implementations, this implies that the different elements of governance should be assessed. The organisation should have a sharp vision of the role of public cloud. This will link to the boundaries set in the companies' policies. In some cases, the vision of the role of public cloud will be reflected in a specific document that outlines the platform strategy. In practice, auditors should verify that the vision sufficiently supports decision-making. For example, is it clear which types of workloads are allowed to land on the public cloud. And if there is more than one public cloud that is being used: Which types of workloads must land on which public cloud?

Furthermore, the goals of the implementation of public cloud should be clear and the management control system should be aligned with these and support accountability. For example, if the implementation is primarily aimed at cost reductions, does the management control system ensure that cost levels are measured and reported on and that it is clear who has been accountable and responsible for these cost levels?

One specific element of governance relates to the requirement (European Banking Authority, 2019) to have appropriately documented plans for the exit from arrangements with Cloud Service Providers that will enable the organisation to exit the arrangement without undue disruption to the business activities. A distinction can be made between the exit strategy and the more concrete exit plans. The European Banking Federation/Cloud Banking Forum has issued a technical paper (European Banking Federation, 2022) that gives guidance to create a common understanding as to the requirements for the exit strategy and exit plans. In the exit strategy, the organisation should include the identification of an alternative solution/provider, and on a strategic level, which threat scenario could ultimately lead to an exit being triggered. It should furthermore contain an overview of the roles and responsibilities, the human and financial resources that are required to execute the exit and the high-level timelines.

With regard to the concrete exit plan, in our opinion this should not just be a more detailed version of the exit strategy. There should be an exit plan for every workload that has been implemented on the public cloud (component 3 'Services and Workloads' in our framework) and one for the Infrastructure Managed Services (component 2 in our framework) separately. Main reason for this is that the exit requirements can vary per criticality of the service or application in question. These plans should take into account the limitations of the alternative solutions (e.g. the services used might not have a good alternative) and they should describe

the steps required to take the data from the service provider and transfer them to alternative providers or back to the organisation.

7 Discussion

As will be clear from the description in Sect. 6, auditing public cloud implementations has many similarities with traditional IT auditing. The subject matter requires specific knowledge on cloud technology in general and the architecture and services of the CSP that this concerns specifically, but the control objectives will be identical and so will most of the control domains. However, there are also some noteworthy differences that require special attention and a different approach that could also have an impact on the required audit resources, both qualitatively and quantitatively. These are elaborated on in the following paragraphs.

7.1 Manual Versus Automated Controls and the Impact on Audit Procedures and Costs

While management of traditional non-cloud environments rely on a combination of manual and automated controls, for public cloud environments, due to the high level of automation and the use of standardised services, they mostly rely on (semi-) automated controls. Typically, these services include out-of-the-box dashboards, metrics, and security baselines. This allows for a shift from distributed/siloed systems to centralised administration (policies/configurations), oversight and control. Consequently, audit procedures will contain more data analyses, which can even be scripted/automated. As more control testing is automated and less manual controls need to be tested, less auditors are required to perform these audit procedures while coverage will usually increase.

However, this is offset by the fact that many companies use more than one public cloud or a combination of private cloud or on-premises systems and public cloud computing. In this situation, even more audit resources are required as more audit terrain is to be covered. Furthermore, one should realise that using public cloud involves outsourcing activities to cloud service providers, and that requires auditors to sufficiently cover these outsourcing arrangements and associated governance and procedure in the audit plan.

7.2 Control over Public Cloud Environments Versus On-Premises IT

The on-premises IT landscape is typically managed by dedicated groups of engineers, and responsibilities between application development and support is usually separated from platform maintenance and support. The high degree of specialisation and sense of responsibility for each on his own area/terrain makes it possible to establish secure and highly available environments. One would expect that cloud environments that are used to develop and host applications are more secure and better controlled because of the potentially strong central control possibilities over the entire environment, such as policy management, continuous monitoring of the implementation/configuration of services, and security event monitoring. Compared to a pluriform on-premises IT landscape, where for each platform a separate set of controls needs to be implemented and central oversight is hard to gain because organisations need to gather and harmonise the data themselves (or connect to central systems like IAM systems, CMDBs, and Splunk), a cloud environment such as Azure at least holds the promise of better control.

But especially while transitioning to the public cloud, there are a few important risks that must be considered. First, the DevOps teams that originated from the former application development and support teams now also need to assume platform maintenance responsibilities, something that they are not accustomed to. Consequently, there is a good chance that these new responsibilities get overlooked. Second, they need to get acquainted with the new (cloud) platform with different services, a new (DevOps) way of working and associated organisational changes and pressure to migrate/transform/rebuild applications during the transition to public cloud. This could be too much to absorb for the teams in question to also keep their environment/application secure. Third, especially in the beginning, finding the right balance between software development and application/platform maintenance and operations tasks is a challenge. Chances are that some of these activities will not get the priority they need. This could manifest itself by configuration/baseline deviations, policy non-compliances, inadequate resource life cycle management and lack of vulnerability management and patching. Azure services like Defender for Cloud and Policy Manager enable organisations to identify many of these shortcomings but these are always easily remediated. Fourth, although Azure supplies powerful services to manage the environment and resources, not all enterprise requirements can be easily met. For example, several access control requirements (e.g. access on least privilege basis) cannot be enforced by one single policy. Another example on access controls pertains to the mapping of authorisations to functions by using authorisation matrices. Unfortunately, there is no automated way to verify these 'soll' requirements against the actual implementations ('ist'). And on data leakage prevention: This requires data labelling, which may be aided by Azure Machine learning, but otherwise is a manual activity. Once labelled (assuming that it will not change), monitoring measures must be developed and implemented. These examples show that considerable effort has to

made to enable the control requirements to be enforced, which is quite similar to the work required to control a traditional on-premises IT landscape.

7.3 Public Cloud and DevOps

The implementation of public cloud technology and the introduction of DevOps often go together (Google Cloud, 2021). Although these two implementations should reinforce each other, there are also disadvantages to it. DevOps teams are relatively autonomous, and the general expectation is that these teams will take full responsibility for their workloads (and—depending on the situation—also the underlying platform). In practice however, the maturity level differs between the teams and—consequently—not all teams are able to keep their environments 'clean', i.e. are able to configure all components or services correctly and keep them up-to-date and patched. This might also be caused by the lack of targets that strike the right balance between run and change tasks and that drive priorities of the teams. The way the teams are then consistently (re)directed by line management will affect their behaviour and performance.

Azure supplies monitoring capabilities but these cannot remediate these issues. To address them partially, one could consider—at least temporarily (in the first period after transitioning to the cloud and into a DevOps way of working)—centralising platform maintenance/platform operations tasks. This would relieve some of the burden of the DevOps teams and give them the opportunity to grow into their role.

7.4 Relevance of the Distinction Between IaaS and PaaS

The shared responsibility model distinguishes between IaaS and PaaS services to make clear where maintenance responsibilities lie. But to what extent is this distinction relevant for IT auditors?

We can imagine that a company would only use IaaS services and build all additional functionality themselves or implement third party software on their virtual machines. In that case many cloud/Azure control measures will not apply (e.g. Defender for Cloud most probably does not know these third party products and Azure policies will not apply. In addition, security event monitoring must be configured largely separately). As the IaaS deployment model comes closest to an on-premise environment, many of the benefits of public cloud will not be enjoyed. For example, the benefits from service features like scalability, elasticity, and site recovery will not be available for organisations that just use IaaS services. However, it will give the highest level of control over what is implemented when and where exactly and it gives the organisation the highest independence of the CSP (which could be beneficial if an exit from the CSP needs to take place). It also requires the

most additional measures to make the environment secure. Besides general IT controls (e.g. logical access, hardening, vulnerability management, and patch management) you may expect to test cloud-specific controls that pertain to the IaaS services used.

Let us elaborate on this a bit further. Typically, IaaS services at Azure are categorised as compute, network, and storage. When we look at compute (hosting services responsible for hosting and running the application workloads), the following services are available: Azure Virtual Machines (VMs), Azure Container Service, Azure App Services, Azure Batch, and Azure ServiceFabric. There are VMs for Windows and Linux. Now suppose that you run the application on Linux, then the aforementioned general IT controls also apply to that Linux VM. Nothing new because you probably already knew Linux from your on-premise IT landscape.

