Blockchain Framework and Guidance

ISACA®

About ISACA

For more than 50 years, ISACA® (www.isaca.org) has advanced the best talent, expertise and learning in technology. ISACA equips individuals with knowledge, credentials, education and community to progress their careers and transform their organizations, and enables enterprises to train and build quality teams. ISACA is a global professional association and learning organization that leverages the expertise of its 145,000 members who work in information security, governance, assurance, risk and privacy to drive innovation through technology. It has a presence in 188 countries, including more than 220 chapters worldwide.

Disclaimer

ISACA has designed and created *Blockchain Framework and Guidance* (the "Work") primarily as an educational resource for professionals. ISACA makes no claim that use of any of the Work will assure a successful outcome. The Work should not be considered inclusive of all proper information, procedures and tests or exclusive of other information, procedures and tests that are reasonably directed to obtaining the same results. In determining the propriety of any specific information, procedure or test, professionals should apply their own professional judgment to the specific circumstances presented by the particular systems or information technology environment.

Reservation of Rights

ISACA

1700 E. Golf Road, Suite 400
Schaumburg, IL 60173, USA
Phone: +1.847.660.5505
Fax: +1.847.253.1755
Contact us: https://support.isaca.org
Website: www.isaca.org

Provide Feedback: To comment, please visit https://support.isaca.org.
Participate in the ISACA Online Forums: https://engage.isaca.org/onlineforums

Twitter: http://twitter.com/ISACANews
LinkedIn: www.linkedin.com/company/isaca
Facebook: www.facebook.com/ISACAGlobal
Instagram: www.instagram.com/isacanews/

Blockchain Framework and Guidance
ISBN 978-1-60420-860-3

Acknowledgments

Blockchain Framework and Guidance is the result of the collective efforts of many volunteers. Special thanks go to:

Lead Author
Ron Quaranta, EC–Information Systems and Technology Project Management, Wall Street Blockchain Alliance, USA

Contributing Authors
Gerard (Rod) Brennan, Ph.D., CFE, Lukka, USA
Sonia Goklani, CEO, CLEARTRACK, USA
Will Janensch, USA
Tuan Phan, CBSP, CISSP, Security+, PMP, Caplock Security, USA
Marc T. Staut, CITO, Boomer Consulting, USA

Expert Reviewers
Marianne Azer, Ph.D., Member of the Egyptian Parliament, Egypt
Vladlena Benson, Ph.D., CDPSE, United Kingdom
Earnest Collins, CISA, CISM, CFE, CISSP, AES, USA
Victor Fang, Ph.D., AnChain.ai, USA
Ramses Gallego, CISM, CGEIT, CCSK, CISSP, Six Sigma Black Belt, Micro Focus, Spain
Niel Harper, CISA, CRISC, CDPSE, CISSP, Denmark
Luis Jugo, CISA, CCSA, CIA, CRMA, PMP, USA
Neil Lappage, CISM, CISSP, Australia
Wickey Wang (Jiewen Wang), CISA, USA

Board of Directors
Tracey Dedrick, Chair, Former Chief Risk Officer, Hudson City Bancorp, USA
Rolf von Roessing, Vice-Chair, CISA, CISM, CGEIT, CDPSE, CISSP, FBCI, Partner, FORFA Consulting AG, Switzerland
Gabriela Hernandez-Cardoso, Independent Board Member, Mexico
Pam Nigro, CISA, CRISC, CGEIT, CRMA, Vice President–Information Technology, Security Officer, Home Access Health, USA
Maureen O'Connell, Board Chair, Acacia Research (NASDAQ), Former Chief Financial Officer and Chief Administration Officer, Scholastic, Inc., USA
David Samuelson, Chief Executive Officer, ISACA, USA
Gerrard Schmid, President and Chief Executive Officer, Diebold Nixdorf, USA
Gregory Touhill, CISM, CISSP, President, AppGate Federal Group, USA
Asaf Weisberg, CISA, CRISC, CISM, CGEIT, Chief Executive Officer, introSight Ltd., Israel
Anna Yip, Chief Executive Officer, SmarTone Telecommunications Limited, Hong Kong
Brennan P. Baybeck, CISA, CRISC, CISM, CISSP, ISACA Board Chair, 2019-2020, Vice President and Chief Information Security Officer for Customer Services, Oracle Corporation, USA
Rob Clyde, CISM, ISACA Board Chair, 2018-2019, Independent Director, Titus, and Executive Chair, White Cloud Security, USA
Chris K. Dimitriadis, Ph.D., CISA, CRISC, CISM, ISACA Board Chair, 2015-2017, Group Chief Executive Officer, INTRALOT, Greece

Page intentionally left blank

TABLE OF CONTENTS

Page intentionally left blank

LIST OF FIGURES

Chapter 4. Generic Blockchain Reference Architecture

Chapter 6. Governance Model and Management Guidelines

Chapter 7. Security and Privacy Considerations

Page intentionally left blank

Introduction

Blockchain Framework and Guidance provides an overview of blockchain, including history, types, benefits, features, concepts and use cases, and offers a framework for the adoption of blockchain technology across enterprises. The ISACA blockchain framework provides foundational information, practical guidance and proposed tools for proper blockchain implementation, governance, security, audit and assurance. The unique aspects of blockchain technology and the blockchain touchpoints with existing technology ecosystems are explained in detail. In addition, *Blockchain Framework and Guidance* maps existing technology implementation disciplines into the process of blockchain adoption.

Note: This framework frequently references blockchain technology terms that may be new to readers; thus, readers are encouraged to reference the glossary in appendix B for definitions.

Page intentionally left blank

Chapter 1
Blockchain Technology Overview

1.1 Introduction

Chapter 1	Blockchain Technology Overview
Description	This chapter provides a brief overview, definitions of blockchain technology and common types, and some important benefits of using blockchain technology.
Key questions answered	• What is a blockchain and what are the origins of blockchain technology? • What are the types of blockchain? • What are the benefits of blockchain? • What is consensus methodology and why is it relevant? • What are some current blockchain use cases?
Stakeholders	Stakeholders include, but are not limited to, board of directors, executive management, business unit managers, IT managers/practitioners, assurance providers, risk management personnel, regulators, and business or vendor partners.
Resources	• https://bitcoin.org • https://bitcoin.org/bitcoin.pdf • https://www.nist.gov/topics/blockchain • https://blockchain.ieee.org/about • https://csrc.nist.gov/Groups/Computer-Security-Division/Cryptographic-Technology

1.2 Blockchain Basics

1.2.1 Overview

A blockchain is a shared transactions ledger that can be accessed by and among multiple parties, using cryptography and peer-to-peer technology to secure data into blocks and store them in an immutable chain of transactions, without any trusted central authority. Blockchain can be considered a subset of *distributed ledger technology* or "DLT," which broadly speaking is a type of database technology which can be shared and made accessible to multiple parties. While there are many types of DLT technology, not every DLT is a blockchain. That said, every blockchain is a form of DLT. By design, each participant in the blockchain possesses the same copy of this ledger and can write to it, assuming that they are properly permissioned to do so. The blockchain fundamental function is to accomplish this sharing and distribution of data or value, without the need for a trusted intermediary and without any enforced system management. It is the design of the software and network itself that allows participants to trust the accuracy and veracity of the information on the blockchain.

1.2.2 History

In 2008, an anonymous technologist and author (or authors) under the pseudonym of Satoshi Nakamoto published "Bitcoin: A Peer-to-Peer Electronic Cash System"[1] on a technology mailing list and, in January 2009, the first version of the open-source Bitcoin cryptocurrency system was made publicly available. It is worth noting, though not

[1] Nakamoto, S.; "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf

germane to this Framework material, that as of this writing the person or persons known as "Satoshi Nakamoto" remain unknown and anonymous (though several have come forward claiming to be Nakamoto, none have been able to provide substantial evidence to prove this claim). The Bitcoin blockchain launch initiated the broad use of a cryptographic, peer-to-peer system for the secure and anonymous exchange of data and value between parties. Its launch created a more than decade-long discussion about the nature of trust, money and identity, and a wave of innovative startups looking to reinvent whole economic models based on blockchain technology.

1.3 Types of Blockchain

1.3.1 Public

The foremost examples of public blockchains to date are the Bitcoin cryptocurrency blockchain[2] and the Ethereum cryptocurrency and programmable smart contract blockchain,[3] although there are numerous others. A public blockchain is permissionless by design, i.e., any person or party can join the blockchain network; participate in the consensus process; produce applications, smart contracts or transactions; and have a transparent view of the history of all transactions on that blockchain. Participation simply requires the download of the blockchain platform open-source software. Because the blockchain system is a decentralized network, no single entity can control the network. There are relevant concerns about the concentration of network processing power, such as in a 51% attack, which is explained later in this framework. Public blockchains operate using a combination of cryptographic verification, e.g., proof of work, which is the consensus methodology used by Bitcoin, and are secured by economic-based incentives to maintain and manage the blockchain ecosystem.

1.3.2 Private

Private blockchains are permissioned. They are designed and operate based on access controls that can restrict which, and what type of, parties can participate, and their functions and capabilities in the blockchain workflow. In private blockchains, one or more stakeholders, or administrators, control the network, introducing the requirement for third-party acceptance to perform transactions on the blockchain, which does not exist in public blockchain implementations. In a private blockchain, only those parties participating in a transaction have knowledge about it. For the most part, other parties cannot have access, although many private blockchains allow for layers of different types of permissions (e.g., submit data, confirm data and view-only). Therefore, although private blockchains can and usually do have a consensus methodology associated with them, trust still plays an important role in private implementations.

1.3.3 Hybrid

Growing use cases for blockchain technology led to the evolution of hybrid blockchains. These are blockchain implementations in which one component is a fully public blockchain (with all the benefits and challenges), and the parallel component is a permissioned private blockchain, allowing for enterprise-level transactions that maintain regulatory compliance and access permissioning. Hybrid blockchains seek to realize the best benefits of public and private blockchain technology, including not being fully open to any single participant, while maintaining data immutability and transparency. In a hybrid blockchain, although transactions are not public, they are always available to be verified when necessary.

1.4 Core Benefits of Blockchain Technology

Following are the core benefits of using blockchain technology:

- **Anonymous**—Also known as pseudonymous, in which private information (for a public blockchain) associated with transactions are linked to wallet addresses and public keys, and no personally identifiable information is viewable.

[2] Bitcoin, "Get started with Bitcoin," https://bitcoin.org/en/
[3] Ethereum, https://ethereum.org/en/

- **Distributed**—More reliable; system is more resilient. Component failure is minimized, and transactions are encrypted and stored on multiple nodes globally.
- **Decentralized**—No need to trust a central authority, which results in less likelihood of a single point of failure and allows the blockchain to be censure resistant.
- **Immutable**—Data is append-only and cannot be modified; for the most part, public blockchain transactions are tamperproof.
- **Transparent**—Transparency allows for transaction history to be more easily audited and offers greater accuracy and consistency.

1.5 Features of Blockchain

1.5.1 Cryptographic Verification

Public/private key encryption is used to manage and verify ownership and transactions on the blockchain.

- Public/private key encryption uses a public key and a private key to accomplish different tasks, such as sending or receiving bitcoin. Public keys can be widely shared; private keys must be kept secret. However, it is critical to note that a private key can never be reset or recovered if lost or stolen.
- Using a person's public key, it is possible to encrypt a message so that only the person with the related private key (the receiver of the message) can decrypt and read it. A digital signature is created using a sender's private key so that the recipient of a digitally signed message can verify, by using the sender's public key, that the message was created by the owner of the private key, i.e., the sender.
- Each bitcoin is associated with its current owner's public key. When someone sends bitcoin, a transaction is created connecting the new owner's public key to the specific amount of bitcoin, and it is signed (not shared) with the sender's private key. When this transaction is broadcast to the Bitcoin network, the network is aware that the new owner of these coins is the owner of the new public key, i.e., the recipient.

Most cryptocurrencies use a wallet to store the public/private key pairings that allow a user to track ownership, and send or receive cryptocurrencies. A wallet can be a hardware device, program or service. Cryptocurrency itself does not reside in any wallet, because cryptocurrencies do not exist in a physical form. Instead, cryptocurrencies are maintained on the blockchain, which consists of transaction records that detail the private and the public keys that have control over these cryptocurrencies.

1.5.2 Consensus Methodology

The foundations of the consensus methodology and its applicability to blockchain have given rise to numerous consensus methods, i.e., algorithms. These algorithms define how a blockchain operates, how transactions and data are persisted into the blockchain, and how the rules for governance on the blockchain network are enforced.

The problem of consensus appears when attempting to control and keep multiple actors synchronized in a network, whereby resilience and fault tolerance are a requirement for the system or network to operate. Blockchain (and all distributed ledger technology and other technologies) seek to be Byzantine Fault Tolerant (BFT). BFT is the property of a system such that the system can withstand any failures and can continue to function even if some nodes fail or act maliciously. To better understand consensus methodologies, it is important to look at some core blockchain-technology concepts:

- Within a blockchain, blocks are built to capture the transactions that are accepted by the network, and these blocks are appended to the blockchain.

- Within a blockchain system a variety of participants engage in the actions associated with creating blocks and may have different roles in the various consensus methodologies. These participants include miners, block creators, block publishers and validators.

Note: The previous core concepts are explained in greater detail in section 4.4.

There is a wide range of consensus methodologies. The following list highlights the major methodologies:

- **Proof of work (PoW)**—Conducted by miners (participants who keep the blockchain running by providing computing resources) who are competing to solve a cryptographic problem (hash puzzle). The PoW algorithm is used to confirm transactions and produce new blocks that are added to the chain.
- **Proof of stake (PoS)**—Type of consensus algorithm by which a cryptocurrency blockchain network aims to achieve distributed consensus. In PoS consensus, the creator of the next block of data is chosen via several combinations of random selection and wealth or age (i.e., the stake) within the blockchain.
- **Practical Byzantine Fault Tolerance (pBFT)**—Consensus mechanism in which all nodes are ordered in sequence with one node being the primary node or leader, and all others referred to as backup nodes. Nodes communicate to prove (agree by majority rule) that messages came from a specific peer node and that a message was not modified during transmission.
- **Proof of Elapsed Time (PoET)**—Consensus mechanism algorithm that is often used on the permissioned blockchain networks to randomly determine the next block publisher using the Intel Software Guard Extensions (SGX). SGX allows applications to run trusted code in a protected environment to ensure that the selection is fair.

Note: See the glossary for definitions of the previous terms.

1.6 Use Cases

This section provides two use cases that leverage blockchain technology. These cases are mostly production-level (i.e., they have moved beyond the proof-of-concept stage and engage actual internal and external stakeholders) and are relevant across multiple industries:

- **Supply chain**—The global supply chain analytics industry is expected to reach US $9.88 billion by 2025[4] and involves thousands of forms and types of participants in diverse industries, such as retail and consumer, manufacturing, and transportation. The supply chain industry is a notable early adopter of blockchain technology because it provides a much higher level of transparency and agility to the supply chain than previous systems. For example, Bumble Bee Foods united several stakeholders in the global fishing industry and launched a blockchain-based track-and-trace platform for fish, from the ocean to the table. A quick response (QR) code provided by the company allows for the tracking of the origins of the fish, but also gives sellers and retailers the ability to obtain real-time confirmed-correct information about the flow of the physical goods, their status, progress and more.[5]
- **Real estate**—The global real estate investment industry, accounting for a market size of over US $9.6 trillion as of 2019,[6] was also keen to understand the possibilities and potential of using blockchain. For example, land

[4] Grand View Research, "Supply Chain Analytics Market Size, Share, & Trends Analysis Report By Solution, By Service, By Deployment, By Enterprise Size, By End-use, By Region, And Segment Forecasts, 2019 – 2025," GVR-1-68038-928-9, August 2019, www.grandviewresearch.com/industry-analysis/the-global-supply-chain-analytics-market

[5] Kotecha, N.; S. Muma; "The Critical Role for Blockchain in the Post-COVID-19 Supply Chain," Modern Materials Handling, 9 June 2020, www.mmh.com/article/the_critical_role_for_blockchain_in_the_post_covid_19_supply_chain

[6] MSCI, "Real Estate Market Size 2019," June 2019, www.msci.com/documents/10199/035f2439-e28e-09c8-2a78-4c096e92e622

titles, which represent the rights to a parcel of property in which the holder has a legal and perhaps equity interest, continue to be paper documents. Using paper for such important documents has inherent risk, including theft, mismanagement, loss and fraud. Enterprises, such as Codefi,[7] are partnering with major real estate and title firms across the globe to remove the complexity of legacy systems, by using blockchain technology to offer an immutable and secure digital registry of title ownership. This registry includes authentication of documents and transaction transparency, resulting in less loss, fraud and legal proceedings, and cost efficiencies to title companies and property owners.

[7] CONSENSYS Codefi, "Blockchain for Global Commerce and Finance," https://codefi.consensys.net

Page intentionally left blank

Chapter 2
Stakeholders and Stakeholder Management

2.1 Introduction

Chapter 2	Stakeholders and Stakeholder Management
Description	Stakeholders are any members of an enterprise or those interacting with the enterprise, who can or would be impacted or affected by the implementation of blockchain-based technology within the enterprise. Stakeholders can be internal or external to the enterprise. In this chapter, private blockchain implementations are the focus.
Key questions enterprises need to ask	• Have we identified all relevant stakeholders for a blockchain implementation? • Do we have a communications management plan to interact with appropriate stakeholders? • Do we have the technical expertise to support a blockchain implementation? • Do we have management buy-in for implementation? • Does the enterprise business model support blockchain, or does blockchain implementation represent a change in business model? • Do we have budget for a blockchain implementation? • Is there a requisite ROI to justify a blockchain implementation?
Stakeholders	Stakeholders include, but are not limited to, board of directors, executive management, business unit managers, IT managers/practitioners, security personnel, assurance providers, risk management personnel, regulators, business or vendor partners.

The importance of stakeholder definition and management buy-in cannot be understated in the context of a blockchain implementation. It is important to recognize that blockchain enterprise implementation is not solely a technical discussion. Given the fundamentally different approach that blockchain entails, the implementation of blockchain requires discussion across multiple parties within the enterprise, including technical, product, project, business management, compliance and risk stakeholders.

2.2 Blockchain Stakeholder Management

For a blockchain implementation, each stakeholder's interests and needs must be properly and fully represented, and proper stakeholder management is critical to implementation success. Like any successful information technology project, blockchain implementation relies on aligning and consolidating the correct technology, processes and people to an agreed strategy. Doing this correctly will allow for the full benefits of blockchain technology implementation at the enterprise level. Alternatively, incorrect or poorly executed stakeholder management can result in technology problems, including misalignment, scope creep, over-budget implementations and, in the extreme, a technology solution that is not fit for purpose.

To ensure the full benefits of blockchain technology implementation, the ISACA blockchain framework proposes a series of guiding steps to allow for proper stakeholder management:

1. **List the enterprise stakeholders**—Start with the appropriate enterprise function, followed by the correct contacts within each functional group. As noted above, these stakeholders can be, but are not limited to, board of

directors, executive management, business unit managers, IT managers/practitioners, security personnel, assurance providers, risk management personnel, regulators, business or vendor partners.

2. **Group stakeholders into buckets**—Consider those working on, those having authority over and those impacted (positively or negatively) by enterprise blockchain adoption. Note that individuals may be part of one or more (or all) buckets.
3. **Create and launch a communications management plan**—This does not necessarily need to be a new, formalized structure. Many enterprises have shared portals and communications processes already in place. Features that should be determined include the following:
   - The information that is to be shared with the relevant stakeholders
   - Timing and frequency of communications
   - Format of communications (e.g., in-person, email, video, telephone and collaboration platform)
   - Approval process for blockchain integration within the enterprise
   - Documentation and formal report creation, etc.

   Although not a comprehensive list, ISACA suggests that, for enterprise blockchain adoption, stakeholders should leverage existing communications protocols as closely as is practical. In addition, the communications management plan should be tightly coupled with any project management plan created for the project, or with the internal project management office of the enterprise.
4. **Provide regular status updates**—These should be tailored to the groups of stakeholders and shared on a regular (weekly, biweekly or monthly) basis as part of the larger stakeholder communications plan. All updates should be archived for post-project review and lessons learned.
5. **Develop appropriate buy-in processes**—The usual approval process may suffice for enterprise implementation; however, given the potential of blockchain to impact actual ongoing processes within the enterprise, implementors should be prepared to have buy-in for implementation across multiple stages of the blockchain project.
6. **Finalize a blockchain implementation report for all stakeholders**—Blockchain integration within an enterprise does not end when the technical aspects of the project are completed. Teams should be prepared to present a summary report of the project to all stakeholders, and show actual versus projected improvements in process flows and return on investment for the project. Team members should also put into place ongoing status updates for stakeholders over time.

After the above steps are initiated and as the project proceeds, the monitoring of stakeholder engagement is an ongoing process even beyond project completion. When a project is underway, personnel are encouraged to create and maintain a lessons-learned document, to capture important problems or challenges, and how they were mitigated; an issues log for problem tracking; and a risk register for risk identification, tracking, responsibility assignments and mitigation. Again, these need not be overly formal (although they may be) but should align with any project management office functions and features currently in place at the enterprise.

Ultimately, stakeholder management in the context of a blockchain implementation mirrors similar stakeholder processes for other IT projects. It is important to keep in mind, however, that blockchain technology may require fundamental process changes. Thus, ongoing stakeholder engagement is even more critical to ensure a successful blockchain implementation at a technical and organizational level.

Chapter 3
Implementation Considerations

3.1 Introduction

Chapter 3	Implementation Considerations
Description	Implementation considerations for a blockchain system covers the standard series of issues associated with new IT implementations, along with some unique issues. In addition to the preliminary challenges, including definition of how the system is to be built/integrated, ensuring operational functions and design, and ensuring that the system meets organization quality standards, blockchain implementation further requires the enterprise to answer additional questions. This section addresses these additional concerns and offers potential suggestions to approach them.
Key questions enterprises need to ask	Preliminary implementation questions include: • What is the scope of the implementation? • Does a blockchain integration further the strategic objectives of the organization? • Will new skill sets or personnel be required? • Who will define and control interactions in the blockchain implementation? • Is there a plan for IT interaction across the enterprise and with external partners, vendors, etc.?
Stakeholders	Stakeholders include, but are not limited to, executive management, business unit managers, IT managers/practitioners, security personnel, risk management personnel, business or vendor partners.

Implementation considerations are one of the keys to the success of any technology project. Blockchain implementations, however, also present a series of additional considerations. This framework section addresses the standard issues related to any technology implementation and the blockchain considerations within each relevant item, by proposing a series of implementation steps.

3.2 Step 1: Determine a Problem/Set a Use Case

As part of implementation, it is critical to determine the specific problem(s) that the enterprise is trying to correct or remedy, and determine if a blockchain implementation is actually the correct approach. This requires fully understanding the current processes being reviewed and the appropriate parties/systems/business units involved. It is important that a blockchain is designed carefully, not just to solve organizational challenges, but to ensure that it fits with existing enterprise processes and anticipates the evolution of those processes. If not, those processes may need to be revised to lay the groundwork for blockchain. Considerations include the following:

1. **Define the specific enterprise problem**—This should include talking to internal and external clients, different companies or external organizations involved, and other stakeholders for whom an enterprise process problem exists (e.g., supply chain is inefficient, data is often wrong and provenance cannot be fully accounted for). In the absence of a clearly defined problem that needs correction, any blockchain testing or prototyping should be relegated to potential innovation groups or personnel whose work is done in parallel to ongoing operations.
2. **Can the stated problem be solved with existing technology solutions?**—Despite the growing use of blockchain across multiple industries, the mere existence of the technology does not necessarily mean that it has an immediate place in the enterprise. Careful consideration of existing in-house or external systems may render possible solutions that preclude the need for a relatively new technology like blockchain. In addition, there are clear examples of a blockchain solution would not be optimal. For example, workflows involving few if any

intermediaries and processes that require high-performance throughput do not lend themselves to the use of current blockchain technology.

3. **Who are the specific stakeholders involved?**—After identifying an existing enterprise problem that may lend itself to a blockchain solution, it then becomes important to specifically identify the network of stakeholders involved in this process. This network of operators may be internal or external to the enterprise, and determining who they are and where they sit in the process is critical to successfully proposing a blockchain use case within the enterprise.
4. **Will stakeholders support the use of blockchain technology?**—Determining the problem and setting the use case forces the enterprise to address the important topic of stakeholder acceptance of a proposed blockchain solution. It is important that executive management and business unit managers support not just the elaboration of the problem but also that blockchain is seen as an optimal solution to the problem and an ideal use case. This framework recommends that such acceptance be documented and approved internally, or else a blockchain project might never get beyond the experimentation stage or the fear-of-missing-out (FOMO) perception. It is also important to keep in mind that any implementation is a collective process between system champions, IT leaders, implementation teams and the entire organization. Engagement and communications should be sustained with all throughout the implementation process.

If findings from the previous considerations show that there are multiple disparate stakeholders, among whom data must be protected, and if there are numerous duplicative processes or friction points, then a blockchain-based solution might be called for.

3.3 Step 2: Initiation Stage

After a possible use case for blockchain within the enterprise is determined, and internal support of the project is achieved, the initiation stage begins. This stage requires the enterprise to examine its existing technology systems, and determine the specifics of the problems that may exist and the opportunities that blockchain technology may provide for the improvement of these processes. Using the global supply chain industry as an example, the act of shipping goods from one destination to another requires multiple internal and external stakeholders (producer, shipper, logistics, insurance, etc.), many of whom have disparate systems or, worse, basic spreadsheets as a means for tracking and sharing data. It is important, in the initiation stage, to ask the following questions:

1. What systems and processes are part of the technology that the enterprise is seeking to replace or upgrade? (See the Business Architecture Level section, in chapter 6, for further context). Specific questions are:
   - What databases or data centers are involved, and what type of data are used? This question aids in determining the appropriate blockchain solution, and the required capacity and throughput available in the blockchain platform.
   - How are the data used and who owns the data? Data ownership/stewardship and an understanding of the data change and manipulation processes aids in determining not only the blockchain solution, but also how the Governance Model (discussed in chapter 6) needs to be structured. Using a blockchain is especially ideal for data processing and transactions that are dependent on prior transactions or changes in the data.
   - How many stakeholders add to/review data in the current database and processes? If there are many internal and external stakeholders, then the blockchain platform must accommodate how each is meant to function in the workflow process. Smart contracts (discussed later in this framework) may play an integral function here. A smart contract is a computer program that prescribes the conditions and outcomes, and is stored and processed on second- and later-generation blockchain networks.
   - What is the trust profile of each stakeholder? For the use case under consideration, defining how the network of stakeholders arrive at consensus is important and relates directly to the blockchain consensus models. However, it is important in the initiation stage to define how data/transactions flow from an end-to-end process in the old system or business process, how the new consensus can be used to achieve the strategic objective in the newly implemented blockchain (e.g., The data sent by department A is accurate and

usable by department B.), and how such consensus will be replicated in the new system. In addition, with a potential reduction in the number of intermediaries, define how this consensus will evolve in a blockchain-based system. Arriving at a consensus, even with known parties, and establishing the source of data or assets being transacted are key to understanding how the blockchain solution will be structured, and what the levels of trust among stakeholders should be. This information also shows clearly if there are specific concerns about sharing all or parts of the data in existing databases.

- Are privacy laws or regulations associated with the data? If the data and transactions require security to adhere to existing laws or regulations and protect privacy and identity, it is advisable to make certain that the appropriate enterprise legal and regulatory experts are made stakeholders in the blockchain project (or outside legal and regulatory experts are engaged if needed), and that they are kept up to date regularly on project developments, consensus and governance rules, data usage, and more. Concurrently, given the unique aspects of blockchain technology, end user privacy and other such agreements may need to be updated to meet project requirements.
- Will new or existing intermediaries add complexity? An important aspect of initiating a blockchain project is to understand not only existing intermediary stakeholders for the use case, but also the intermediaries who may need to be added in the future (e.g., new vendors and customers). These intermediaries should be considered when initiating a blockchain project. The blockchain needs to be flexible enough to allow for scale and growth, while maintaining the governance and controls required for a blockchain ecosystem. (Governance and controls are discussed in more depth in chapter 6.)
- Will the use of blockchain technology allow for portions of the existing process or workflow to be removed? Considering the information gained in the previous initiation stage questions, if it is determined that trusted parties will be involved either within the enterprise or externally, this is an opportunity to rationalize reducing the number of steps in the process.
- What is the time sensitivity of data, if any? Keeping in mind the current, relatively low throughput state of enterprise blockchain technology, it is important to determine the time sensitivity of the enterprise process in question. Although future blockchain developments may allow for more real-time throughput, enterprise processes requiring very rapid data or asset changes, or manipulation may be difficult with a blockchain solution. It may be determined that a distributed ledger can reduce the time it takes for data updates to reach counterparties or other stakeholders, particularly if existing processes are manual, or data are slow or inconsistently replicated to secondary systems or across stakeholders.
- Are the data sets involved normalized across all parties? Are there data gaps across the existing system processes? The existence of nonnormalized data or significant data gaps within processes make it challenging to initiate a blockchain process. Care should be taken to minimize these issues to allow for a reasonably effective blockchain implementation.

Note: Readers are encouraged to keep abreast of blockchain developments and updates to this framework for developments in blockchain throughput.

2. What specific criteria are being used to choose a particular blockchain technology platform? ISACA recommends using the following list, though not exhaustive, to help narrow the field of potential blockchain solutions:
   - Conduct a gap analysis of the current technology and process structure, and/or any new business flows. Consider using a vendor framework analysis as part of this exercise, which may suggest possible responses to following steps.
   - When setting a use case, was a determination been made about whether to use a public, private or hybrid blockchain? The previously noted issues about types and number of stakeholders, the trust level associated with them, etc. can help to inform the appropriate decision.
     - How many transactions per second/minute will your system require now and in the future? Having answered the time sensitivity of data previously, the total throughput needed to perform in a satisfactory manner now

and in the future is critical. When scoping a solution, ensure that the technology can demonstrably show the ability to manage estimated current and future throughput to the project specifications.

- What is the confidence level of the technology team? Whether using internal or external personnel, it is critical to conduct proper due diligence on the team. How much experience do they have with enterprise blockchain implementations? How many systems have they successfully implemented? What is their level of knowledge of ongoing blockchain developments? How knowledgeable are they about hard or soft forks, for example (if relevant to the specific project), and can they explain these challenges and any remediations to other stakeholders? Can they implement within your existing project timetable? These and other questions are especially important when vetting a blockchain implementation, particularly with external technology teams. A poorly implemented blockchain project can be very costly.
- Will new training be required for a blockchain implementation? Training is key to success, during and after implementation of a blockchain solution. If needed, be sure that an adequate training policy is in place for all stakeholders. This is particularly important if an external blockchain team is being used, because their knowledge needs to be shared at the appropriate levels across the stakeholder groups. Appropriate time and resources should be allocated for this training and knowledge sharing. Define and communicate clear deliverables and proposed solution. Like any major technology implementation, it is important to regularly deliver to all stakeholders not only clear plans of deliverables, but also status updates and key metrics of success (e.g., meeting project deadlines, onboarding stakeholders, beta test dates and success), to convey project progress. If an internal project management team is not available, be sure to store and share a simple project planning document and update it regularly.

3. Critical to any blockchain implementation discussion is understanding the state of existing blockchain platforms and solution providers to determine criteria for choosing a specific blockchain technology. Some of the more prominent blockchain platform providers and their blockchain-solution classification type follow:

- **Amazon—**From its website:[8]

  Amazon's Managed Blockchain Service is a fully managed service that makes it easy to create and manage scalable blockchain networks using the popular open source frameworks Hyperledger Fabric and Ethereum. The service makes it possible to build applications where multiple parties can execute transactions without the need for a trusted, central authority. It is a fully managed service that allows participants to set up and manage a scalable blockchain network easily, while eliminating the overhead required to create the network, and automatically scales to meet the demands of thousands of applications running millions of transactions. Once the network is up and running, Managed Blockchain makes it easy to manage and maintain the blockchain network. It manages certificates and lets participants easily invite new members to join the network."

- **Digital Asset Modeling Language**—From its website:[9]

  Digital Asset Modeling Language ("DAML") is a computer language designed for modeling agreements between financial institutions and their customers. It was developed by a technology firm called Elevence and was published by Digital Asset after it acquired Elevence. DAML was designed specifically for use in multi-party smart contracts.

- **Ethereum**—From its website:[10]

  Ethereum is based on Bitcoin's protocol and its Blockchain design but is tweaked so that applications beyond money systems can be supported. The two Blockchains' only similarity is that they store entire transaction histories of their respective networks, but Ethereum's Blockchain does a lot more than that. Besides the history of transactions, every node on Ethereum network also needs to download the most recent state, or the current information, of each smart contract within the network, every user's balance and all the smart contract code and where it's stored. Essentially, the Ethereum Blockchain can be described as a transaction-based state machine.

[8] AWS, "Amazon Managed Blockchain," https://aws.amazon.com/managed-blockchain/
[9] MarketsWiki, "Digital Asset Modeling Language," http://www.marketswiki.com/wiki/Digital_Asset_Modeling_Language_(DAML)
[10] Cointelegraph, "What is Ethereum. Guide for Beginners," https://cointelegraph.com/ethereum-for-beginners/what-is-ethereum

- **Hyperledger**—From the website:[11]

  Hyperledger is an open source community focused on developing a suite of stable frameworks, tools and libraries for enterprise-grade blockchain deployments. It is a global collaboration, hosted by The Linux Foundation, and includes companies in multiple industries including finance, banking, supply chain, manufacturing and more. Built under technical governance and open collaboration, users are invited to participate in the development and promotion of the platform. Hyperledger has a modular approach to hosting projects and builds technical communities to develop blockchain and shared ledger POCs, use cases, field trails and deployments.