When the Azure service is not familiar to the organisation because it does not have a counterpart on on-premise implementations, such as probably Azure Batch, then you would probably also not use it in the cloud. But if you do, then from an audit perspective you would probably look at the same aspects that are covered by the general IT controls. Because after all it is just software that provides functionality. No matter how magical the cloud services sometimes may seem because they are unparalleled in the on-premise domain, it is software that was coded (with potential flaws that need to be patched) and can be configured (which may affect hardening). In other words, the Azure services may look different from what you are used to, but in essence the same general IT controls apply. That does not mean that nothing changes in the audit practice. When you as an IT auditor have access to the Azure portal with read access on most resources, you will have to get used to the interface, get acquainted with most used services, learn to use services like the Policy Manager or Defender for Cloud, get a feeling of where critical settings are to be found, etc. We can tell from experience that it is another world in appearance, and it takes time to get used to.

If it were possible to use only PaaS services, then your audit activities would change compared to only IaaS, because the number of components that you cannot 'see' increases. For example, Azure SQL Database is a fully managed platform as a service (PaaS) database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring without user/customer involvement. Azure SQL Database is always running on the latest stable version of the SQL Server database engine and patched OS with 99.99% availability (Microsoft, 2022a). All the PaaS services have in common that regarding the general IT controls you can no longer assess hardening measures, vulnerability management, and patch management, because they are not under the customer's control. The CSP takes care of these and if you want assurance on how they do it, you have to rely on external certifications or carry out an audit on the relevant CSP activities.

So, which audit activities remain? In this case and most probably in general they would pertain to logical access controls and data controls. Regarding access, you would like to verify whether the required access levels, described in e.g. an authorisation matrix, are actually enforced and cannot be circumvented. It should cover all types of access by users, administrators, and applications/NPAs, including emergency/troubleshoot access. Regarding confidential data you would expect

encryption of data in-transit and at-rest. The encryption measures should comply to your (and regulatory, e.g. GDPR) requirements, which could mean that you would have to assess cryptographic key management measures performed by your company as well. When availability requirements are high, you would have to verify whether the appropriate measures have been taken. In this case, e.g. turning on Azure SQL geo-replication and making sure that your Allowed Location policies apply. Maybe regulatory retention periods apply to your data. Then you would have to implement additional measures to meet those. The IT auditor must assess whether the design is adequate and whether the measures are operationally effective.

Most probably, application developers/DevOps teams will use a combination of IaaS and PaaS. In that case of course the beforementioned considerations regarding IaaS and PaaS apply. But you would need to make the distinction in order to be able to decide which controls you need to test. The first control would be to verify whether only the services described in a solution design are deployed in a specific resource group. Chances are that in time more services are used/added then originally foreseen and documented.

In principle all the services in that resource group need to be assessed. From each you have to determine whether it is IaaS or PaaS before you can start assessing the applicable controls per service, considering company and regulatory requirements. Regarding data flows you may need to verify whether they are as designed/required and whether network security measures comply to your requirements (like our internal network security framework).

When availability requirements are high, it is important to establish per service used which measures apply (because they can differ per service) and whether they have been implemented adequately. Based upon a view on the individual services, you can assess impact on application availability.

When performing an audit, the distinction between IaaS and PaaS is very relevant. In view of the way things organised at your company, e.g. with centrally managed infrastructure services and (decentral) DevOps teams responsible for application development, it makes sense to divide audit activities as well. The audit department is traditionally mirrored to the organisation and therefore facilitates the division. Application auditors focus on the deployments/workloads/applications and the auditors assigned to IT infrastructure focus on the infrastructure managed services. Typically, with every application audit, an auditor from the infrastructure team participates. This is beneficial because it stimulates knowledge cross-pollination and allows better understanding of the relevant aspects, which will ultimately result in better risk assessments and audit engagements.

7.5 Managing Costs

One of the strengths of CSPs is that they provide services on a pay-per-use basis. To support this feature, usage of every service needs to be metered. The customer has

access to these statistics and the costs. Usage/costs can be viewed from different angles and on different levels.

From an audit perspective you would not only be interested in whether a targeted cost reduction on company level was met, but also whether productivity increased and time-to-market was reduced. Additionally you could assess whether budgets for application development or development of new features have decreased, compared to the actual costs and when there are differences, whether the appropriate measures have been taken.

In our experience the expectations may be too high. When your organisation, or at least the application development part, has a high level of maturity, the DevOps teams are used to the new way of working, they have perfect understanding of the cloud services and are stable in composition (i.e. no or low attrition), then your chances of realising your targets are the best. But from the list of requirements you can already deduce that for companies embarking on a cloud journey most probably these requirements will not all be met, certainly not from the beginning. Should you give DevOps teams carte blanche at the start? That is completely at the other end of the spectrum and probably nobody would agree. For sure DevOps teams need to experiment and learn how to use the services and what their features are. This will take time and resources and it would not be fair to expect the same productivity from a starting team as from an experienced team.

Of course, CSPs can only measure consumption of their services. That however, are not the only costs of application development. Companies hire staff or outsource functions, have management costs, provide their staff with working places, laptops and mobile phones, etc. So it would be an oversimplification to say that migration to a cloud environment would give you more control over your IT costs. That only holds true for the cloud services consumption part.

In addition, productivity of DevOps teams is only very indirectly related to cloud resource consumption. You can measure usage of services but you must not confuse that with the development efforts to provide functionality. Suppose a new feature needs to be added to an already developed application that consists of a number of (IaaS and PaaS) services and additional application code. The design, development, testing, and deployment efforts will consume cloud services but are no indication that functional requirements have been met. At best, higher resource consumption during development may indicate complexity.

Probably, the challenge to predict development time and effort to realise application functionality (and thereby DevOps team productivity) in cloud environments does not differ much from traditional environments, but this would be an interesting topic to explore.

8 Conclusions

We briefly described the rise of public cloud computing and the initial hesitance to adopt public cloud technology by the financial services industry (Sect. 1). Next, we elaborated on cloud deployment models (private, public, and hybrid) and service models (IaaS, PaaS, SaaS) to generally set the scene for audit activities (Sect. 2). In Sect. 3, publicly available audit frameworks and work programs were evaluated in terms of suitability for usage for audits. Section 4 presented the case study of the IT/Cloud transformation of our organisation and in Sect. 5, the audits activities that we performed were presented, which formed the basis for our conceptual framework (Sect. 6). In Sects. 6.2 and 6.3 we have provided examples of concrete/detailed controls regarding commonly used cloud services configuration that can help as a starting point for audits.

Although the look and feel of cloud environments differs hugely from traditional IT landscapes, we came to the conclusion that the audit attention points are largely similar. Therefore the execution of an audit will differ in components and configurations to cover, but risks remain largely the same. The implementation of controls will differ, because cloud environments offer other/new tools and services.

Compared to traditional on-premise IT landscapes, the level of control for a number of areas can be higher in cloud environments. That can be largely attributed to the environment having a uniform basis and being able to have general oversight via maintenance/management tooling. Regarding preventative controls, the same policies can be enforced on all subscriptions and resource groups, which is a very strong control. However, designing and implementing the appropriate policy set can be challenging. Likewise, regarding detective controls, the range of vulnerability scanning and security event monitoring can be across all your subscriptions and resource groups. But also these have their pitfalls: you have to evaluate reported vulnerabilities for applicability and you have to identify security events. Although, the latter may be compensated by machine learning in the near future.

While every customer environment and DevOps team can be different, from enterprise control perspective it can be rewarding to centralise the following functions:

1. Identity Management
2. Policy Management
3. Subscription Management/Secure landing zones
4. Network Management
5. Support for implementing Security Event Monitoring

The organisation size and auditor experience/education are key factors to consider before engaging in cloud audits. Knowledge of technologies, products, and services is essential and larger audit teams are better equipped to facilitate cross-learning between auditors.

References

Akdeniz, D., Bani Hashemi, S. J., Putters, J., & Yavuz, A. (2020). *Pooled audits on cloud service providers—Part 1*. Retrieved from https://www.deitauditor.nl/business-en-it/pooled-audits-on-cloud-service-providers/

Amazon. (2021). *Shared responsibility model*. Amazon Web Services, Inc. Retrieved March 23, 2022, van https://aws.amazon.com/compliance/shared-responsibility-model/

Amazon. (n.d.). *What is cloud computing*. Amazon Web Services, Inc. Retrieved March 23, 2022, from https://aws.amazon.com/what-is-cloud-computing/

Association for Financial Markets in Europe. (2019, November). *The adoption of public Cloud Computing in capital markets*. Retrieved from https://www.afme.eu/Publications/Reports/Details/The-Adoption-of-Public-Cloud-Computing-in-Capital-Markets

Axelos. (2020). *ITIL4*. Retrieved from https://www.axelos.com/certifications/itil-service-management

Bani Hashemi, S. J., Putters, J., & Yavuz, A. (2020). *Pooled audits on cloud service providers—Part 2*. Retrieved from https://www.deitauditor.nl/business-en-it/pooled-audits-on-cloud-service-providers-2/

CSA. (2021). *Cloud controls matrix*. Retrieved from https://cloudsecurityalliance.org/research/cloud-controls-matrix/

European Banking Authority. (2019, February). *EBA guidelines on outsourcing arrangements*. Retrieved from https://www.eba.europa.eu/eba-publishes-revised-guidelines-on-outsourcing-arrangements

European Banking Federation. (2022). *Cloud exit strategy—Testing of exit plans*. Retrieved from https://www.ebf.eu/wp-content/uploads/2020/06/EBF-Cloud-Banking-Forum_Cloud-exit-strategy-testing-of-exit-plans.pdf

European Securities and Markets Authority. (2020, December). *Guidelines on outsourcing to cloud service providers*. Retrieved from https://www.esma.europa.eu/press-news/esma-news/esma-publishes-cloud-outsourcing-guidelines

Gartner. (2021, August 2). *Gartner says four trends are shaping the future of public cloud* [Press release]. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2021-08-02-gartner-says-four-trends-are-shaping-the-future-of-public-cloud

Google Cloud. (2021). *Accelerate State of devops 2021*. Google Inc. Retrieved from https://services.google.com/fh/files/misc/state-of-devops-2021.pdf

Haes, S. D., Grembergen, W. V., Anant, J., & Huygh, T. (2015). COBIT as a framework for enterprise governance of IT. In *Enterprise Governance of Information Technology: Achieving Alignment and Value, Featuring COBIT 5* (2nd ed., pp. 103–128). Springer. https://doi.org/10.1007/978-3-030-25918-1

Institute of Internal Auditors. (2018). *Auditing Third-party Risk Management-supplemental guidance—Practice guide*. Institute of Internal Auditors.

ISACA. (2020–2021). *Cloud Computing Management Audit Program*. Retrieved from https://store.isaca.org/s/store#/store/browse/detail/a2S4w000004KoH1EAK

ISACA. (2020–2022). *Azure Audit Program*. Retrieved from https://store.isaca.org/s/store#/store/browse/detail/a2S4w000004KoGTEA0

Jones, E. (2021). *Types of Cloud Computing—An extensive guide on cloud solutions and technologies in 2021*. Retrieved from https://kinsta.com/blog/types-of-cloud-computing/

Mell, P., & Grance, T. (2011). *The NIST definition of Cloud Computing*. National Institute of Standards and Technology. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Microsoft. (2022a, February 5). *What is the Azure SQL Database service?* Microsoft Docs. Retrieved March 23, 2022, from https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview

Microsoft. (2022b, March 1). *Azure regions and availability zones.* Microsoft Docs. Retrieved March 23, 2022, from https://docs.microsoft.com/en-us/azure/availability-zones/az-overview

Microsoft. (2022c, March 1). *Shared responsibility in the cloud.* Microsoft Docs. Retrieved March 23, 2022, from https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility

Microsoft. (2022d, March 18). *Azure AD built-in roles.* Microsoft Docs. Retrieved March 18, 2022, from https://docs.microsoft.com/en-us/azure/active-directory/roles/permissions-reference

Microsoft. (2022e, March 21). *Organize your resources with management groups*. Microsoft Docs. Accessed March 18, 2022, from https://docs.microsoft.com/en-us/azure/governance/management-groups/overview

Microsoft. (n.d.). *Tour of Azure services.* Microsoft Docs. Retrieved March 23, 2022, from https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/tour-of-azure-services

Monterie, A. (2020). *ABN Amro maakt geslaagde migratie van AWS naar Azure*. Retrieved from https://www.computable.nl/artikel/achtergrond/cloud-computing/7039237/1444691/abn-amro-maakt-geslaagde-migratie-van-aws-naar-azure.html

Rosa, J., & Dee, M. (2020). *Transformation at ABN AMRO Bank*. Retrieved from https://www.agilealliance.org/resources/experience-reports/devops-transformation-at-abn-amro-bank/

Process Mining for Detailed Process Analysis

Mieke Jans and Manal Laghmouch

1 Introduction

Processes are an important part of a company's daily operations. They are the core to creating value for the end user, both internally and externally. Where organizations are functionally organized around departments such as purchasing, marketing, production, sales, and finance, it is the processes across these departments that ensure smooth operations. For example, the production process will grind to a halt if no goods are purchased, resulting in a purchase need. This information stems from the production planning where a purchase request is registered, but will not just stay within this department. Presumably the purchasing department will take over and place an order with a specific supplier. This is then shared with the warehouse as well as the production department and the finance department. Based on this information, each department will take further action (e.g., approve orders, receive goods, pay invoices). In this example, a clear start and end point can be defined, along with a fixed set of activities performed to achieve a certain goal: registering the need to purchase, registering a purchase order, approving the purchase order, receiving the purchased goods, receiving the invoice, and paying the invoice. This is a typical example of a business process that runs across the various functional departments within an organization, as most processes do.

In general, a process is initiated by a particular need and ends by fulfilling that need. The purchasing process example starts with the need to purchase goods and

M. Jans (✉)
Research group Business Informatics, Hasselt University, Hasselt, Belgium

School of Business and Economics, Maastricht University, Maastricht, Netherlands
e-mail: mieke.jans@uhasselt.be

M. Laghmouch
Research group Business Informatics, Hasselt University, Hasselt, Belgium
e-mail: manal.laghmouch@uhasselt.be

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_9

ends with the goods being purchased. To fulfill the need, a set of activities is performed in a logical sequence. A business process is typically defined as follows:

> A business process is a set of activities performed in a coordinated manner with a specific business goal in mind.

The defined sequence of activities is part of an organization's set of processes, which are often modelled and portrayed in a process model. A process model depicts how the process (according to the organization) should run.

The remainder of this chapter is structured as follows. In Sect. 2, we discuss process modelling and analysis by describing the modelling languages that are commonly used and introduce the field of Business Process Modelling and Process Mining. Section 3 describes the core principles of process mining and the required input for process mining analyses. Section 4 explains how internal and external auditors can use process mining in practice. In Sect. 5, we conclude this chapter.

2 Process Modelling and Analysis

In this section, we look at fundamental concepts related to process modelling and analysis. We start by explaining the relevance of process modelling and analysis by introducing model-based process analysis. Next, we describe the types of process modelling languages that are commonly used. We conclude this section by introducing a family of data-driven process analyses: process mining.

2.1 Model-Based Process Analysis: Business Process Management

Business processes present how things should be handled within an organization. They represent the organization's daily operations and form the basis of improvement opportunities. The discipline that focuses on designing, executing, analyzing, monitoring, and improving business processes is Business Process Management (BPM) (Dumas et al., 2018).