- **Microsoft® Azure®**[12]—The Microsoft Azure Blockchain Platform allows users to deploy fully managed blockchain networks simply and govern at scale with built-in governance and codeless consortia management. Modular controls provide easy member onboarding, codeless permissioning and simplified policy enforcement. It also lets users build blockchain applications with confidence on an open, flexible platform that integrates with the developer tools, data sources and applications that they already use.
- **R3**[13]—R3 is an enterprise blockchain software firm working with a broad ecosystem of dozens of participants across multiple industries, from private and public sectors, to develop blockchain applications on Corda, an open-source blockchain platform, and Corda Enterprise, a commercial version of Corda for enterprise use.

Many of the capabilities of the providers in the previous list are designed to work with, or be embedded in, each other. For example, Hyperledger code can be used within Amazon Web Services. Enterprises are encouraged to continue to research existing solution providers, including the multiple blockchain-as-a-service and platform providers.

A variety of blockchain platforms and solutions are available for numerous types of use cases. Some platforms and solutions are designed for enterprises, such as Hyperledger Fabric to build and deploy as permissioned private blockchains; others, such Bitcoin and Litecoin blockchain, are designed for payments using specific cryptocurrency. When analyzing these and other vendor providers, it is important to consider the overall taxonomy of the major blockchain platforms, their solutions and industry uses, their available protocols and tool sets.

Finally, after determining the use case, completing the initiation stage and initiating a blockchain implementation plan, it is worthwhile to establish a continuous review of some fundamental blockchain properties that will continue to impact operations well past implementation. For example:

- Will the immutable aspect of data on a blockchain be a challenge from a legal and regulatory perspective and/or from a product evolution and enhancement perspective? Given the potential challenges of the entered data being unchangeable and incapable of removal, it is important to have ongoing perspectives of key legal and regulatory developments around the world and governance and security issues related to blockchain use. In-house legal and regulatory experts, where applicable, not only should be updated on blockchain system changes but also should convey relevant legal and regulatory changes to the business and product teams.
- How will evolutions in blockchain technology affect the implementation? There are important considerations to note as blockchain technology capabilities grow and change over time. For example, the use of public or hybrid blockchain in an implementation may be impacted by blockchain ecosystem changes (e.g., a soft or hard fork). In addition, changes on the enterprise side may impact how blockchain is used in the future. It is important for implementers of blockchain technology to maintain a holistic and forward-looking perspective because the growth of blockchain, like any new technology, will have potential significant impacts on existing and future enterprise implementations.

[11] The Linux Foundation Project, "About Hyperledger," https://www.hyperledger.org/about
[12] Microsoft Azure, "Azure Blockchain Service," https://azure.microsoft.com/en-us/services/blockchain-service/#product-overview
[13] R3, "About r3: Industry-wide Collaboration is in our DNA," https://www.r3.com/about/

Page intentionally left blank

Chapter 4
Generic Blockchain Reference Architecture

4.1 Introduction

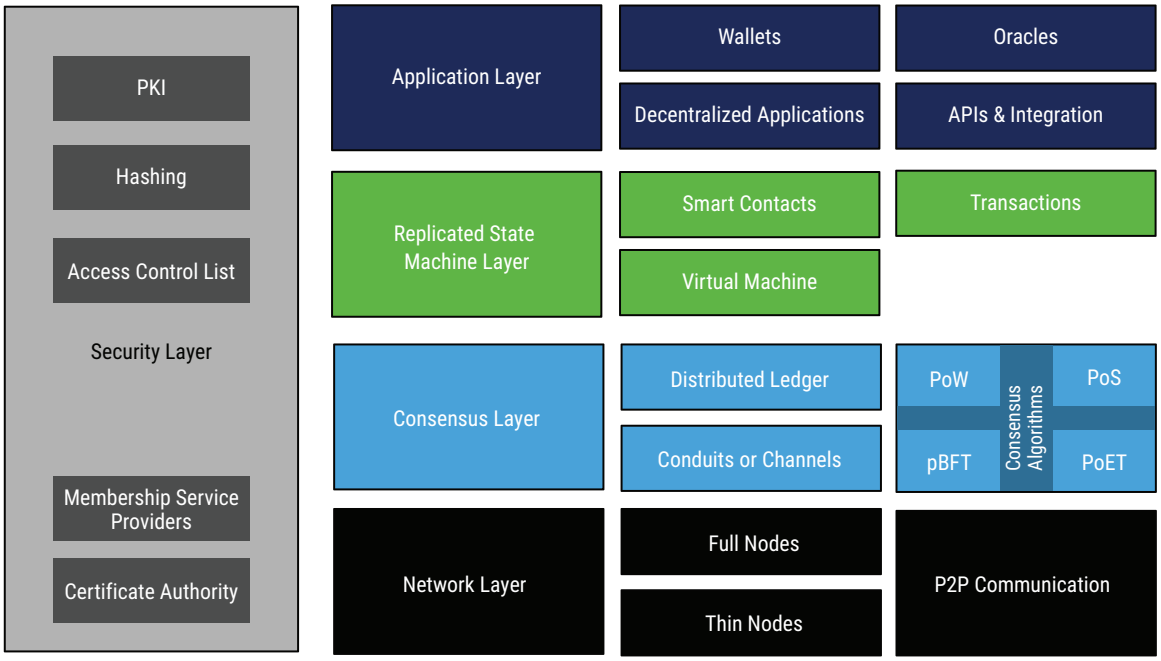
Chapter 4	Generic Blockchain Reference Architecture
Description	This chapter provides a generic blockchain reference architecture model that can be used across the various blockchain designs. (Refer to chapter 6, "Governance Model and Management Guidelines," as needed, for further explanations.)
Key questions answered	• What are the design considerations for a blockchain implementation? • How do the components work together? • What are the risk concerns and benefits of the various components?
Stakeholders	Stakeholders include, but are not limited to, the board of directors, executive management, business unit managers, cybersecurity personnel, IT managers/practitioners, system architects, developers, assurance providers, risk management personnel, regulators, and business or vendor partners.

In describing a generic blockchain reference architecture, it may be useful to briefly outline the current perspectives on blockchain generations, keeping in mind that these may change or evolve in type and number in the future. Blockchain platforms and solutions are typically grouped into five generations (categories) as follows:

- First-generation blockchains focus on the processing of transactions and maintaining records for specific cryptocurrencies. These blockchain platforms lack or have limited capability to create applications, that is, smart contracts, other than what the platform is designed to do. Most first-generation blockchains, such as Bitcoin and Litecoin, use PoW as the primary consensus protocol and likely derive the core design from Bitcoin codebase.
- Second-generation blockchains, that is, general-purpose blockchains, add on to the convenience of an immutable ledger with support for the development and deployment of smart contracts, allowing for many use cases. Ethereum is the earliest example of a second-generation blockchain.
- Third-generation blockchains attempt to make blockchains more efficient by addressing key issues, including scalability and cross-chain interoperability. For scalability, three key factors are considered:
  - Transactions per second writing to a block must be improved to scale the network.
  - The network must be scaled to address the growing demand for an increasing number of transactions conducted by millions of users.
  - Data scaling must be managed to reduce the amount of data that each node manages, which directly impacts consensus verification, network performance and transaction speed.
- Fourth-generation blockchains attempt to integrate artificial intelligence, Internet of Things (IoT) and blockchain capabilities. These projects are highly experimental, and it is not yet known what will ultimately emerge with the fourth generation of blockchain.

The generic blockchain reference architecture model, shown in **figure 4.1,** is used across the various blockchain designs, some of which may look familiar to IT professionals.

Figure 4.1—Generic Blockchain Reference Architecture Model

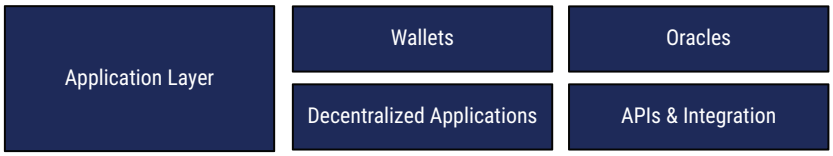


Source: Adapted from Balani, N.; "Blockchain Reference Architecture," *Medium*, 17 December 2019, https://medium.com/@naveenbalani/blockchain-reference-architecture-a3df6643532c

4.2 Application Layer

The application layer is where the user or application resides (**figure 4.2**). This is the interface with which the user wallet interacts for a decentralized application.

Figure 4.2—Application Layer



Source: Adapted from Balani, N.; "Blockchain Reference Architecture," *Medium*, 17 December 2019, https://medium.com/@naveenbalani/blockchain-reference-architecture-a3df6643532c

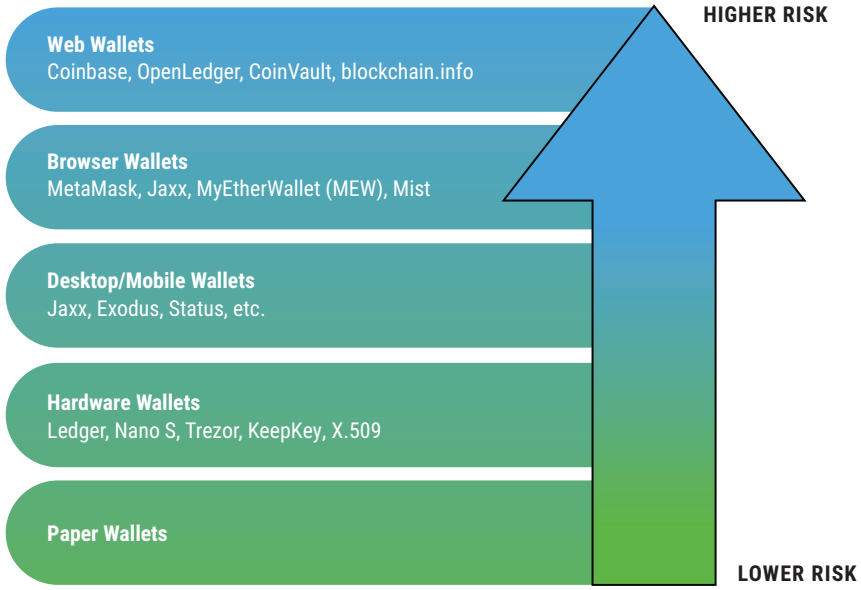
This layer holds four major subcomponents: wallets, decentralized applications, oracles, and application programming interfaces (APIs) and integrations.

4.2.1 User Wallet

The user wallet serves as the primary interface to the blockchain. The wallet controls access to a user's money by managing keys and addresses, tracking the balance, creating and signing transactions, and interacting with contracts. The wallet does not hold any cryptocurrencies, but it is the container for the user private keys and is the system for managing these keys. User wallets on a permissioned blockchain may be the smart card assigned to a specific user, where the user controls the decryption to present the private keys required for authentication or signing.

There are several types of wallets with varying degrees of usage risk, as shown in **figure 4.3**:

Figure 4.3—Types of Wallets



Web wallets are wallets that are integrated into web applications and exchanges. In web wallets, the operator of the application or exchange becomes the custodian of the wallet and manages the private keys on the user's behalf. These are also commonly known as custodial wallets.

Browser, desktop and mobile wallets are considered noncustodial wallets, in which the end users are responsible for the management of the private keys. The operators or developers do not have access to an end user's private key and are not able to recover it if an end user loses the key.

The above types are hot wallets, in that they are persistently connected to the blockchain.

Cold wallet types include hardware wallets and classic paper wallets. Cold wallets offer the best protection for end users because air gaps, or lack of connectivity, minimize any exposure in case of a user or network breach.

Wallet design has two categories:

- Nondeterministic (random)
- Hierarchical deterministic

Figure 4.4 shows the key attributes of the two wallet designs.

Figure 4.4—Wallet Design Categories	
Nondeterministic (random)	**Hierarchical Deterministic (HD)**
Private keys are randomly generated.	Multiple private keys are derived from a seed in a tree structure.
Requires a keystore of key parameters.	Uses mnemonic codes.
Uses key derivation function (KDF) to safeguard against brute-force, dictionary and rainbow attacks.	Relies on the entropy of the mnemonic codes (from 128 to 256 bits), checksum, plus the KDF to derive the 2^{512} binary seed of BIP-39.
Must be maintained and cumbersome to back up. Yields greater risk to the loss of private keys over time.	Can be regenerated from the seed words.
Not recommended for general use except for simple tests and experimentation.	Recommended for users.

Modern wallet design uses hierarchical deterministic design.

4.2.2 Decentralized Applications

Decentralized applications (DApps) are blockchain-enabled applications that leverage the immutability characteristic of the smart contract to provide transparency to the functions behind the application. A DApp is deployed as a website or web application that is linked to specific deployed smart contracts and comprises:

- **Front end**—Provides user interface (UX) built on traditional web technologies (such as HTML, CSS and Java scripts); using standards libraries, such as web3.js, the front end interacts with a remote blockchain node using either an HTTP or IPC connection
- **Back end for data storage**—Provides offchain data storage for sensitive and large data sets
- **Smart contract**—Provides the logic processing and transparency to the intent of the functions behind the DApp

Unlike traditional applications, where developers can alter and redeploy the applications unknowingly to its users, the business logic of the DApp cannot be altered without a redeployment of all involved smart contracts.

4.2.3 Oracles

Blockchain oracles provide a link between offchain and onchain data because blockchains and smart contracts cannot access offchain data. Oracles transmit information to smart contracts from sources in real time, significantly extending the capabilities of the blockchain ecosystem because they broaden the scope in which smart contracts can operate.

Oracles are not limited to just providing the data but act as an external authoritative source to query, verify and authenticate the data from one or more external sources and to relay the authenticated data to the requester. While most data flow from oracles to a blockchain is unidirectional, certain oracles can handle bidirectional exchange of information.

Oracles have two primary forms:

- Software oracles
- Hardware oracles

Software oracles extract online information from various sources and transmit the data to the blockchain. Use cases for software oracles include exchange rates, digital asset prices, location information and real-time flight information.

Hardware oracles obtain data from hardware devices, such as bar-code scanners and temperature and humidity sensors, and relay these data to the blockchain. Typical use cases for hardware oracles are supply-chain related, such as in the transportation of perishable goods.

The main challenge with designing oracles is that if the oracle is compromised, the smart contract relying on it is also compromised. This is important because oracles are not part of the main blockchain consensus; therefore, they are not part of the security mechanisms that public blockchains can provide. This is often referred to as the oracle problem.

The two arbitrage exploits on the decentralized lending protocol bZx in February 2020 resulted in a loss of US$1 million to the exchange and highlighted the importance of proper oracle design in the smart contract.[14] A smart-contract designer must always consider the use of multiple oracles to derive final decisions to mitigate man-in-the-middle attacks, where a malicious actor manipulates or gains access to the data flow between the oracles and the contract and modifies or falsifies the data.

4.2.4 APIs and Integration to Decentralize Services

An API enables information and tasks to be exchanged quickly and easily between an API provider, users and services, such as decentralized file storage (e.g., through IPFS,[15] SIA,[16] Storj[17] and Swarm[18]) and decentralized identity management (e.g., uPort[19] and Sovrin[20] from Hyperledger).

APIs help developers use existing functionality and data rather than using a work-around or building it themselves. While the implementation of an API can vary with applications, on blockchain exchanges, the user API key typically has three distinct levels of permission:

- **Readability**—Get user account and trade information.
- **Withdraw**—Authorize withdrawal of funds from user accounts.
- **Execute**—Allow execution of trades using programmable rules.