Business Process Management (BPM) focuses on the modelling, implementation, monitoring, and improvement of business processes. It provides a structured methodology with the goal of continuously improving processes: the BPM life cycle, as shown in Fig. 1. The BPM life cycle consists of five activities, after identifying the business processes that are present in the organization. The first activity is *process discovery*. This leads to uncovering the actual process, as performed within the organization. This is classically based on available documentation and interviews. Process discovery within BPM results in an as-is process model. It is important to identify this model to perform the next step: *process analysis.* Process analysis generates insights on possible process improvements. These improvement options

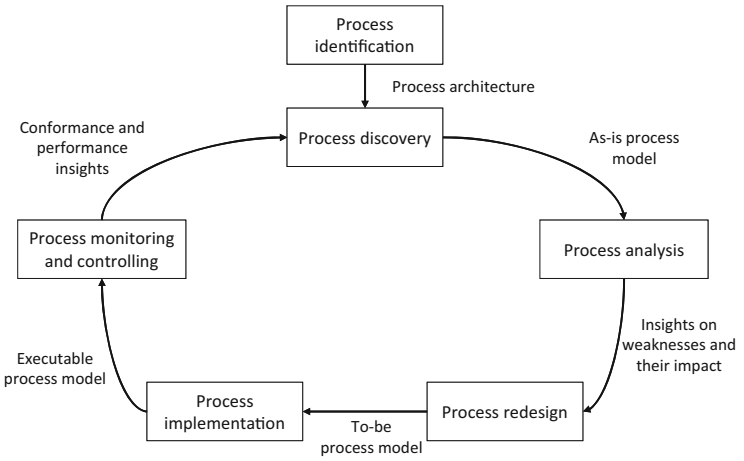


Fig. 1 BPM lifecycle. (Source: Fundamentals of BPM, 2nd ed., 2018, Dumas et al.)

can relate to both increased operational efficiency and better coverage of potential risks. Based on these insights, a *process redesign* follows: the process design is reviewed and adjusted where possible. A new process model is born: we call this the to-be process model. The adjustments associated with the to-be process are implemented in the next step. Both the configuration of the information system and the instructions to the parties involved are adjusted to the new process design. When this new process is put into use, the next step will be *process monitoring and controlling*. This activity will generate new insights on top of the existing documentation. This can be used as input for a new cycle that starts with the mapping of the current as-is process model.

In the traditional interpretation of the BPM lifecycle, there is a striking separation between process models on the one hand and process data on the other. The activities *process discovery, analysis, redesign,* and *implementation* are often based on process models (in textual or graphical form). In contrast, the *process monitoring and control* activity is often data-driven: key figures of the process are monitored and analyzed. In the purchasing example, this could have referred to the number of open orders and blocked invoices. However, there is no default interaction between the process models and the automatically generated process data. Data on how a process is actually executed is often not taken into account.

Process analyses based on documented process descriptions and interviews are called model-based analysis techniques. Although these techniques provide interesting insights, they have a number of limitations. For example, the quality of the analysis depends on the quality of the available process descriptions and a model-based analysis does not provide valuable insights when the models do not match reality. A mismatch between the model and reality can have several causes. A model is an ideal image of reality or a guiding tool. Consequently, a model is often a simplistic representation of a desired situation in which, unlike in reality, no

exceptions occur. In addition, processes can change unintentionally over time and (rather subjective) discussions with different people involved in the process can result in different models. All these elements call for a new approach to analyze business processes: an objective and realistic approach—process mining. Before getting into the topic of process mining, we discuss two different ways of modelling business processes.

2.2 *Process Modelling Languages: Procedural vs. Declarative*

A well-organized enterprise models the core processes needed to achieve an organization's business goals. Process modelling has numerous advantages; it creates a picture of where a company places its emphasis. Furthermore, it provides guidance to actors involved in the process and focuses on the goals to be pursued. The design of well-thought-out processes prevents operational inefficiencies on the one hand and integrates desirable control mechanisms on the other. Through these two elements, correctly followed processes translate into value creation as well as risk reduction (Dumas et al., 2018).

Broadly speaking, there are two different approaches to capturing a process model. A process can be described procedurally or declaratively. A *procedural approach* means that all possible process executions are specified exactly in the model. An example of a procedural process model can be found in Fig. 2. Figure 2 is an example of a procedural process model for a purchasing process (in BPMN). This model specifies that there are only three different process paths for the execution of a purchase:

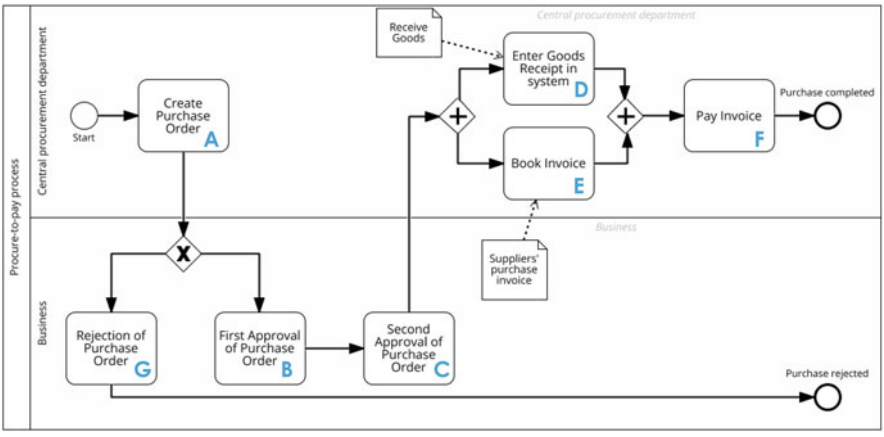


Fig. 2 Example of a procedural process model for a purchasing process

Table 1 Sample list of rules to describe a buying process

If ...	, then ...
"Pay invoice" is present	"Book invoice" is present
"Approve order form—2nd level" is present	"Approve order form—1st level" is present
"Reject order form" is present	"Book invoice" is not present
"Approve order form—1st level"	Executor "Approve purchase order—1st level"
and	$\neq$
"Approve order form—2nd level" are present	Executor "Approve purchase order—2nd level"

A-B-C-D-E-F
A-B-C-E-D-F
A-G

Any other implementation is a violation of the process model. Often, a procedural approach fits well with the modelling of highly structured processes.

Several modelling languages exist within the procedural approach. Although, in the past, flowcharts and EPC models have found their way into business, there are numerous drawbacks associated with these types of models. These drawbacks are mainly about the ambiguous model interpretation and the specific language dependence of software (Dumas et al., 2018). As a solution, a standard was developed for procedural process modelling: Business Process Modelling and Notation. The BPMN standard was created by the Object Management Group (OMG), which is an independent party that develops system-independent standards for computer systems.[1] Process models drawn up according to this standard are easy to interpret. At minimum, a process consists of activities in rectangles, arrows, and additional semantics to indicate relations, like parallelism and choice relationships. For example, Fig. 2 contains a parallelism of activities D and E, indicated by a diamond with a plus, and a choice after activity A, shown as a diamond with an X.

The second way to describe a process is through a *declarative approach*. In a declarative process model, relationships between activities are determined by rules. An example of such a rule is as follows: "the activity *register order* always takes place before the activity *approve order*." The basic principle of declarative modelling is that a process may be executed in any way, given that certain rules are followed. Rather than capturing the process in fixed paths from start to finish, the total set of rules then defines the process. Opposed to a procedural approach, a declarative approach is recommended for less structured, flexible business processes. Table 1 provides an example of a set of rules that could describe the process from Fig. 2. Depending on how many rules are included in such a set, the process is defined more or less constrained. For example, working with partial (and therefore multiple) deliveries would violate the process model in Fig. 2, but would not violate the rules in Table 1.

[1] https://www.omg.org/bpmn

A declarative process model is a set of business rules that describe the constraints that a correct process execution should adhere to. A framework that formalizes and standardizes the declarative modelling approach is DECLARE. With DECLARE, a set of formalized constraints using Linear Temporal Logic can be written (Pesic et al., 2007; Van der Aalst et al., 2009).