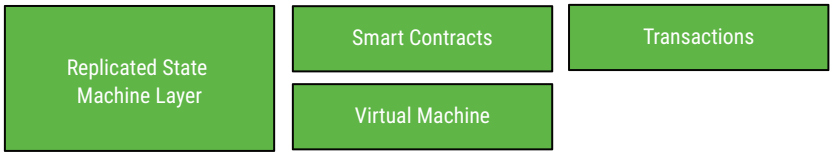
One or more permission levels may be granted when the user creates the API key. Poor API key management by the user or automated service was identified as the primary source of account breaches, such as the one incurred by Binance that resulted in a loss of US$40 million in May 2019.[21]

4.3 Replicated State Machine Layer

The replicated state machine layer of the generic blockchain reference architecture model manages the smart contracts that provide the business processing rules for decentralized applications. This layer interacts with three main components, as shown in **figure 4.5.**

[14] Heasman, W.; "Are the bZx Flash Loan Attacks Signaling the End of DeFi?" Cointelegraph, 22 February 2020, https://cointelegraph.com/news/are-the-bzx-flash-loan-attacks-signaling-the-end-of-defi
[15] IPFS, "IPFS Powers the Distributed Web," https://ipfs.io/
[16] SIA, "Decentralized Storage for the Post-Cloud World," https://sia.tech
[17] Storj Labs, "Decentralized Cloud Storage Is Here," https://storj.io
[18] The Swarm Cynny Space, "Swarm Technology," https://www.theswarm.co/swarm-file-system-object-storage.html
[19] Uport, "We Build Trust so You Can Grow," https://www.uport.me
[20] Sovrin, "Control Your Digital Identity," https://sovrin.org
[21] Lam, E.; "Hackers Steal $40 Million Worth of Bitcoin From Binance Exchange," Bloomberg, 8 May 2019, www.bloomberg.com/news/articles/2019-05-08/crypto-exchange-giant-binance-reports-a-hack-of-7-000-bitcoin

Figure 4.5—Replicated State Machine Layer



Source: Adapted from Balani, N.; "Blockchain Reference Architecture," *Medium*, 17 December 2019, https://medium.com/@naveenbalani/blockchain-reference-architecture-a3df6643532c

Using events, a smart contract can communicate, or broadcast, transaction state transitions to the underlying DApp. The DApp can also listen to these events and handle them accordingly.

Although implementation can vary with different blockchain designs, smart contracts are written in higher-level programming languages, such as Solidity, Go, Java, Plutus and others. Smart-contract execution and validation typically run within a virtual machine (VM) (e.g., Ethereum VM [EVM] for Solidity or Java VM for Corda), where the higher-level programming language is compiled into low-level machine instructions by the VM. Every full node maintains its version of smart contracts and the VM, but the contract states are shared across the blockchain network in world-state trie,[22] which provides a mapping between the addresses to the account states. The blockchain network can be thought of like a computer, and the world-state trie is like the hard drive of that computer.

The encoding process of this state information is handled using the trie data structure, which is a modified Merkle Patricia tree. Either the same or similarly derived applications of the concept are used for other blockchain designs, including Hyperledger Fabric, Corda and Ripple.

4.3.1 Upgradability of Smart Contracts

Because smart contracts are immutable, significant up-front planning must be done to ensure that smart-contract logic can be modified while preserving stored data and without the need for data migration. There are three general design patterns to consider in the upgradability of smart contracts:[23]

- **Primary-secondary contracts**—The primary contract holds the addresses of the secondary contracts and returns the address as needed. The secondary contracts always obtain the latest address of other contracts from the primary whenever they need to communicate with other contracts. Upgrading a contract requires deploying the new contract and updating the address in the master contract. A key limitation of this technique is the difficulty in migrating the data or asset of the contract to the new contract.
- **Eternal storage contracts**—The logic and the data contracts are separated from each other, allowing the logic contracts to be upgraded as required. Because the data contract is permanent and nonupgradable, any flaws, such as data structure or bug, can render the contract unusable.
- **Upgradable storage proxy contracts**—The eternal storage contracts act as a proxy to the logic contracts. Both the proxy contract and the logic contract inherit the same storage contract so that the storage references align in the EVM. The proxy contract has a fallback function that calls the logic contract so that the logic contract can make changes in the storage of the proxy.

[22] Woods, G.; "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Petersburg version 3e2c089, 5 September 2020, https://ethereum.github.io/yellowpaper/paper.pdf

[23] Gupta, M.; "How to Make Smart Contracts Upgradable!" Hackernoon, 6 September 2018, https://hackernoon.com/how-to-make-smart-contracts-upgradable-2612e771d5a2

4.3.2 Legality of Smart Contracts

In the United States, the basis for legal acceptance of smart contracts is derived from several key regulations:[24]

- Electronic Signatures in Global and National Commerce (ESIGN) Act[25]
- Uniform Electronic Transactions Act (UETA)[26]
- FDA 21 CFR Part 11—Electronic Records; Electronic Signatures[27]

The US regulations establishing the requirements for the validity of smart contracts are as follows:

- **Intent to sign**—Electronic signatures, like traditional wet ink signatures, are valid only if each party intended to sign.
- **Consent to do business electronically**—The parties to the transaction must consent to do business electronically. Establishing that a business consented can be done by analyzing the circumstances of the interaction, but consumers require special considerations, such as disclosures.
- **Association of signature with the record**—To qualify as an electronic signature under the ESIGN Act and UETA, the system that is used to capture the transaction must keep an associated record that reflects the process by which the signature was created, or generate a textual or graphic statement (which is added to the signed record) proving that it was executed with an electronic signature.
- **Record retention**—US laws on esignatures and electronic transactions require that electronic signature records be capable of retention and accurate reproduction, for reference by all parties or persons entitled to retain the contract or record.

In the European Union, the basis for legal acceptance of smart contracts is similar and is found in the following:

- **Electronic Identification, Authentication and Trust Services Regulation**—(eIDAS) (910/2014/EC)[28]

For the EU, the requirements for using smart contracts are simpler and require the following:

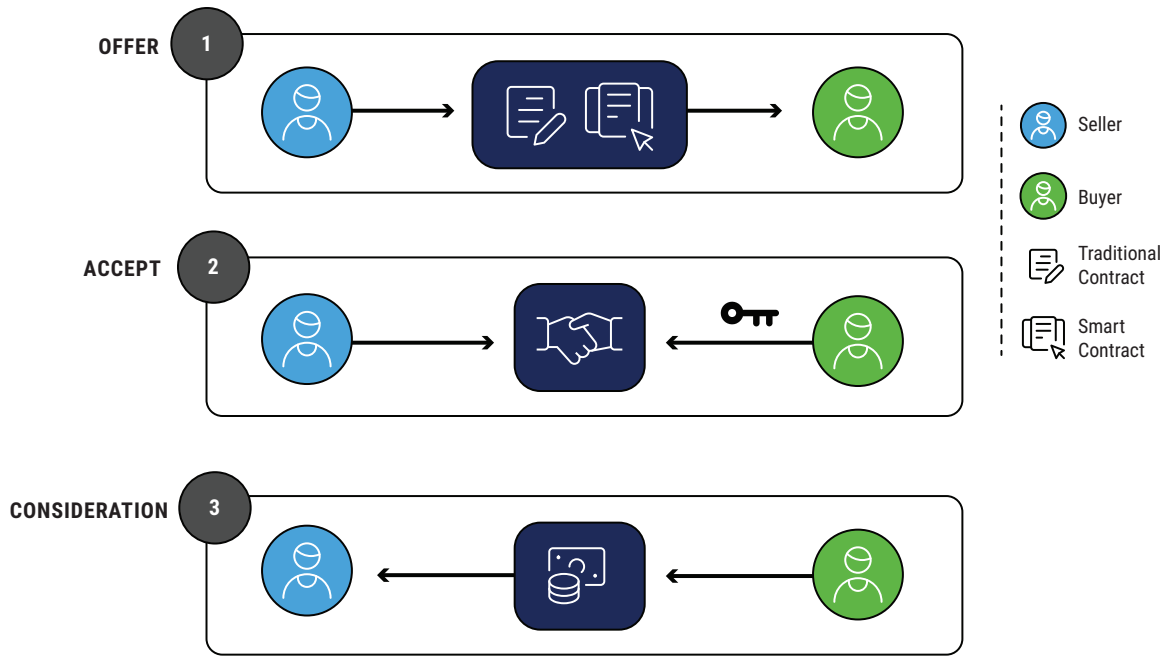
- **Basic electronic signatures**—Allows esignature to have the same legal effect and admissibility in legal proceedings.
- **Advanced electronic signatures**—Allows unique identification and authentication of the signer and enables verification of the signing using digital certificates, where the signing action is made using the user's uniquely held private keys.
- **Qualified electronic signatures**—Applies to certificates that can be used by a certification authority (CA), such as a hardware security token.
- **Electronic seals**—Similar to electronic signatures but applies to corporate entities.

These regulations related to smart contracts do not constitute a comprehensive list. Multiple jurisdictions around the world have new or evolving laws and regulations in this regard, and enterprises are advised to research where needed or consult with appropriate legal counsel as required.

With the core regulations in place, modern contract law requires three conditions to be present for the contract to be valid, as shown in **figure 4.6**.

[24] LabCFTC, "A Primer on Smart Contracts," 27 November 2018, https://www.cftc.gov/sites/default/files/2018-11/LabCFTC_PrimerSmartContracts112718_0.pdf
[25] FDIC, "FDIC Consumer Compliance Examination Manual," January 2014, https://www.fdic.gov/regulations/compliance/manual/10/x-3.1.pdf
[26] Uniform Law Commission, "Electronic Transaction Act," https://www.uniformlaws.org/committees/community-home?CommunityKey=2c04b76c-2b7d-4399-977e-d5876ba7e034
[27] FDA, "Part 11, Electronic Records; Electronic Signatures—Scope and Application," September 2003, www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application
[28] European Commission, "eIDAS Regulation (Regulation (EU) N°910/2014)," https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014

Figure 4.6—Key Conditions of a Modern Contract



1. **Offer**—If the smart contract is deployed on a blockchain network by the seller, and the buyer interacts with the contract, then, by definition, an offer exists between the two parties.
2. **Accept**—The buyer may indicate acceptance by signing the transaction with the private keys. This is the basis for electronic signatures in the United States and European Union.
3. **Consideration**—This is the exchange of something of value. For smart contracts, this may be a payment or commission from the buyer to the seller.

There are two models of how terms and conditions (T&C) are handled by smart contracts. In the external model, the traditional contract is the primary contract and provides for T&C between the parties with the smart contract, which automates certain aspects of the traditional contract. The specifics and extent of automation are defined in the T&C of the traditional contract.

In the internal model, the smart contract is the primary contract and recognizes the code as law, meaning that the smart contract encompasses the entire contract and a traditional contract has no legal effect. The internal model is the most common model used for smart contracts running on public blockchains.

4.3.3 Properties of Smart Contracts

Smart contracts must be Turing-complete in the VM. This requires the smart contracts to support data storage, conditional branching and calculations, given enough resources.

Smart contracts are immutable when deployed on the blockchain; therefore, they cannot be changed, disabled or removed without requiring hard forks. It is important to note that smart contracts can be terminated (or self-destruct) if preprogrammed with that task.

Smart contracts must be visible on the blockchain; however, by default, smart contracts are considered untrusted because the compiled bytecode is not human readable. To be trusted, the human-readable form of source code must be submitted to be compiled and verified against the deployed bytecode at a production contract address. Trusted smart contracts provide greater confidence for the intended use.

A smart contract must also be deterministic, that is, it always provides the same outcome of the execution for everyone who uses it. If the smart contracts require some random inference, the likelihood or probability of the outcome must still be the same for all users of the same contract.

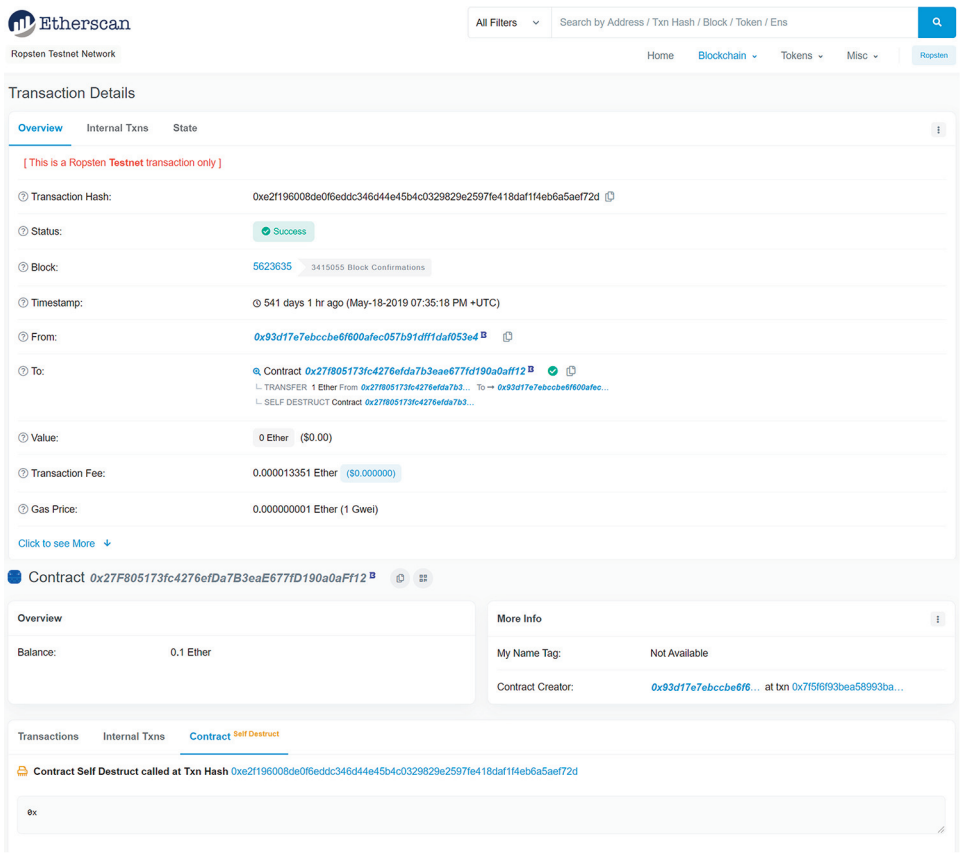
Smart contracts are atomic, meaning that one or more conditions defined by the smart contract must be met for the transaction to execute in its entirety. Any changes in the global state to the contracts and accounts are recorded only if all execution terminates successfully.

Smart contracts can interact with interfaces, including other smart contracts and APIs. Through APIs, smart contracts can also connect to traditional applications to bring additional functionality. The contracts can call or be called by other contracts. This feature can be both good and bad—capabilities can be extended by leveraging other smart contracts; however, an attacker can also craft a malicious smart contract to attack an existing smart contract.

A unique characteristic for Solidity and intended for Ethereum public blockchains is the smart contract self-destruct function, which is preprogrammed in the smart contract so that it can be deleted by the owner (see **figure 4.7**).

Deleting a smart contract does not remove the contract from the blockchain; deleting only flags the contract as self-destructed.

Figure 4.7—Self-Destructed Contract



Source: Etherscan, https://etherscan.io

All past-transaction history and the contract remain at the original address because the blockchain itself is immutable. Any value remains in the contract immediately before the self-destruct is sent to the smart-contract owner. Self-destruct is executed only once. Any value sent to a self-destructed contract is lost forever. Self-destructed contracts may have grave consequences for other smart contracts because the self-destructed smart contract may result in loss of functions, locked funds or frozen contracts (e.g., inoperable).

4.3.4 Transactions

Transactions typically follow one of two generally accepted transaction models,[29] depending on the design of the blockchain network:.

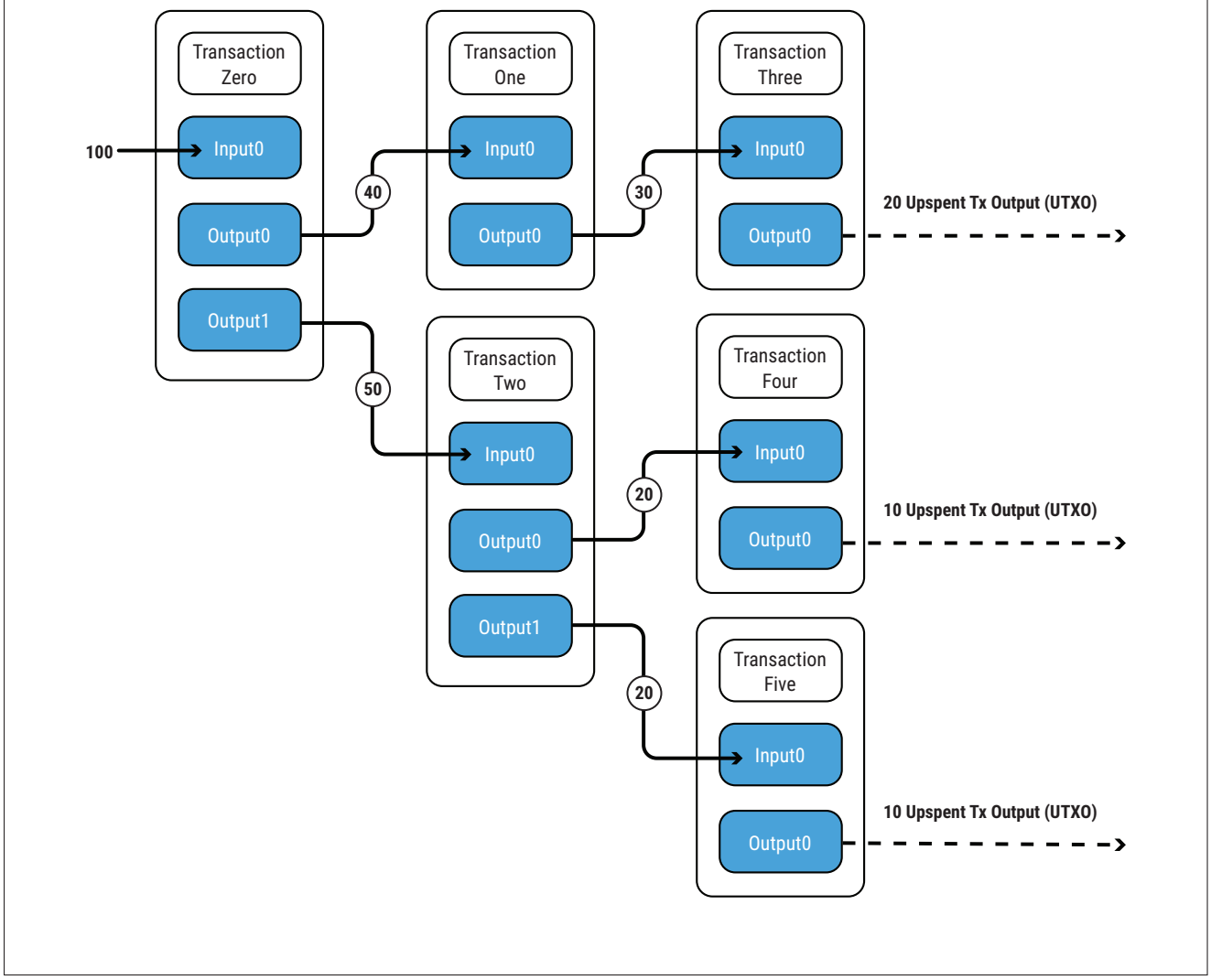
1. Unspent transaction output (UTXO) model
2. Account/balance model

The UTXO model, as shown in **figure 4.8**, is used by Bitcoin and its derivative chains, including LiteCoin, Bitcoin Cash and Cardano. Bitcoins are not stored in wallets; bitcoins move from transaction to transaction. An input is UTXO that the owner of the public key (user address) received from a previous transaction and can unlock (with a private key) and use as output for a transaction payment that the owner wants to make. Each transaction spends

[29] Sun, F.; "UTXO vs. Account/Balance Model," 14 April 2018, https://medium.com/@sunflora98/utxo-vs-account-balance-model-5e6470f4e0cf

output from prior transactions and generates new outputs that can be spent by transactions in the future. When a user does not have a UTXO, or a combination of UTXOs, that exactly matches the transaction payment value, the user must create output that exceeds the payment value and create two UTXOs—one paying the recipient the exact payment value, and the other sending the difference (i.e., change) back to the user.

Figure 4.8—UTXO Model



The account/balance model, similar to how traditional bank accounts are tracked today, maintains the balance of each account as a global state. The balance of the account is credit for any deposit from inbound transactions that send values to the account and debit the corresponding values from the account of any outbound transactions.

The UTXO model is generally more complex in implementation than the account/balance model. In the UTXO model, every send transaction (output) requires several computations across a set of unspent UTXOs belonging to the user account to determine the subset of UTXOs that will produce the desired output to meet the send transaction value. The UTXO model is stateless and is not well suited to applications with functions beyond asset issuance and transfer, which are required for stateful-oriented smart contracts.[30]

[30] Buterin, V.; "Thoughts on UTXO," 9 March 2016, https://medium.com/@ConsenSys/thoughts-on-utxo-by-vitalik-buterin-2bb782c67e53

In both models, transaction or network fees are also paid from the accounts and are added to the transaction before debit from the corresponding transaction or account, depending on the transaction model.

Blockchain transactions from Ethereum blockchain are triggered from interactions with externally owned accounts (i.e., addresses), which may be users, or from interactions with other smart contracts.[31] Unconfirmed transactions are maintained in a queue awaiting inclusion in a block. Accordingly, any unconfirmed transaction may be reverted. It is important to note that transactions waiting in a queue can also be read and thus possibly exploited using front-running of transactions, to the advantage of attackers.

The number of transactions per block varies with different blockchain designs, consensus algorithms and transaction complexity; however, the transactions are always time-ordered and prioritized based on the amount of fees the transaction is willing to pay. The higher the fees, the sooner the transaction is likely to be confirmed.

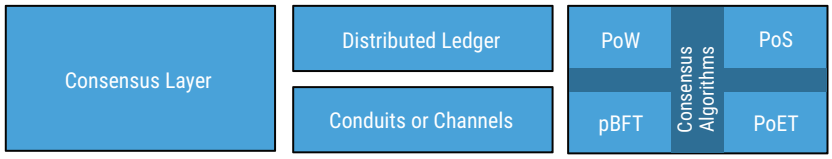
The transaction reaches finality when the transaction is included, that is, confirmed, in a block. Two types of finality exist:

- Chain-based protocols (Bitcoin, Ethereum) use probabilistic finality, which is based on the longest chain. The probability that a transaction will not be reverted increases as the number of confirmations increases—the deeper the block, the more likely that the fork containing that block is the longest chain.
- Practical Byzantine Fault Tolerance (pBFT)–based protocols (Hyperledger, Ripple, Stellar) rely on absolute finality, in which the transaction is considered finalized when it is included in a block after approval of a sufficient fraction of a committee of validators.

4.4 Consensus Layer

The consensus layer of the generic blockchain reference architecture model deals with validating transactions, ordering validated transactions, maintaining transacted records and providing decentralized trust to the network. The key components of this layer are shown in **figure 4.9**.

Figure 4.9—Consensus Layer



Source: Adapted from Balani, N.; "Blockchain Reference Architecture," *Medium*, 17 December 2019, https://medium.com/@naveenbalani/blockchain-reference-architecture-a3df6643532c

4.4.1 Distributed Ledger

At the core of this layer is the distributed ledger, which, in the context of a public blockchain, provides a decentralized and distributed database containing the transaction entries. These entries are recorded in the order of occurrence and composed into hashed blocks. The database or the ledger represents a chain of hashed blocks of transactions, with each block referring to the previous block in the chain. The ledger is shared across the blockchain

[31] GitHub, "Ethereumbook," https://github.com/ethereumbook/ethereumbook/blob/develop/04keys-addresses.asciidoc

network, which means every node has a copy of the ledger, and each node verifies the transactions independently. When every node agrees and confirms the authenticity of the transaction, the ledger is said to be in consensus.

4.4.2 Consensus Algorithms

Consensus algorithms provide the trust mechanisms that verify that blocks of transactions are valid and, therefore, the ledger is representative of the state of the blockchain. It is important to note that although there are different metaversions of consensus algorithms, consensus algorithms can be generically grouped into four categories:

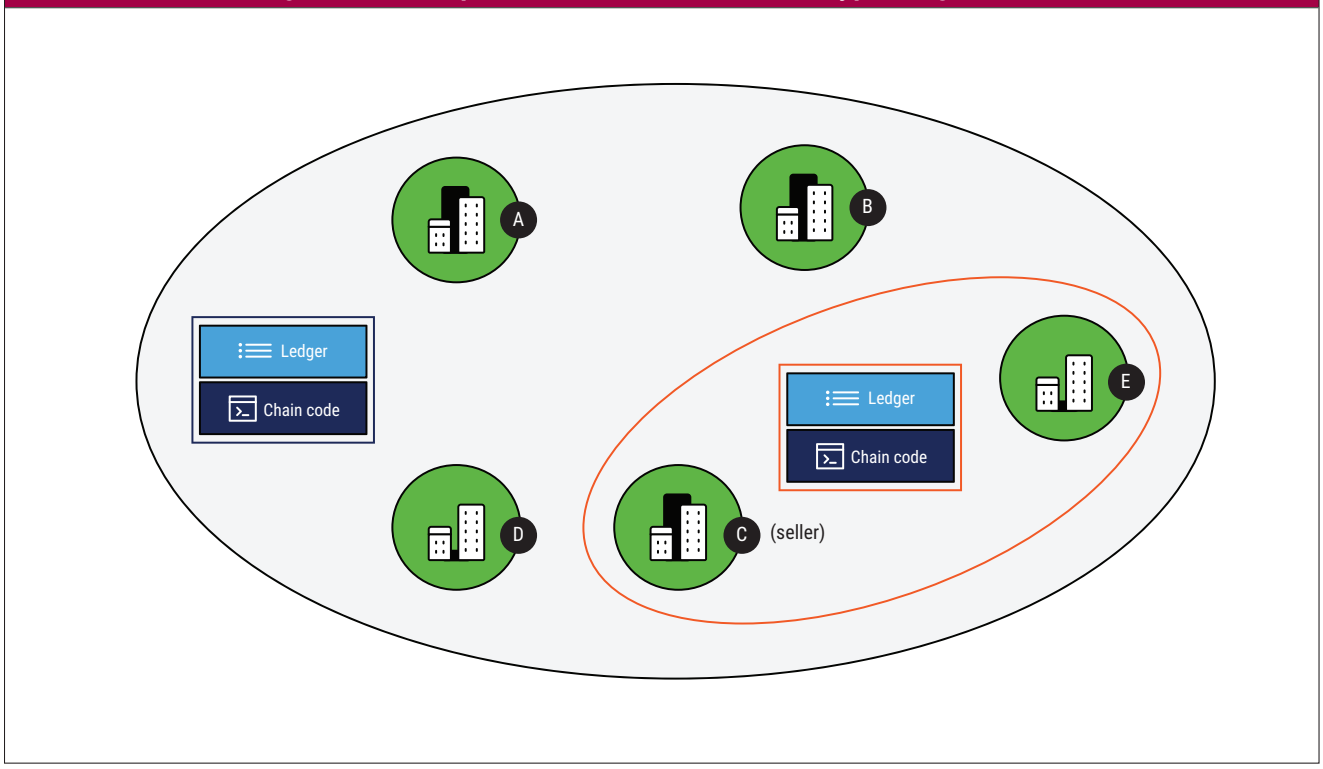
- Proof of work (PoW)
- Proof of stake (PoS)
- Practical Byzantine Fault Tolerance (pBFT)
- Proof of elapsed time (PoET)

The choice of consensus mechanisms directly impacts the transaction speed, the cost of the transaction and the blockchain scalability. The PoW mechanism is the slowest in terms of transaction speed and requires the greatest number of nodes to reach consensus, as noted for Bitcoin and Ethereum. PoET, PoS and pBFT mechanisms provide greater transaction speed and benefit from requiring a smaller number of nodes to reach consensus.

4.5 Ledger Conduits or Channels

Ledger conduits or channels can be thought of as private channels in the permissioned blockchain network where two or more nodes perform private transactions (**figure 4.10**).[32]

Figure 4.10—Implementation of Channel in Hyperledger Fabric



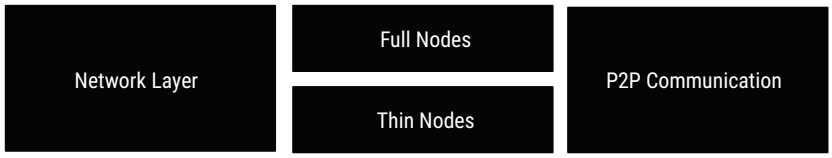
[32] Hyperledger Fabric, "Hyperledger Fabric Architecture Reference," Hyperledger Foundation, Release v2.2, https://hyperledger-fabric.readthedocs.io/en/release-2.2/channels.html

Multiple conduits can exist within a large blockchain network and intend to facilitate transactions that are visible only to the members of those conduits.

4.5.1 Network Layer

The network layer of the generic blockchain reference architecture model contains the nodes and the communication protocols that support the various blockchain activities, as shown in **figure 4.11.**

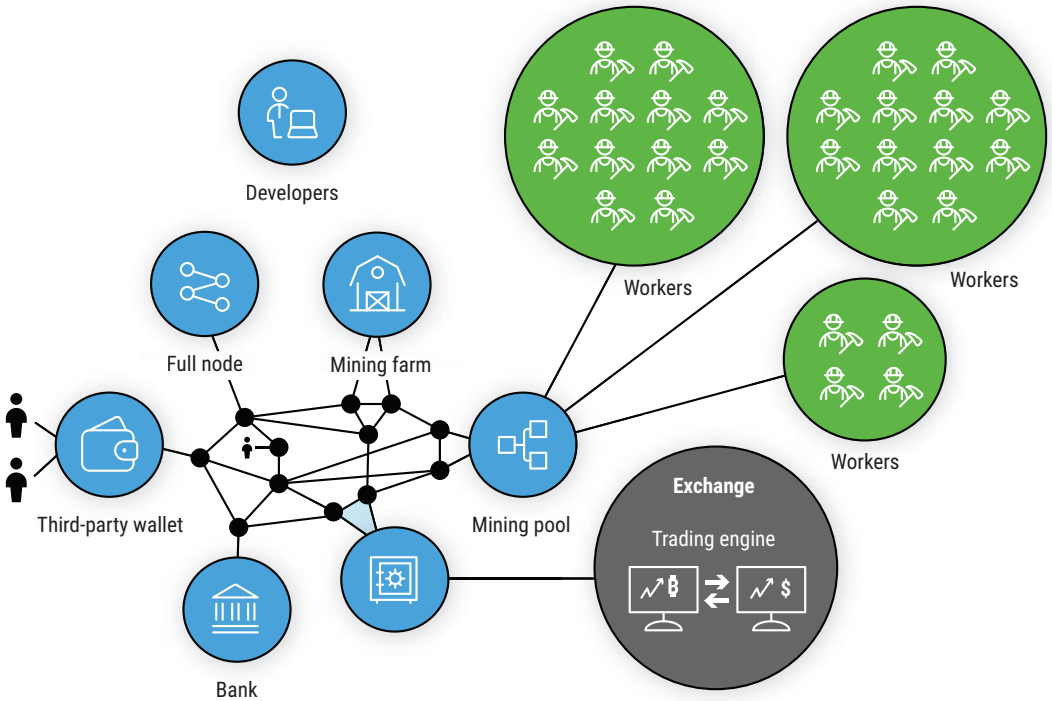
Figure 4.11—Network Layer



Source: Adapted from Balani, N.; "Blockchain Reference Architecture," *Medium*, 17 December 2019, https://medium.com/@naveenbalani/blockchain-reference-architecture-a3df6643532c

These nodes can vary from full nodes supporting consensus to specialized full nodes, such as wallet nodes, mining nodes, or bank and exchange nodes (**figure 4.12**).

Figure 4.12—Typical Topology for Second Generation Blockchain Network



Full nodes maintain a complete history of blockchain transactions dating back to the genesis node, whereas thin or light nodes hold only a portion of the recent blockchain transactions. Full nodes require significant network connectivity and data storage capability versus the fraction required in thin nodes. Full nodes verify that all preceding blocks are valid to guarantee that a transaction is valid, whereas thin nodes limit the verification to a certain depth of blocks and only check that the blocks are well formed.

The nodes are connected to the blockchain network through the Internet, where each node is uniquely identified by its IP address. Due to the distributed nature of the blockchain network, any nodes can connect, disconnect or reconnect at any given time. Each node independently maintains a list of peers with which it communicates.

Communication between nodes is with messages that are directed at the Internet addresses of the peers with which a node communicates, through the Border Gateway Protocol (BGP). The process repeats, with the receiving peer relaying the same message to its peers, and so forth. The messages are not always guaranteed to arrive at the address, or they may arrive in off-timed sequence or as duplicating messages. Regardless, peer nodes use timestamp and blockhash to correctly reorganize the messages to incorporate into their version of the ledger.

The peer-to-peer communication protocol allows the nodes to achieve the following purposes:

- Provide availability to the network by keeping connections alive.
- Establish new connections to enhance the security of the network.
- Distribute new information to other peers to support consensus.

4.5.2 Security Layer

The security layer of the generic blockchain reference architecture model securely connects the layers and provides the required features to authenticate and authorize any user or transaction request. The access control list serves as the underlying mechanism that binds the user identity to the permissions assigned based on the user role.

In a permissioned blockchain, the user identity is handled by public key infrastructure (PKI), where a trusted certification authority (CA) issues users the credentials in the form of smart cards. The CA for a permissioned blockchain is also referred to as a membership service provider. For a permissionless blockchain in which anyone can participate, user identity is inferred from the externally owned address and the possession of the corresponding private keys that are held in the user wallet. In certain blockchain designs, the hash of the user private keys or public keys is used to derive the externally owned address. Unlike permissionless blockchains, where the recovery of assets and user credentials or the rollback of transactions is nearly impossible, permissioned blockchains can be designed to provide mechanisms to facilitate these needs.

To provide integrity to the ledger, verified transactions are aggregated into a block, which is incorporated into the ledger based on an append-only approach and a time-order basis, using the hashes of the transactions and the hash of the block header of the prior block. Accordingly, the chaining of the current block to the previous block makes any attempt at altering past transactions extremely difficult and prevents tampering with the transactions after they have been accepted as valid.