2.3 Data-Driven Process Analysis: Process Mining

Process mining is an umbrella term for all data-driven process analysis techniques. It brings together the disciplines of data mining and BPM to gain insights into business processes. Process mining allows analyzing a set of data, in particular to better understand operational processes and enterprise activities. The input of a process mining analysis is an event log. Such an event log contains the automatically generated data during the execution of a process. It is comparable to an audit log which is structured in a specific way. This log is used to obtain a realistic representation of the actual process during the *process discovery* activity. Unlike a traditional approach to this activity in the BPM lifecycle, an approach via process mining shows the actual process performed along with the process variants that took place (instead of the desired (normative) process model) (Van der Aalst, 2016).

Process mining techniques that are relevant within the audit can be divided into two groups: "process discovery" and "conformance checking." Process discovery embraces techniques that discover process models from structured process data. These techniques start from the data stored in the information system during process execution to discover process patterns in this data. The discovered patterns are then visually represented in a process model. This provides an objective representation of the actual process performed and can be used to identify improvement opportunities. Conformance checking goes a step further by testing the conformance of the actual process against the normative process model or against business rules. When conformance is determined by comparing discovered actual process executions to a normative process model, the result is an overview of mismatches between the actual and the normative process model. When conformance consists of checking business rules, the result is an overview of transactions that do not conform to the business rules. Figure 3 visualizes process discovery and conformance checking. You can think of these two types as the core of a process mining analysis.

2.4 Six Phases Within a Process Mining Analysis

While performing a process mining analysis as part of an audit, there are typically six phases that are completed.

Every process analysis starts with *the construction of an event log*. An event log is a specific structured data file that minimally consists of case identifiers, related

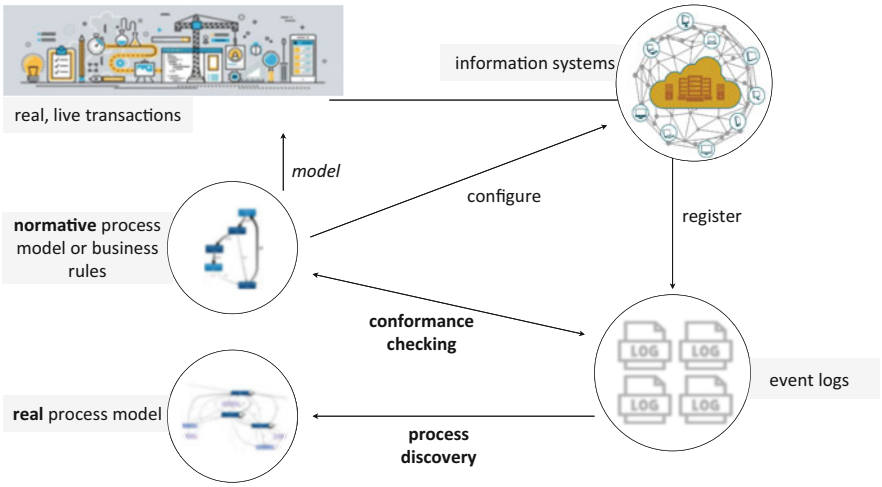


Fig. 3 The core types of process mining

activities, and the timestamps of transactions in a certain process. This will be discussed in more detail in the next section.

When an event log is available, an analyst will apply *process discovery* to discover the actual process in the form of a process model. The output of process discovery can provide initial insight into how structured a process is or is not.

To gain more insights, a *conformance check* is usually performed. This can be done in two ways: either actual process executions are compared to a (procedural) *process model* or to *a list of rules* (declarative approach).

Once there is more insight into how often and where in the process deviations are made from the prescribed process, a *variant analysis* can be performed. A "variant" is a path that is followed for at least one process execution. The variant analysis allows for a closer examination of the different ways in which the process has been followed in reality. For example, an information system may have logged 1000 purchases, 600 of which have followed a similar path (and thus belong to the same variant of the process). Let's call this path variant A. The remaining 400 purchases follow a different path and belong to variant B. For example, variant A looks like this: <"create purchase order", "approve purchase order—1st level", "approve purchase order—2nd level", "register goods receipt", "book invoice", "pay invoice">. Process executions in variant B, on the other hand, follow a different path of activities: <"create purchase order", "reject purchase order">. Variant A and B represent two ways the process was carried out. In reality, there will be many more variants than stated in the previous example. An examination of the different variants of the process will provide many insights. A variant analysis can, amongst others, answer questions on which process path is most frequently followed and on how many different ways the process was carried out.

Finally, a process mining analysis ends with a *case analysis*. During this analysis, specific characteristics of certain transactions are examined in depth. For example,

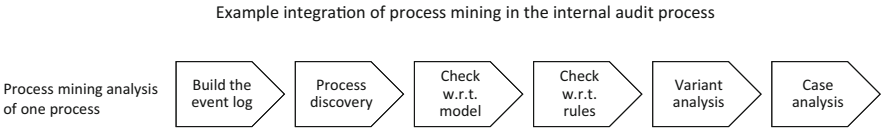


Fig. 4 The phases of a process mining analysis during an audit

one can zoom in on the process performances linked to a certain document type, a certain supplier, a certain period, or a certain level of materiality. This is an in-depth analysis at the level of a subset of transactions.

Figure 4 visually depicts the six elements of a process mining analysis.

3 Requirements and Core Principles of Process Mining

In this section, we will look more closely at what is minimally required to engage in process mining and what the basic principles of process mining algorithms are.

3.1 The Event Log

The most important step of a process mining analysis is to collect the right data in the right format, the event log. This section describes what information should be contained in the event log and which structure is required. As briefly mentioned in the introduction, an event log is a structured file that contains all relevant data of a process. In other words, it is a log of events (also called actions) that make up the process and forms the input for a process mining analysis. It is therefore important to know what it takes to build a high-quality event log.

An event log combines data that may come from different information systems in an organization. The raw data from these systems, as automatically stored during process executions, is the starting point of the event log. Often this data is stored in different systems or at least in different tables that are connected via references. Combining the relevant data, selected from thousands of tables, requires a lot of effort, time, and expertise. Think for example of the business processes supported by a SAP© or Oracle© ERP system. Since such an implementation can consist of tens of thousands of tables, some knowledge is needed to know where to find the right process related data. Identifying relevant event data and converting it to a structured event log is not a task without effort. Therefore, it is important to work with someone who has the right knowledge about the information systems and data in the company to build the event log. Nonetheless, in this chapter we give the basics so that you can assess whether a process analysis based on event data might be possible within a certain organization.

Table 2 Process mining terminology

Concept	Description	Example
Process	A systematic, structured series of actions carried out with the intention of achieving a particular goal.	The delivery process in a postal company.
Case	An instance (execution) of a process.	The execution of the delivery process of goods to Mrs. Lize Kelders who received her goods on April 15, 2021.
Activity	An action to be performed in the process.	The creation of a delivery note in the system.
Activity instance	The specific execution of an activity at a specific time for a specific case.	The creation of delivery note D1001 in the system on April 4, 2021 for delivery 1001.
Event[a]	An atomic part of an activity instance.	The departure of delivery 1001 (the arrival of delivery of 1001 would be a different event; both events are part of the activity instance "delivery 1001").
Time	The time related to an event.	The 30th of January 2021
Resource	The performer of an event.	Mr. Jan Thomas.

[a] In administrative business processes, the events and activity instances are often the same. For example, actions such as "approve," "register," or "book" do not have separate start or end moments. Therefore, in this chapter we often use the term "event" to refer to an activity instance, as this increases readability

The construction of the event log depends on the type of questions you are trying to answer through a process mining analysis. For the example purchase process, on the one hand you might want to answer questions about the flow of purchase orders over time. On the other hand, you might also be interested in the flow of purchase invoices to gain insight into the company's invoicing. Although both questions sound similar, they require different event logs because they look at the process from a different perspective: from the perspective of purchase orders and from the perspective of purchase invoices.[2] To better understand the content and construction of an event log, it is important to become familiar with terminology used in process mining. Table 2 lists the most important terms related to process mining and an event log with a description of their meaning.

Table 3 shows, for clarification, an excerpt from an event log of a sales process. The given event log consists of the following six columns: "Case ID," "Event ID," "Timestamp," "Activity," "Resource," and "Value (in €)." Each row in the event log represents an event and belongs to the execution of a particular case, which in this example is a sales order. The given excerpt shows the events of three cases. Case 1 consists of four events that are already arranged chronologically. Event 51425446101 describes the creation of a sales order (activity) with a value of € 2000 by Jan (resource) on 13 April 2021 at 12:00:00 (time). Linking all events for

[2] For details on the technical construction of an event log, please refer to the report "From Relational Databases to Valuable Event logs" by Prof. Mieke Jans of Hasselt University in Belgium, freely available on LinkedIn.

Table 3 An excerpt from an event log of a sales process

Case ID (Order ID)	Event ID	Timestamp	Activity	Resource	Value (in €)
1	51425446101	13/04/2021 12:00:00	Register sales order	Jan	2000
	45458455102	14/04/2021 09:23:12	Approve order	Ines	–
	45689454103	19/04/2021 21:34:33	Send goods	Lies	–
	45856655104	20/05/2021 08:23:11	Receive payment	Ahmed	1700
2	45545886291	15/04/2021 17:25:00	Register sales order	Jan	30,000
	77496564292	16/04/2021 01:35:12	Approve order	Ines	–
	85452211297	16/04/2021 22:00:01	Approve order	Peter	–
	85625444301	17/04/2021 07:43:15	Send goods	Lies	25,000
3	87,312,542,221	11/03/2021 09:34:54	Register sales order	Marie	1500
	87,312,542,222	26/03/2021 09:37:11	Receive payment	Ahmed	1500

case 1 results in one specific execution of the sales process, from the point of view of the sales order. The process started for case 1 on April 13, 2021 with the creation of an order by Jan and ended with the receipt of payment on May 20, 2021. The sequence of the four listed activities in this specific order reflects one process variant. In this excerpt, this variant is not repeated. However, it might emerge later in the event log that this variant is the most frequent variant of all the process executions.

At a minimum, an event log contains information about the case IDs, activities, and related times (columns 1, 3, and 4).[3] Based on these minimum requirements, process mining is able to represent the real flow of actions over time. In addition, an event log can contain additional information about events, such as the resource and value in this example. In a process mining context, these properties are called attributes. You can add as many attributes as desired to the event log. Note that the more attributes you add, the larger (broader) the event log becomes. It is therefore recommended to only include those attributes that add value to your process analysis. As a consequence, it is important to, as a preparatory step, unambiguously identify business questions that the process analysis should answer.

[3] A standard has been developed for event logs, XES (https://xes-standard.org). This format is system and software independent and is used by most process mining tools on the market as a log format. Often there is also the possibility to load the log as a csv file and the tool itself converts this into an xes file.

To make an initial assessment of whether or not a process mining analysis is possible, it is important to ask the following questions:

– Is it possible to designate a case to follow through the process?
  Note that a "purchase" or "sale" is not a suitable candidate, as this is usually not stored as a separate entity in the information system. Underlying documents such as a "purchase order" or an "invoice" may be more appropriate.
– Is it possible to link the activities that make up the process execution of the case to the chosen case?

Only if these two questions can be answered positively, you can further consider a process mining analysis to answer your business questions. For more information on this specific topic, we refer the interested reader to Jans (2019) and Jans et al. (2019).

3.2 Process Discovery

After the event log is built, process mining analyses can be performed. As already mentioned, process discovery is the first analysis that is performed. It aims to represent the process as it was actually performed within an enterprise. An event log of one specific process is the input for such process discovery analysis. Based on the event log, process discovery can then discover a set of process models that together reflect the actual business process. In what follows, the mechanism behind process discovery algorithms is explained, as well as the possible outputs.

3.2.1 The Mechanism Behind Process Discovery

As described in Sect. 2.1, an event log consists of at least three elements: a case ID, an activity, and the time an event was recorded. These three data points are necessary to visually represent the flow of a process. Figure 5 visually represents the mechanism behind process discovery in a simplified way. On the left side of the figure, an event log consisting of three cases is represented. The cases in the log go through a number of activities. To explain the process discovery mechanism as simply as possible, the figure abstracts from the time when the activities occurred. For this example, it may be assumed that the activities are arranged in chronological order. A process discovery algorithm starts by identifying the path of each individual case. For example, case 1 follows the path <A, B, C, D, D, E>. Case 2 follows a different path, which is <A, B, C, D, E> and the path of case 3 looks like this: <A, C, B, D, E>. Finally, the algorithm combines the paths and learns patterns that can be visualized. Each of the previous three paths starts with activity A. Then, activities B and C follow activity A. In two cases, B follows first and then C. In the other case, it is the other way around. It is inferred that the order of these two activities is of secondary importance. After the execution of B and C, activity D takes place, either twice or not. The process ends in all observed paths with activity E. The combination

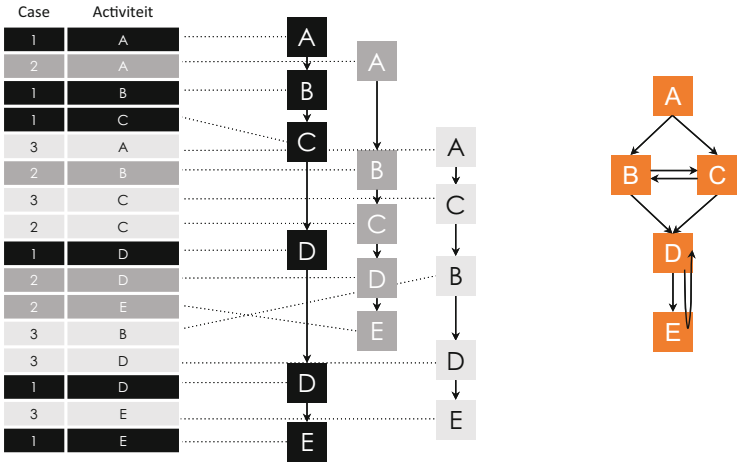


Fig. 5 A simplistic illustration of the mechanism behind process discovery. (Source: Process Mining Book at https://fluxicon.com)

of these discovered patterns results in a process model that reflects the process that is actually followed. Figure 5 shows this process model in orange.

3.2.2 Levels of Abstraction

Process discovery outputs a set of process models that together represent the behavior captured in an event log. Although a set of process models are the output of process discovery, they are often presented "on top of each other" in one process view (as in Fig. 5). The end user can determine the level of abstraction of the given process model. You can compare this principle with a dynamic road map that can zoom in or out. To get a general idea of how a route network is structured, an overview of the most frequently used roads, usually motorways, is sufficient. An abstraction is then made of other roads that do exist and are used, but less intensively. Following this analogy, a high level of abstraction is also desirable in an initial introduction to the process. Getting an understanding of how the process works in most cases is sufficient. If the end user is interested in more details, then a low level of abstraction better suits the analyst's needs. In our road map analogy, local roads, and possibly even bike lanes are included in the map.

Figure 6 shows two levels of abstraction of the same process, departing from the same event log. The model on the left is a more high-level representation of the process: it is more abstract than the model on the right. Depending on the purpose of the process mining analysis, one level of abstraction fits better than the other. Throughout the analysis, the abstraction level can be changed by zooming in and out on the process (process mining software allows for easy variation in abstraction level).