Page intentionally left blank

Chapter 5
Interoperability Concerns

5.1 Introduction

Chapter 5	Interoperability Concerns
Description	This chapter defines the blockchain interoperability types and how such interoperability defines the choice of blockchain type and how that blockchain platform will operate and function with other platforms, as part of defined business and technical processes.
Key questions answered	• What is interoperability? • What are the types of blockchain interoperability? • What are the benefits and challenges associated with an interoperable blockchain platform?
Stakeholders	Stakeholders include business unit managers, IT managers/practitioners, security personnel, risk management personnel, and business or vendor partners.
Resources	• www3.weforum.org/docs/WEF_A_Framework_for_Blockchain_Interoperability_2020.pdf • www.ingwb.com/media/2667864/assessing-interoperability-solutions-for-distributed-ledgers.pdf

5.2 Interoperability/Infrastructure

Blockchain interoperability is the ability to exchange, access and make use of information across different systems and/or networks without the need for intermediaries. Blockchain interoperability enables the transfer of an asset between two or more networks or systems without changing the state of the asset.[33]

5.2.1 Platform Options

Following are the blockchain interoperability platform types:

- **Private/self-hosted**—Most corporate blockchains are manually provisioned, custom-built blockchain systems. Participants are known and trusted, but limited, which increases processing speed and scalability. A major drawback is the lack of decentralization and anonymity, which are present in public blockchains.
- **Managed blockchain service providers (e.g., AWS and Azure)**—These service providers offer fully managed services that make it easy to create and manage blockchain networks using popular open-source frameworks such as Hyperledger Fabric or Ethereum. Advantages are in:
  - Scalability—Easy to add peer nodes and replace poor performing nodes
  - Security—Key management service
  - Simplicity—Eliminates need for hardware and simplifies software deployments
- **Hybrid/multicloud blockchain**—This blockchain infrastructure leverages two blockchains. One blockchain is public and provides accessible visibility; the other is permissioned for transactions. This allows for security, speed and scalability, while providing flexibility and visibility.

[33] World Economic Forum, "Inclusive Deployment of Blockchain for Supply Chains: Part 6 – A Framework for Blockchain Interoperability," April 2020, www3.weforum.org/docs/WEF_A_Framework_for_Blockchain_Interoperability_2020.pdf

5.2.2 Data Ownership and Standards

Key to interoperability between blockchain systems are data ownership and standards. Data ownership, that is, the possession of and responsibility for the use and transaction of specific information, is a critical component that must be defined if multiple blockchains using different technologies and consensus methods are to interact. Given the relatively young nature of blockchain technology, there are no universally accepted standards for data ownership and blockchain interoperability, but several initiatives are addressing these issues, with more initiatives possible in the future. Following are two of these initiatives:

- **GS1**—This initiative is designed to improve the efficiency, safety and visibility of supply chains across physical and digital channels in 25 sectors. Using GS1 standards for identification numbers and data sharing provides a common language that identifies, captures and shares key information about products, locations, assets and more.[34]
- **Enterprise Ethereum Alliance**—The EEA is a member-driven standards organization whose charter is to develop open blockchain specifications that drive harmonization and interoperability for businesses and consumers worldwide.[35]

5.2.3 Data Normalization

Data normalization in the context of blockchain interoperability is an often overlooked, though critical, component. Like any data integration, a potential lack of standards or underlying common data schemas may make integration of data as part of interoperability difficult. For example, data lakes, which are repositories of large and varied sets of data in raw/native format,[36] and other such data groups often take significant effort and resources to achieve a uniform and normalized set. To accommodate the potential challenges of data normalization for blockchain interoperability, it is suggested that the disparate sets of data be identified early in the project process and that project teams and relevant stakeholders dedicate resources at the beginning of the project to identify data normalization challenges. Addressing this issue at the outset helps avoid significant interoperability and ultimately project delays.

5.2.4 Types of Interoperability

Following are the types of interoperability:

- **Digital asset exchange**—Distinct blockchain platforms can trade assets across disparate blockchain architectures. This requires publicly verifiable signatures from each platform.
- **Arbitrary data exchange**—Allows triggers of events (i.e., smart contracts) if the blockchains share a universal code verification and interpretation methodology.

This methodology sharing has proved to be inherently difficult due to signature verification challenges; however, sidechains (defined in the following section) are opening up several new possibilities, and are leveraged in the Cosmos Interoperability Project.

5.2.5 Approaches to Interoperability

Interoperability approaches include the following:

- **Cross-authentication:**
  - **Notary schemes**—A group of credible nodes act as notaries to attest that a specific event happened on blockchain 1 and prove it to the nodes of blockchain 2. The nodes agree to a select consensus mechanism and issue a verifiable signature. Most often, this is accomplished through sidechains.

[34] GS1, "Bridging Blockchains: Interoperability Is Essential to the Future of Data Sharing," www.gs1.org/sites/default/files/bridgingblockchains.pdf
[35] Enterprise Ethereum Alliance, https://entethalliance.org/
[36] RedHat, "What Is a Data Lake?" www.redhat.com/en/topics/data-storage/what-is-a-data-lake

  - **Relays**—Similar to notary schemes, relays allow reading, verification and action triggered by specific events (i.e., smart contracts) between disparate chains. Instead of leveraging credible nodes, relays execute changes on one chain in a manner that is readable by the other chains, through the storage and use of partial copies.
  - **Hashlocking**—Similar to relays, in that a credible trusted node is not required, hashlocking requires significantly less information sharing because it requires the exchange of only a single hash between chains. Events are triggered by the generation of cryptographic proof (a previously agreed-on hash). Hashlocking is used primarily for atomic swaps (defined later in this section).
- **API gateway**—Allows two applications or systems to share information through a shared set of classes, procedures, functions, structures or constants. This gateway can allow a blockchain to be a decentralized authority for external systems that need transactional data and histories.
- **Oracles**—Third-party services that find, verify and provide smart contracts with external information, such as real-world events. Although oracles are not data sources themselves, the sources can be software or hardware - based. Because oracles are simply bridges for offchain data to interact with or trigger blockchain-based systems (onchain), oracles are not considered truly interoperable.
- **Isomorphic cross-chains**—Security mechanism, consensus algorithm, network topology and block generation/verification logic are consistent between blockchains; interaction is relatively simple.
- **Heterogeneous cross-chains**—Interoperability between two independent blockchains, where the security mechanism, consensus algorithm, network topology and block generation/verification logic are different for each. They are complex to design and implement, and third-party tools may be necessary.
- **Sidechains**—Separate smaller-scale blockchains that operate alongside the primary blockchain. Sidechains include independent consensus mechanisms that are connected, or linked, to the primary chain. Some sidechain implementations can operate independent of their primary chain.
- **Proxy token**—A digital representation of an underlying asset that can exist on other blockchains. A proxy token allows distinct blockchains and token-based projects to interact freely and frictionlessly. An example is the Universal Protocol Alliance.[37]
- **Atomic swaps**—Peer-to-peer exchange of assets across separate blockchains triggered by predetermined rules, without the use of a third-party service, using self-enforced smart contracts. Requires an exchange of assets on both blockchains, or the transaction will not occur.

5.2.6 Existing Blockchain Interoperability Projects

Several existing projects meet the need for interoperability between blockchains. Following are descriptions of these projects:

- **ARK**—ARK describes its project as follows:[38]

  The ARK Ecosystem allows blockchain developers to customize a sovereign blockchain complete with required feature sets. ARK's value proposition lies in the array of services offered through the ARK Blockchain Platform. These services include interconnection with different blockchains, seamless integration of custom business logic, flexible development of tailored transaction types, and access to a global, community support system—all rooted in an intuitive development experience.
- **Cosmos**—Cosmos has been working on blockchain interoperability since 2014, with the release of Tendermint, which is a Byzantine fault-tolerant consensus engine and a peer-to-peer network gossiping protocol. The Cosmos Network has the Inter Blockchain Communication (IBC) Protocol to allow blockchains to interact with other blockchains. The network of blockchains communicates through IBC, with the Cosmos Network as the central hub. Blockchains are connected in a hub-and-spoke model to the Cosmos Hub.[39]

[37] Universal Protocol, "Connecting Digital Assets," https://universalprotocol.io
[38] ARK, "ARK Ecosystem Whitepaper," Version 2.1.0, 27 September 2019, https://ark.io/Whitepaper.pdf
[39] Kajpust, D.; "Blockchain Interoperability: Cosmos vs. Polkadot," 27 June 2018, Medium, https://medium.com/@davekaj/blockchain-interoperability-cosmos-vs-polkadot-48097d54d2e2

- **Open Application Network (OAN)**—A multitiered distributed blockchain platform created to address scalability, interoperability and privacy challenges that are associated with interactions between existing blockchain networks. OAN uses an AION token to allow parties to participate in the infrastructure. These parties are incentivized via the issuance of these tokens to act as miners and stakers (i.e., block creators).[40]
- **Polkadot**—Similar to Cosmos, the Polkadot Network uses the relay chain as the central connector, functioning like the Cosmos Hub. The blockchains connecting to the relay chain are named parachains. The Polkadot Network security is pooled and shared, meaning that separate chains can leverage collective security without having to start from the beginning to gain traction and trust from participants.[41]

[40] The Open Application Network, "Open Kits," https://developer.theoan.com/
[41] *Op cit* Kajpust

Chapter 6
Governance Model and Management Guidelines

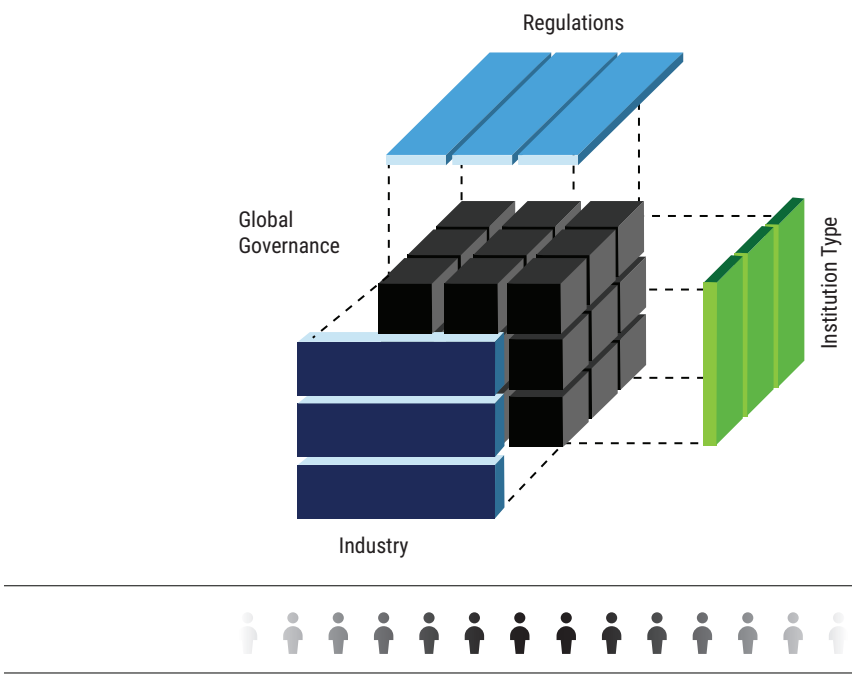
6.1 Introduction

Chapter 6	Governance Model and Management Guidelines
Description	This chapter describes the overall governance architecture of blockchain implementations and provides some principles for creating a common business vision for different types of blockchain implementations. This chapter also analyzes the importance of governance at the business architecture and technical architecture levels.
Key questions enterprises need to ask	• How are the governance models relevant for various types of blockchain implementations described and analyzed? • What governance models are most appropriate for the use case being considered? • Who are the responsible parties for governance definition and process at both the business and technical levels? • How is ongoing governance management defined and implemented?
Stakeholders	Stakeholders include, but are not limited to, the board of directors, executive management, business unit managers, cybersecurity/information security practitioners, IT managers/practitioners, assurance providers, risk management personnel, regulators, business or vendor partners, and miners/node.
Resources	Numerous frameworks were used as resources for this chapter, including COBIT 2019, NIST CSF V1.1, ISO/IEC 27001 and SOC 2.

Broadly speaking, governance is the method by which an enterprise ensures that stakeholder needs, conditions and options are evaluated to determine that balanced, agreed-on enterprise objectives are achieved. Governance involves setting direction through prioritization and decision making and then monitoring performance and compliance against the agreed-on direction and objectives.

Figure 6.1 provides a sample representation of the overall governance architecture for blockchain implementations.

Figure 6.1—Governance Model



Global governance for any blockchain implementation should be looked at in a multidimensional way, across regions, to make sure the implementation is properly sponsored and implemented and has buy-in.

Figure 6.2 provides the three major areas to review.

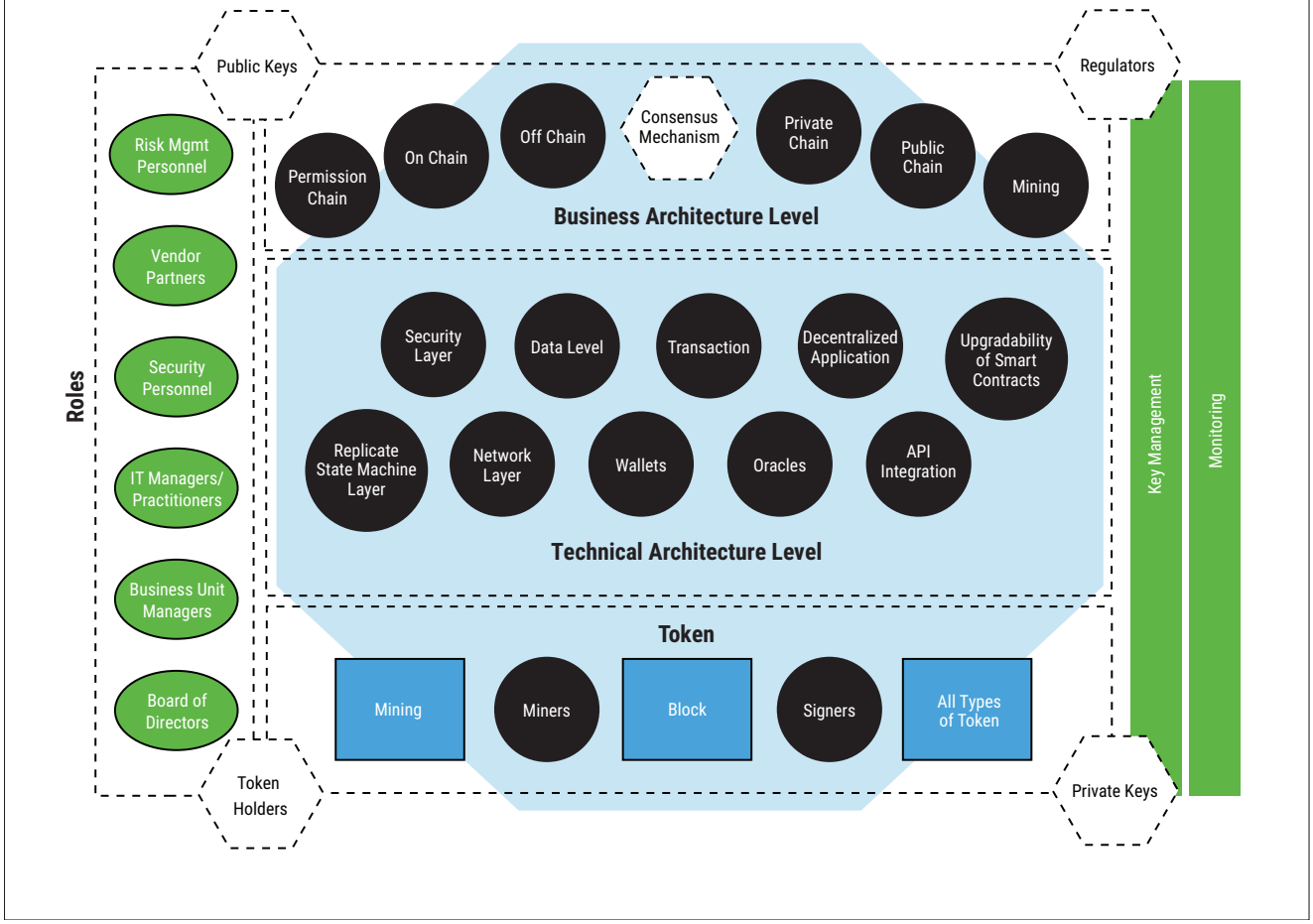
Figure 6.2—Governance Areas

1	Industry	For which industry is the solution being considered? What are the trends in that industry? Does the enterprise get a competitive edge in that industry? Are industry key leaders on board with blockchain in their world?
2	Regulation	Every industry has different regulations that the solution must conform to. For example, in the United States, the SEC reporting requirements must be taken into consideration and, in the US pharmaceutical sector, it is 21 CFR Part 11.
3	Institution type	The type defines the client/user it services. The institution has different challenges by industry or regulation, or globally. For example: A bank incorporated in the United States has branches in Japan; therefore, it also has to conform to Japanese banking laws and regulations.