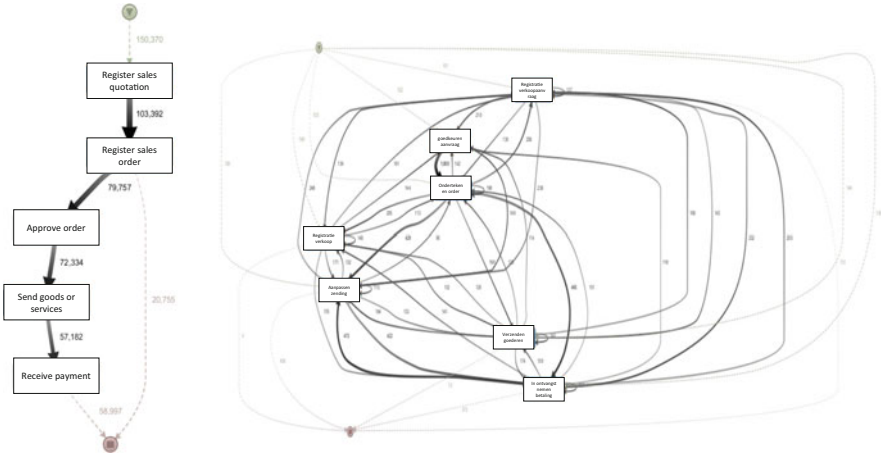


Fig. 6 Two different levels of abstraction of process discovery output

3.2.3 Output

The most commonly used visualization in process mining software is the "process map." It shows the activities in rectangles and connects these with arrows if one activity (in the event log) is followed by another activity. The more often this relationship is observed, the thicker the arrow. This is called a "directed graph." Although the modelling language is very intuitive, it consists of ambiguous relationships. Take for example the process as discovered in Fig. 5. It is not clear whether after activity A both activity B and C follow, or whether only B or C is sufficient, or whether many repetitions of B and C must follow. The core of the problem with this process representation lies in not being able to represent parallelism and choices unambiguously. On top of that, there may be combinations of arrows in the model that are not actually present in the event log. Given these shortcomings, an output according to the BPMN standard is a good addition.[4]

3.3 Conformance Checking

Whereas process discovery provides insights about the actual processes within an organization, conformance checking can identify where the actual process matches or deviates from prescribed procedures or business rules. Conformance checking compares actual process behavior (as recorded in the event log) with procedures, either in the form of a process model or business rules. Through this comparison, process deviations are identified. Identified process deviations can result from two

[4] More and more software packages are providing this functionality.

causes: on the one hand, they can be exceptional cases that require a different approach than the standard process execution. On the other hand, a process deviation can be the result of errors or fraud. To determine deviations, there are two possibilities, as mentioned before: a test of the actual behavior (as contained in the event log) against the normative process model or against business rules. Both possibilities are briefly explained.

An event log and a normative process model are required to perform a conformance check against a model. There are several approaches to technically perform a conformance check, but we limit ourselves here to a description of the underlying principle. For a check against the model, each case in the log is played back on the process model representing the desired process. For each case, it is determined whether or not it conforms to the model. A case that deviates from the model is a case whose activities do not run through the model completely or incorrectly. The amount of detail given as output depends on the technique used.

There are naive and advanced techniques to perform a conformance check. The naive techniques only show which cases deviate from the model. Advanced techniques go a step further by providing additional information about where exactly things go wrong and why that step is identified as a process deviation. Thus, the output of a conformance check against a normative model is a list of deviating cases or a list of more detailed process deviations. To illustrate, take a case—order 201—in which the activity "*send invoice*" is missing. A naive check will indicate that order 201 is not conforming to the model, while an advanced check will indicate that there was no invoice sent to the customer for order 201.

In addition to a check against a normative process model, the actual process behavior from the event log can be compared with business rules. For this, the business rules must be converted to a formal language.[5] The set of business rules forms the declarative process model. The event log is then tested against the set of rules. If a case from the event log violates a business rule, then that case does not conform to the declarative model. The advantage of a check over rules is that it is known exactly why a case deviates: a case is not compliant with the process because rule X and Y were violated. Furthermore, the analyst can establish the rules that are of principle interest to check. By its nature, this approach leans close to the work of a financial auditor.

4 Process Mining in the Audit

The insights flowing from a process mining analysis form a good basis for improving business processes in terms of efficiency and risk. It gives a view on the level of control an organization has over its operations. Given the auditor's responsibility to understand a client's environment when performing a risk assessment, process

[5] The declarative language LTL is appropriate for this purpose.

mining can be a good support (Jans et al. (2013), Jans et al. (2014). In what follows, we discuss how process mining can support the audit.

Since the added value of a data-driven process analysis is broadly applicable, process mining is often implemented by the organization itself as part of the internal audit (Chiu and Jans, 2019). The internal auditor has more resources and time to perform a comprehensive analysis, possibly even on a continuous basis, than the external auditor. Moreover, for the external auditor the investment of a process mining analysis is relatively large compared to a total audit engagement. We will therefore first discuss how process mining can be incorporated within the internal audit. However, the principles are the same for the external auditor. Finally, we expand on the use of the process mining implementation by the external auditor.

4.1 Process Mining and the Internal Audit

An internal audit is an indispensable element for checking the internal organization of a company. The internal auditor systematically examines whether the working methods and business processes of the company are efficient and under control. This usually consists of five steps. The internal auditor starts by drawing up a multi-year audit plan. Afterwards, the internal auditor plans the process audit, carries out the audit and communicates the results. Finally, the results are followed up. The most comprehensive implementation of process mining is one that is woven into all of these steps. In what follows, we describe what this might look like. This includes an abstraction of the size of the investment required to accomplish this.

During the *planning the audit schedule* for the next few years, an event log could be created of each process in the company. These logs can be analyzed via process discovery to get a global overview of how structured the processes are. To do this, the different logs are examined with a fixed set of parameters of the algorithm (such as the level of abstraction). A process model that looks very orderly will likely represent a process with less risk than a process model with many more activities and arrows. In addition to the visual aspect, a number of metrics can be listed and calculated to determine the schedule. Examples could be: the number of cases, the number of different variants per 100 cases, and the number of variants needed to cover 80% of the event log. All of this information together can assist the auditor in creating the audit schedule.

While *planning the process audit*, the process discovery analysis from the previous phase can be repeated for the selected process. In this phase, the discovered process can be examined in more detail. Where in the previous step only the structure of the process was looked at, attention is now paid to the logic shown in the discovered process model. Does this process broadly correspond to the expectations? What paths and activities emerge when we lower the level of abstraction? Without seeing much detail, an arrow between “Create order form” and “Pay invoice” can already be an indication to investigate further in a later stage.

In addition to process discovery, an initial conformance check can be performed against the procedural model. This provides an initial, general view of the anomalies that occur. Are there possible explanations for variants that occur frequently in the event log, but that conflict with the model? Anomalous variants that occur frequently but for which the auditor can formulate an explanation are, for example, variants that can be expected in the occurance of partial deliveries. Whether these are actually partial deliveries will need to be verified later during the audit. All these aspects are included in the plan of the audit to be performed. It is a different scenario if the auditor sees a deviating variant for which no explanation can be formulated. This too will be included in the audit plan, but perhaps with more investigative actions.

After planning, the *audit* will be *conducted* and the event log will be examined in more detail. The triggers for further analysis that were uncovered in the previous phase will now be included. Indeed, the deviations from the process model must be cleared out: are they alarming deviations or are they logical process variations? To do this, the auditor will iteratively develop explanations for the identified deviations and then verify these explanations (Jans and Hosseinpour, 2019). In the example of partial deliveries, the auditor may approve the variants that include multiple deliveries, noting the assumption of partial deliveries. Separate analyses can verify that controls such as the 3-way match have worked effectively. Another anomalous example might be the absence of a goods receipt. One possible explanation for this is that it includes services, not goods. To clear this up, the auditor can take all the variants (and their cases) in which a goods receipt is missing and then test whether these cases were services. The cases for which this was the case will be "cleared," while for the other cases another explanation should be sought. Perhaps there are certain suppliers for whom other agreements apply or where the delivery takes place at a different place and this is not recorded in the system? For each explanation, the auditor will try to clarify the discrepancies based on the data. This will require a combination of variant analysis, case analysis, and rule checking. If no explanation can be given for certain deviations, the case will be included on a list of potential anomalies.

During the *communication of the results*, visual support for process mining analyses will play a particularly powerful role. The different phases of the internal audit and how process mining can support it are visually summarized in Fig. 7.

4.2 Process Mining Interaction Internal and External Audit

In addition to providing support for the internal auditor, a process mining analysis can add value for the independent external auditor. By analogy with Fig. 7, Fig. 8 shows how process mining can support the external auditor. Here, it may or may not be possible to build on what the internal auditor provides.