This architecture is divided into three main levels—business architecture, technical architecture and token level, as shown in **figure 6.3**.

Blockchain governance models are becoming an ever more important part of the enterprise discussion across multiple disciplines. Governance represents the incentive mechanisms and ultimately the economics by which a blockchain network operates and the rules associated with the ongoing functioning of the network. It is important to see these various perspectives across the business architecture, technical architecture and token levels.

Figure 6.3—Governance Levels

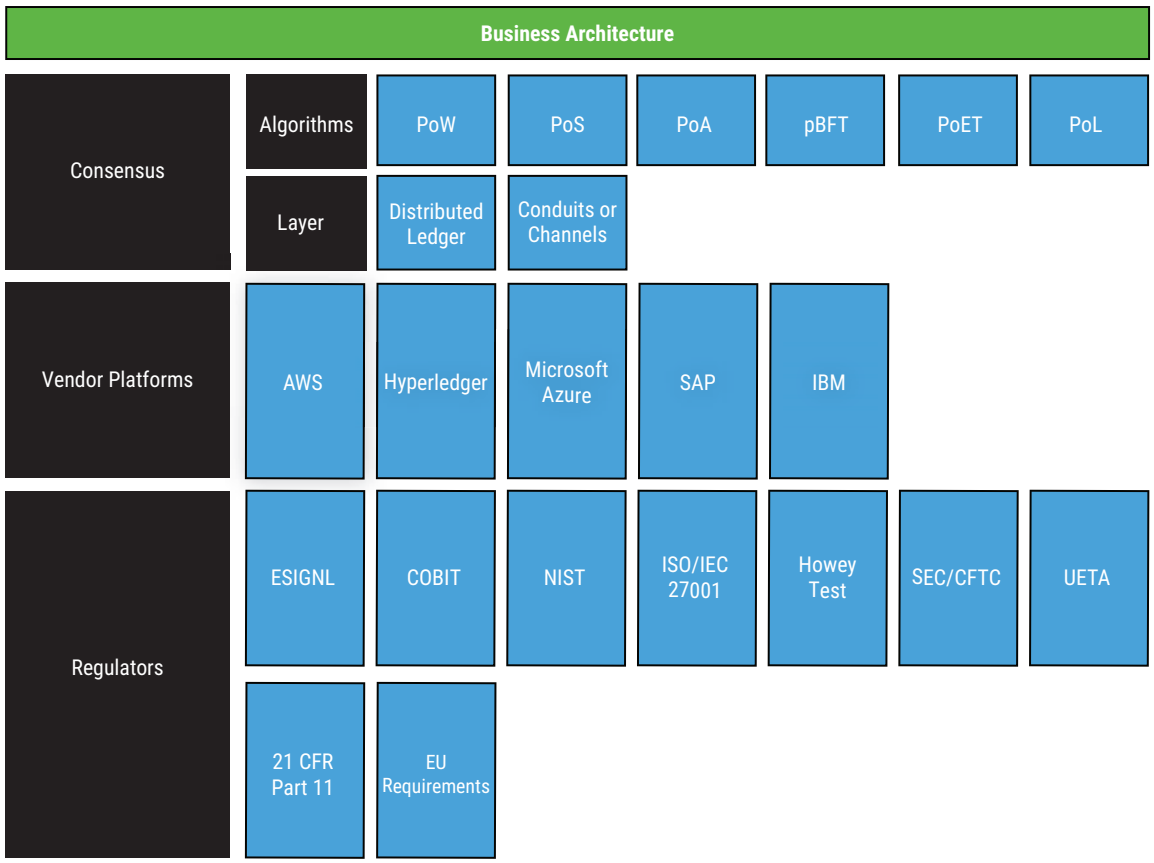


6.2 Business Architecture Level

Viewed through a business architecture lens, using a framework that guides the decision-making process is key. It does so by providing a clear delineation of roles and responsibilities and ensuring that there are agreed-on processes to reach and communicate decisions. See **figure 6.4**.

Governance at this level represents the top of the decision-making pyramid. This tier requires executive sponsorship from the various organizations and business units and is responsible for ensuring that the vision and strategy align with network objectives. It should also be focused on ensuring that business benefits are realized.

Figure 6.4—Business Architecture



Independent of process complexity, another consideration that should be addressed is whether decision making is centralized or decentralized. Distributing the power of decision making may make the process seem fair, reduce the risk of undue control and encourage free thinking, but in doing so, it may create delays in the achievement of a consensus. Thus, it is critical to address the definitions and challenges of offchain versus onchain governance. Given its distributed nature, blockchain technology offers unique solutions for governance models:

- **Offchain governance**—This type of blockchain governance model is designed to offer a balance of governance control among participants in a particular blockchain ecosystem. These participants often include the blockchain miners, core developers, business units and users. It has been argued that Bitcoin and Ethereum follow the offchain governance model,[42] because it allows for code changes that must be agreed on by developers and then accepted by the miners and community. Thus, there is a level of socialization or informality with this type of governance and, therefore, a relatively greater degree of centralization of governance authority.
- **Onchain governance**—This is a relatively new method by which a blockchain can operate. Onchain governance means that the rules of governance of the blockchain are hard-coded or written directly into the blockchain protocol. This is widely considered to be far less centralized and more formal than offchain from a governance perspective. In practical terms, this means that all decisions regarding the blockchain (e.g., changes to the size of a block) are submitted through code updates, which all nodes can vote to accept or decline. In certain models, not all nodes have equivalent voting power, but for simplicity of discussion, it is assumed that they do. If the

[42] Pool of Stake, "Revisiting the On-Chain Governance vs. Off-Chain Governance Discussion," Medium, https://medium.com/@poolofstake/revisiting-the-on-chain-governance-vs-off-chain-governance-discussion-f68d8c5c606

majority of nodes accept this code change, then it is included in the blockchain going forward. There are occasional challenges and, in some instances, rollbacks to prior code versions if the proposed change is not successful. Onchain governance works entirely online. There is no socialization aspect to this type of governance. Only a distributed model like blockchain can allow for onchain governance, which was not possible in prior technology governance models. The blockchain world is looking forward to the further development of onchain governance models and how they evolve.

In the context of business architecture and governance, it is important to give considered thought to the aspects of the stakeholder network that are relevant to and directly impact how governance is achieved at this level. These aspects include but are not limited to:[43]

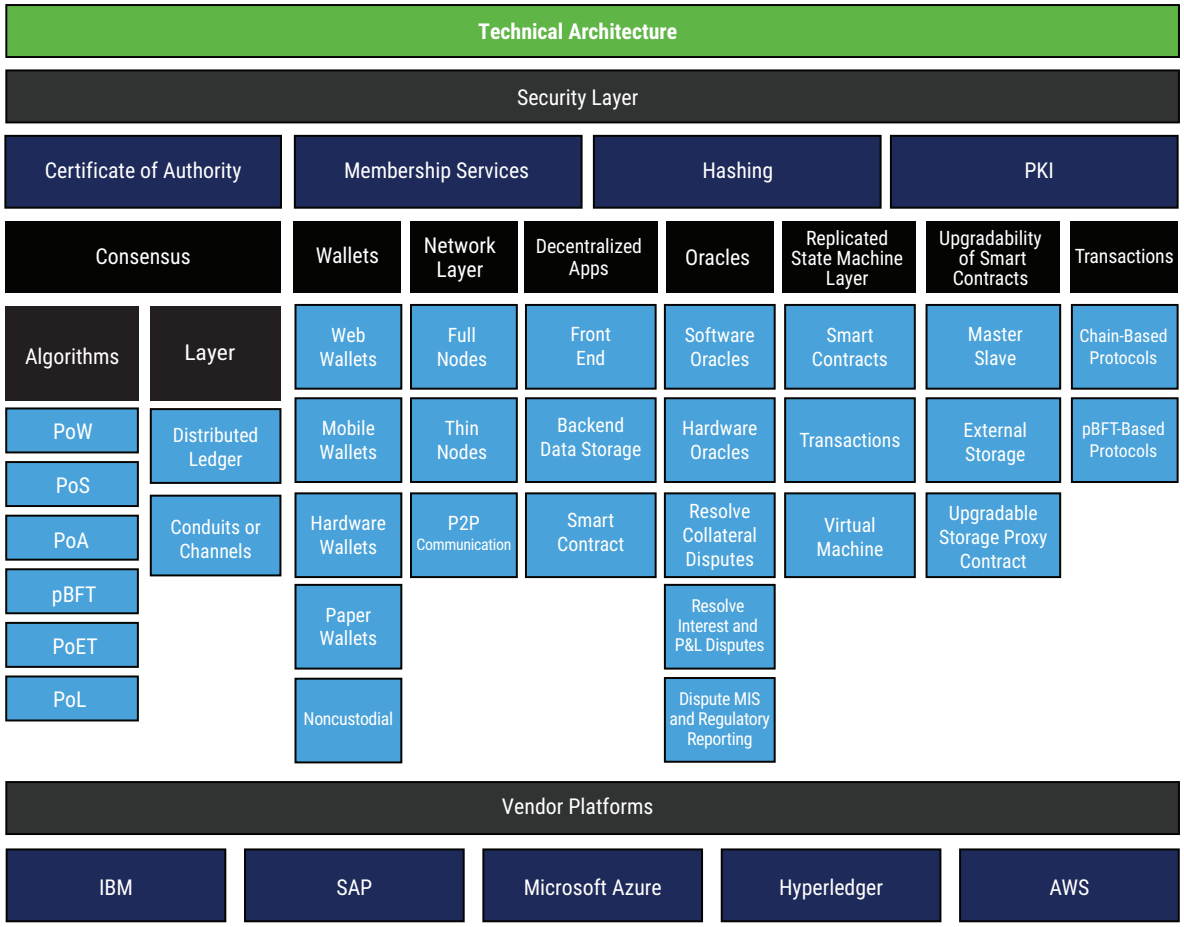
- **Membership life cycle**—Decisions associated with the process of onboarding and offboarding participants to the network.
- **Funding and fees**—Decisions focused on how the network will be funded. This may cover areas such as centralized infrastructure, common services and staffing.
- **Regulation**—Most industries need to meet specific regulations that are often geographically bound. This category focuses on key decisions to ensure that these regulations are met and enforced.
- **Education**—Decisions on the level of training to provide to members and external organizations regarding the use of and integration into the network.
- **Dispute**—Because disputes are almost always unavoidable, these decisions deal with the resolution process.
- **Cost versus risk**
- **Competition versus cooperation**
- **Formalism versus agility**

6.3 Technical Architecture Level

The technical architecture level focuses on converting the vision into a program with milestones that meet the requirements of the network (**figure 6.5**). This normally involves business stakeholders, directors, IT architects and legal counsel.

[43] See also chapter 2, "Stakeholders and Stakeholder Management."

Figure 6.5—Technical Architecture



Key technical governance themes across these areas are shown in **figure 6.6.**

Figure 6.6—Security Layer

Area	Description	Method	Network/Node Operator (where applicable)
Spoofing	Use of a token or other credential to pretend to be an authorized user or compromise a user's private key.		Manage certificate revocation list distribution among network participants to ensure that revoked members can no longer access the system.
Tampering	Modify information (e.g., an entry in the database).	Use of cryptographic measures (SHA256, ECDSA) to make tampering infeasible.	
Repudiation	An entity cannot deny who did what.	Tracks who did what using digital signatures.	

Figure 6.6—Security Layer *(cont.)*			
Area	**Description**	**Method**	**Network/Node Operator (where applicable)**
Replay attacks	Replay the transactions to corrupt the ledger.	Read/write sets to validate the transaction. A replay of transactions will fail due to an invalid read set.	
Information disclosure	Data exposed through intentional breach or accidental exposure.	Several blockchain solution platforms provide support for using TLSv1.2 for in-transit encryption. It does not encrypt ledger data at rest (the operator's responsibility). Information about all peers in the system and their transactions is exposed to the ordering service.	It is the operator's responsibility to prevent information disclosure by following information security best practices and at-rest encryption. It needs to be drafted in the agreements executed—user level or service agreements.
Denial of service	Makes it difficult for legitimate users to access the system.	Operator's responsibility.	It is the operator's responsibility to prevent denial of service to the system.
Elevation of privileges	Gain high-level access to the application.	Issued identities cannot upgrade their access (e.g., create an identity) without manual review of access.	Permissioning is set up based on the type of chain environments. It is the responsibility of the network/node operator to limit access.
Ransomware	Using cryptographic or other means to prevent access to data on the file system.	Operator's responsibility.	It is the operator's responsibility to ensure that ransomware cannot prevent access to a node's ledger.

6.3.1 Network Layer

The ordering service can view all transactions (hashes or key/value pairs) across all channels that it serves. If it is necessary to hide the transaction data from the ordering service, only hashes of the read/write set in a transaction should be sent to the ordering service while exchanging the data directly between peers.

After an ordering service is established for a network, it must be configured with the digital identities of peers of founding members. This is typically done by configuring the digital certificates of peers in the ordering service genesis blocks. The peers must also be configured with the digital identity of the ordering service.

6.3.2 Data Level

The entities must agree on a data model that will be stored in a blockchain, which in turn is determined by the chain code. The founding members of a network or a channel that is deploying a chain code determine the key/value pairs that are stored in a channel. Furthermore, the members decide which data they will share with other members and which data they will keep private among themselves or a subset of members. The data model should be devised so that it is useful for the business functions that members desire to accomplish, is reasonably future-proof and does not inadvertently leak information. All participating peers in a channel store the committed transactions (and their key/value pairs).

A process for defining the data model that will be stored in a channel should be established. The steps can be summarized as follows:

1. Determine who will run the ordering service.
2. Configure digital identities of founding members in the ordering service.
3. Create channels and determine the channel policy for admitting new members.
4. Define the governance for writing, distributing, deploying and instantiating chain code.

This process can be formalized into chain code governance to require mandatory reviews from all relevant members who instantiate the chain code on their nodes.

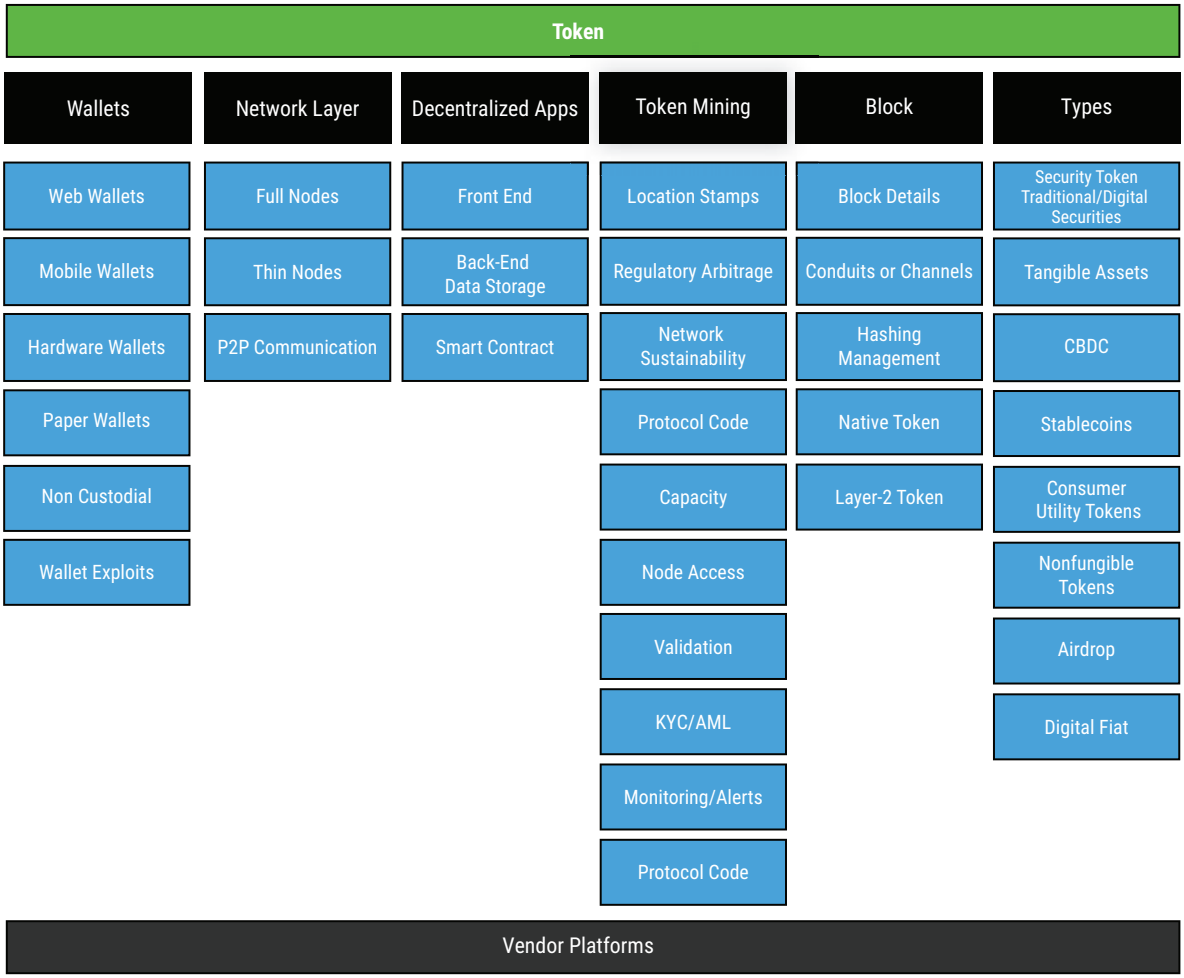
Establish a process for deploying chain code on a member's peer, including manual reviews and verification of a digital signature of the chain code author.