By using process discovery to visually represent the actual business processes, along with a first comparison of the log data against the desired process model, the

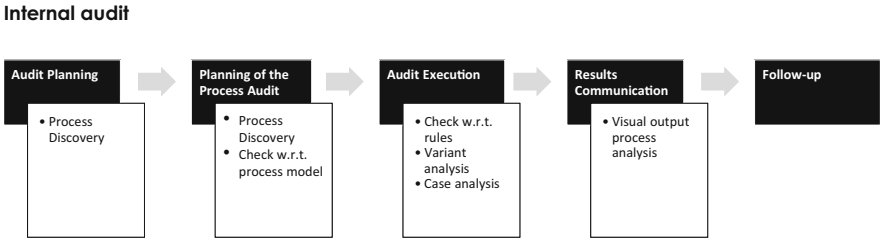


Fig. 7 Integrating process mining into internal audit

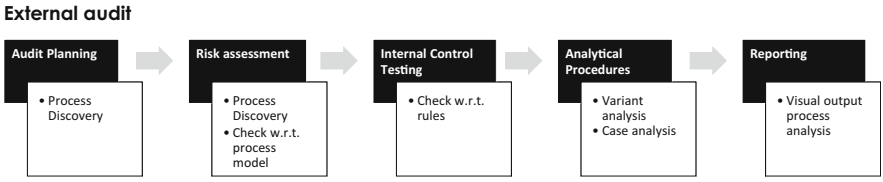


Fig. 8 Integrating process mining into external audit

auditor can obtain an initial overview of the business processes. This can serve as support for the planning phase and risk assessment work.

Consistent with conducting the internal audit, the external auditor will address exposed nonconformities. Given the external auditor's focus on financial reporting, a different emphasis may be placed in the deviations to be examined. For example, repeated approvals of the same voucher will generate interest from an efficiency standpoint, but perhaps not from an audit standpoint. Despite a potentially different selection of deviations, the approach to clarify them is similar to what has already been described for the internal auditor. This will require a combination of a review of process executions against business rules, variant analysis and case analysis. Rule testing is well suited as a control test. Indeed, each control mechanism can be formulated as a rule: "if..., then...." For example, "if a receipt is created, then it is approved later." Variant and case analysis are used to answer more targeted questions and lean closely towards substantive controls.[6] Examples include reviewing transactions of a specific person, process executions in which manual activities have taken place and activities outside of working hours.

As with the internal audit, communication will take place, supported by the visual output of the analyses. An important aspect in this is that the findings are based on objective data and that they are easily transferable if the right graphics are used. Figure 8 summarizes the audit process supported by process mining.

If the auditor wishes to expand on the findings of the internal auditor, he or she must, as with any other audit, build in a number of checks regarding the quality and

[6] However, the split between control testing and substantive testing is no longer strictly applicable if the full data set is used to verify the operation of a control.

completeness of the information provided. In the context of process mining, there are the following specific points of interest:

- If the event log is provided, the auditor should check the underlying script and
  1. verify that no errors have been made
  2. check and take into account the underlying assumptions and filters used
- What type of systems were consulted to build the event log? Are these systems well managed in terms of access and control? Can information from these systems be relied upon?
- If discovered process models or anomalies are provided: what algorithm was used with what settings (which parameters are used)? Is there a script to replicate (and check) this analysis? Which normative model or set of rules was tested?

4.3 Practical Applications and Available Software Tools

While research in the field of process mining continues to increase, there is an increased adoption of process mining in companies. The Task Force on Process Mining (https://tf-pm.org) aims to promote process mining, publish scientific research on it, establish standards, and organize workshops. Furthermore, the Task Force keeps up-to-date information on developments within the process mining domain and publishes event logs, introductory and other videos, and case studies from the industry. For an up-to-date overview of practical process mining applications and software tools, we refer you to the website of the Task Force on Process Mining, where the aforementioned information is available under the tab "resources."[7]

5 Conclusion

Business processes are at the core of a well-functioning organization. They reflect how information should flow and what actions should be taken to achieve business goals. Because processes reflect the functioning of a company, including the implemented control mechanisms, they are a good starting point for gaining an insight into the environment in which financial reporting occurs.

The discipline of Business Process Management (BPM) focuses on managing and improving business processes within an organization. It is often partially adopted during an audit. Traditionally, BPM starts from an analysis technique that mainly relies on interviews and consulting existing process documentation. Based on the

[7] Although there is not an exhaustive list of tool vendors, many have their own introduction video on this site.

(derived) process models, insights are gained about how the company manages its processes and whether it is in control.

Although the traditional BPM approach can provide valuable insights into an organization's processes, it is limited to analyzing prescribed procedures in the form of normative process models. Normative process models do not describe the actual processes within an organization but rather propose an ideal image, a procedure that should be followed. As a result, the quality of the process analysis depends on the quality of the models and the extent to which the model matches reality. This is because the actual process executions often contain situations that are not included in the prescribed processes. Depending on how much these exceptions occur, there is a small or a large(r) mismatch between reality and the process models that form the basis of analyses.

To ensure that process analysis leads to correct insights, process mining can be applied. Process mining is a collective name for all data-driven process analysis techniques that start from an event log. It combines the strengths of the BPM approach with data analysis techniques to gain insights into the actual business processes. Process mining allows us to analyze the entirety of recorded activities to understand the business processes better. More specifically, process mining techniques provide insights into the ordering of activities, the timing of activities, and the actors involved in the actual process.

Every process mining analysis starts with the collection of data. Data from one or more sources are combined to build an event log. An event log contains data about one specific business process and is therefore used as the input for process mining analyses. A process mining analysis for one process usually includes the following six steps: (1) building an event log, (2) process discovery, (3) a conformance check against a process model, (3) a conformance check against a set of rules, (5) a variance analysis, and (6) a case analysis.

Process discovery and conformance checking are the two main types of process mining that are relevant to auditing. Based on an event log, a process discovery analysis can reveal the actual business process. This actual process is then usually visually presented in a process model. Conformance checking goes a step further by comparing the process behavior from the event log with a normative process. This normative process can either take the form of a procedural model or a set of business rules. Plotting the recorded activities against the norm leads to the identification of process deviations. Process deviations can either indicate exceptional cases, or errors and fraud. Filtering out the second group currently remains a challenge, as with all data analysis approaches in the context of an audit.

After performing a process discovery and conformance check, a variant or case analysis may be of interest. A variant analysis is an examination of the different ways in which the actual process was executed. A case analysis takes a close look at a specific subset of transactions by analyzing certain characteristics in depth. An example are the transactions performed on a particular day or by a particular department or person.

The insights generated from a process mining analysis provide a sound basis for improving business processes in terms of efficiency and risk. This broad view of

processes ensures that the insights generated are relevant to both the internal and external auditor. This chapter discussed how the various process mining analyses can support both auditors, as well as the concerns of the external auditor if he or she wishes to elaborate on the analyses of the internal auditor.

References

Chiu, & Jans. (2019). Process mining of event logs: A case study evaluating internal control effectiveness. *Accounting Horizons, 33*(3), 141–156. https://doi.org/10.2308/acch-52458

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. (2018). *Fundamentals of BPM* (2nd ed.). Springer.

Jans, M. (2019). Auditor choices during event log building for process mining. *Journal of Emerging Technologies in Accounting, 16*(2), 59–67.

Jans, M., & Hosseinpour, M. (2019). How active learning and process mining can act as Continuous Auditing catalyst. *International Journal of Accounting Information Systems, 32*, 44–58.

Jans, M., Alles, M., & Vasarhelyi, M. (2013). The case for process mining in auditing: Sources of value added and areas of application. *International Journal of Accounting Information Systems, 14*(1), 1–20.

Jans, M., Alles, M. G., & Vasarhelyi, M. A. (2014). A field study on the use of process mining of event logs as an analytical procedure in auditing. *The Accounting Review, 89*(5), 1751–1773.

Jans, M., Soffer, P., & Jouck, T. (2019). Building a valuable event log for process mining: An experimental exploration of a guided process. *Enterprise Information Systems, 13*(5), 601–630.

Pesic, M., Schonenberg, H., & Van der Aalst, W. M. (2007, October). Declare: Full support for loosely-structured processes. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)* (p. 287). IEEE.

Van Der Aalst, W. M. (2016). *Process mining: Data science in action*. Springer.

Van der Aalst, W. M., Pesic, M., & Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science-Research and Development, 23*(2), 99–113.