6.4 Token Level

Governance for the token level should be considered across the different areas in the matrix shown in **figure 6.7.**

Note: For more details on wallets, see the section in chapter 8 on digital tokens.

Figure 6.7—Token Architecture



6.5 Monitoring

Monitoring focuses on the day-to-day activities of the operation of the network. At this level, the focus is on aspects of the design, build and operation of the network. Monitoring includes various stakeholders from business, legal and technical teams. Tasks include the following:

- Enforcing standards
- Smart contract code reviews
- Deployment planning
- Organization onboarding
- Security audits
- Reporting

6.6 Roles

At the time of creation of a network or a channel, the founding members must define the policy on admitting new members to the network or channel. By default, this policy is the one chosen by the majority (two out of two, two out of three, three out of four and so on). The members may decide on any other policy for admitting new members to the network. Any change in the policy to admit new members is typically decided through a business agreement. When an agreement is reached, the channel configuration can be updated per the current policy to reflect the new policy for admitting new members.

The creation of the genesis block and subsequent transactions to update configurations are privileged operations and must be approved by the peer administrator before being confirmed.

6.6.1 Creating the Network

When entities decide to create a network, they must decide on the following:

- Who will run the ordering service?
- How many different instances of ordering service will be in the network?

The role of the ordering service is critical because, depending on the configuration, it has visibility into transaction hashes or transaction data across all channels that flow through it. The entities deciding to form a network may choose to trust one of the entities to act as the ordering service; they may also decide to trust a neutral third party to run the ordering service.

6.7 Interoperability Governance

In all blockchain implementations, interoperability governance must conform to the existing infrastructure or global interoperability precedents for the enterprise. There are multiple matters to consider, such as platform limitations (Hyperledger and AWS, or a traditional Unix environment).

One example is General Data Protection Regulation (GDPR) considerations. GDPR[44] is a European Union law that defines how personal data are acquired, processed and ultimately erased from a computing system. The definition of personal data in GDPR is quite broad. Examples include name, email address and IP address.

Blockchain, by design, creates an immutable, permanent and replicated record of the data. A blockchain network based on Hyperledger Fabric encompasses these three properties. Thus, storing personal data on a blockchain network that cannot be deleted or modified can be challenging from the perspective of GDPR and other compliance requirements to which many must adhere. Similarly, it is important to know with whom that personal data are shared.

The channel and the channel private data feature of Hyperledger Fabric provide a mechanism for determining the entities with which data are shared. Channel private data are never stored on a blockchain, but their cryptographic hashes are stored on the chain. Through a governance process, peers can determine the other peers with whom to share these data. The channel private data feature in Hyperledger Fabric can potentially provide a mechanism to store personal data off the chain, determining with whom these data are shared, while maintaining the integrity of the data through cryptographic hashes stored in the blockchain.

[44] Intersoft Consulting, "General Data Protection Regulation GDPR," https://gdpr-info.eu/

Chapter 7
Security and Privacy Considerations

7.1 Introduction

Chapter 7	Security and Privacy Considerations
Description	This chapter provides a list of common security and privacy concerns, examples and mitigation techniques specific to blockchain technologies.
Key questions answered	• What are the different attacks on blockchains and how can they be mitigated? • What can enterprises implement to protect users from threats while using blockchain applications? • How can developers implement more secure smart contracts? • To what extent can privacy be safeguarded on a blockchain?
Stakeholders	Stakeholders include, but are not limited to, the board of directors, executive management, business unit managers, cybersecurity personnel, IT managers/practitioners, developers, assurance providers, risk management personnel, regulators, and business or vendor partners.
Resources	• https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010-2017%20(en).pdf • Consensys Known Attacks • Consensys Smart Contract Best Practices

7.2 Infrastructure/Network Level

The infrastructure/network of a blockchain provides the backbone components from which transactions are facilitated and processed. Accordingly, these infrastructure components are prone to various forms of attack designed to impact the availability and/or integrity of transactions.

7.2.1 51% and Long-Range Attacks

A 51% attack[45] is a double-spending attack against the proof-of-work consensus algorithm and can impact first and second-generation blockchains. The attack allows the attacker to spend cryptocurrency on the blockchain network without having to give up any money. A long-range attack takes an approach that is similar to the 51% attack; however, instead of beating the required number of block confirmations, the attack tends to fork the chain from the genesis block. There have been several 51% attacks, including multiple attacks on Ethereum Classic in August 2020.[46]

The blockchain network is most vulnerable to attack if the network is not fully decentralized, especially during chain reorganization events such as network upgrade, where miners may need to update their node software to support the ongoing PoW consensus algorithm.

[45] Poston, H.; K. Bennett; *Certified Blockchain Security Professional Official Exam Study Guide,* Blockchain Training Alliance, 2018, https://blockchaintrainingalliance.com/products/cbsp-official-exam-study-guide

[46] Voell, Z.; "Ethereum Classic Hit by Third 51% Attack in a Month," updated 31 August 2020, https://www.coindesk.com/ethereum-classic-blockchain-subject-to-yet-another-51-attack

Anti-Pattern

The term anti-pattern refers to "any repeated (but ineffective) solution to a common problem."[47]

The 51% attack (**figure 7.1**) typically executes in four steps:

1. The attacker creates the malicious version of the blockchain and continues to add blocks; however, the attacker does not broadcast the malicious blockchain.
2. The attacker then conducts the transactions on the legitimate version of the blockchain. This is when the attacker spends legitimate cryptocurrencies on digital goods and allows the transactions to be confirmed. On the malicious chain, the attacker does not record any of these transactions because they do not exist.
3. The attacker extends the malicious version of the blockchain longer than the legitimate version using additional hash power, which can be obtained through the hacking of legitimate devices or leased from dark web providers.
4. When the malicious version is longer than the legitimate version, the attacker broadcasts the malicious version to the blockchain network.

Because the blockchain policy complies with the longest chain rule and the malicious version is longer than the true version, the blockchain network is likely to adopt the malicious version and erase any existence of the attacker's spending.

Figure 7.1—51% and Long-Range Attacks OWASP Mapping

A10:2017-Insufficient Logging and Monitoring[48]	Chain reorganization event provides unique opportunity for attackers to launch 51% attack because the network hash rate is drastically reduced, as demonstrated in January 2019 with Ethereum Classic. In addition, network monitoring and response are likely delayed because the project team is likely to focus on supporting the various rollout issues relating to the reorganization.

Mitigation

The best mitigation of the 51% attack is to monitor and ensure that the hash rate of the network is maximized and is geographically and noncentrally distributed, so that no single entity or entities can collude and seize control of at least 51% of the network hash rate. Layer in additional safeguards, such as checkpointing and notarizing of blocks. However, it is important to note that, although the notarizing of blocks or checkpointing does not prevent 51% attack, its usage can reduce the impact by keeping the attacker from rewriting too much of the ledger history.

Consider using alternative consensus algorithms, such as proof of stake or pBFT, or using trusted validators or known nodes; however, that may require a tradeoff by reducing the decentralization of the network. Mechanisms to prevent collusion and ensure randomness and fairness in the selection of block creators are critical to preventing a 51% attack on PoS consensus.

Require a certain number of confirmations for any given transaction before final acceptance. The number of confirmations is dependent on the blockchain network average transaction time and should scale based on the amount being transacted. On the Bitcoin network, six confirmations are safe and secure to prove that the transaction is valid and permanent, but the same transaction on the Ethereum network requires 37 to 40 confirmations to achieve the same.

[47] National Cyber Security Centre, "Security Architecture Anti-Patterns," 22 May 2019, UK, www.ncsc.gov.uk/whitepaper/security-architecture-anti-patterns#:~:text=The%20term%20'anti%2Dpattern,Reusable%20Object%2DOriented%20Software'

[48] OWASP, "OWASP Top Ten 2017," https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010-2017%20(en).pdf

7.2.2 Eclipse Attack

Blockchain nodes connect to their neighboring peers via a peer-to-peer communication protocol to process transactions and maintain their own blockchain ledger. At a given time, a full node typically talks to between eight (Bitcoin) and 13 (Ethereum) adjacent nodes, depending on the network. An eclipse attack seeks to isolate the victim node by blocking its access to its adjacent peers and then trick the victim node with false transaction information. For example, the Bitcoin blockchain was the target of an eclipse attack in 2019.[49]

Anti-Pattern

The eclipse attack (**figure 7.2**) typically executes in four steps:

1. The attacker identifies the neighboring nodes of the victim node.
2. The attacker launches a distributed denial of service (DDoS) against the peers and places its malicious nodes between the peers and victim node, thereby isolating the victim node.
3. The attacker waits for the victim node to recognize the malicious nodes as peers, by examining the peer connections on its malicious nodes.
4. The attacker conducts transactions with the victim node but falsely submits transaction information from the malicious nodes, tricking the victim node to believe that actual transactions are taking place and confirmed.

Figure 7.2—Eclipse Attack OWASP Mapping	
A4:2017-XML External Entities (XXE)	Blockchain nodes rely on messages communicated to neighboring peers. By masquerading as legitimate peers, malicious peers can provide inaccurate and false transactions information.

Mitigation

Increasing the number of connected peers and randomization of node selection can reduce the likelihood of an exploit, because it requires the attacker to scale to at least the same or larger number of malicious nodes. An exploit attack can be mitigated through whitelisting of known and trusted nodes and requiring the node to connect to at least one of the whitelist nodes. Like the 51% attack, likelihood can be reduced by requiring multiple confirmations and manual verification of transaction finality using the blockchain explorer.

7.2.3 Denial of Service Attack

Several denial-of-service (DoS) attacks (**figure 7.3**) can be carried out against a blockchain network. This type of attack impacts the ability of the blockchain to process blocks, execute smart contracts or enable users to interact with the blockchain. However, due to the decentralized nature of blockchain, a distributed denial of service (DDoS) can reduce network activity only to a certain level. Based on the current execution speed of the network, the impact of a DDoS attack may be limited as well.

Anti-Pattern

Transaction flooding inundates the network with transactions to increase the size of the queue for transactions waiting to be added to blocks. This may reduce the difficulty level of the PoW problem, enabling the attackers to concurrently pursue a 51% attack more efficiently.

[49] Raj, K.; "What can Blockchain Developers Learn From Eclipse Attacks in a Bitcoin Network," Packt, 2019, https://hub.packtpub.com/what-can-blockchain-developers-learn-from-eclipse-attacks-in-a-bitcoin-network-koshik-raj/

In an artificial difficulty-increase exploit, the attacker contributes a significant amount of hash power to the network, temporarily increasing the PoW blockchain network hash rate, thereby requiring the network to push up the PoW difficulty to reduce the increased resources.

In a block forger DoS attack, the attacker targets a traditional DoS attack against the next block creator on a PoS consensus blockchain, preventing the block creator from accessing the network and preventing the block from being added to the chain.

In a permissioned blockchain, users only connect to the permissioned blockchain network only after they have been authenticated by one or more membership service providers (MSPs). In an MSP DoS, the attacker performs a DoS attack against the MSPs, thereby denying users access to the blockchain.

Figure 7.3—DoS (INL) Attack OWASP Mapping	
A4:2017-XML External Entities (XXE)	Flooding the node peers or network with phantom transactions increases transaction queue size, requiring an automated downward adjustment to the PoW problem difficulty level, preventing block creation or the MSP from handling user access, which allows attackers to launch other exploits.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

Mitigation

The mitigation methods vary, depending on the type of DoS, and can include:

- Wait out the attack or intentionally create blocks to clear flooded transactions from the queue.
- Set a difficulty-increase interval to minimize attack impact.
- Use a large pool of block creators (e.g., stakers).
- Randomize the selection of PoS block creator.
- Implement traditional DDoS protection for nodes.
- Leverage AWS or Azure to manage blockchain nodes.

7.2.4 Sybil Attack

Unlike an eclipse attack, which targets a specific node, a sybil attack focuses on the whole blockchain network. The sybil attack aims to corrupt a peer-to-peer network by forming several fake identities (including nodes on a blockchain) that participate and manipulate transactions. A large-scale sybil attack can facilitate a 51% attack or potentially enable an attacker to supersede honest nodes in a blockchain network. These attack consequences can ultimately result in nodes refusing to transmit or receive blocks, thus obstructing network operations. However, several types of consensus methods can make such a sybil attack impractical, given the compute processing power involved.[50]

Anti-Pattern

The attackers establish several fake nodes that appear to be genuine to their peers (**figure 7.4**). These fake nodes take part in corrupting the network to validate unauthorized transactions and alter valid transactions. The attackers can use several devices, virtual machines or Internet Protocol (IP) addresses as fake nodes for the attack to outvote legitimate

[50] Binance Academy, "Sybil Attacks Explained," https://academy.binance.com/security/sybil-attacks-explained

nodes to favor malicious transactions. A sybil attack requires amassing a very high amount of hash power or collusion with at least one-third of the stakers (i.e., miners, block creators).

Figure 7.4—Sybil Attack OWASP Mapping	
A4:2017-XML External Entities (XXE)	Blockchain nodes rely on messages communicated to neighboring peers. By masquerading as legitimate peers, malicious peers can provide inaccurate and false transaction information.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

Mitigation

The best mitigation for a sybil attack is to require the bad actors to expend a significant amount of resources that may not be feasible for them to attain, such as requiring participating nodes to spend on hash power, stake a reasonably large number of tokens per node or otherwise provide proof that the user has expended a resource that is deemed valuable to generate new blocks or transactions.

7.2.5 Border Gateway Protocol (BGP) Hijacking or Routing Attack

BGP hijacking, also known as a routing attack, is an exploit against BGP, where the Internet service provider (ISP) makes false announcements over the routing system to divert traffic. This attack typically requires a combination of partitioning the networks and then diverting traffic to delay transaction processing.

Anti-Pattern

In a BGP hijacking (**figure 7.5**), the attacker advertises short routes between two network segments containing nodes of a blockchain network, allowing traffic between those nodes to flow through the attacker and enabling the attacker to isolate the two parts of the network and delay transaction processing by miners. The attack reduces miner revenue, renders the network much more susceptible to double spending, and prevents merchants, exchanges and other large entities from performing transactions.

Figure 7.5—BGP Hijacking or Routing Attack OWASP Mapping	
A4:2017-XML External Entities (XXE)	Blockchain nodes rely on messages communicated to neighboring peers. By masquerading as legitimate peers, malicious peers can block transactions or provide inaccurate and false transaction information.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

Mitigation

Combatting BGP hijacking requires a community approach using multiple techniques:

- **Multihomed nodes**—If a node has Internet connections to two different segments, it is more difficult for an attacker to find a way to split the network.
- **Intelligent neighbor selection**—The more nodes that connect to nodes in different segments, the more communications the attacker needs to control.

- **Known route selection**—Uses known and trusted routes to communicate with nodes in other network segments.
- **Network statistics monitoring**—An attacker's rerouting and monitoring are likely to significantly increase network latency. Monitoring for this may detect ongoing attacks.
- **Encrypted authenticated communications**—Encryption and authentication ensure that an attacker cannot monitor and change the communications occurring between nodes.

7.3 Node Level

Nodes support consensus on the blockchain network and facilitate end-user transactions, such as sending and receiving cryptocurrencies as payment for goods and services. Node-level attacks primarily target nodes to divert the node resources to mine for the attacker's benefit.

7.3.1 Cryptojacking Attack

A cryptojacking attack is an emerging attack that leverages malware to install PoW mining code on user devices to steal the device's computing resources, such as CPU or GPU, to mine for cryptocurrencies for the malware owner. In a cryptojacking attack on Coinhive in 2017, the Coinhive cryptomining services did not account for coding errors that allowed attackers to take illicit control of websites and routers to aid illegal cryptomining operations.[51]

Anti-Pattern

Cryptojacking malware is typically delivered through phishing emails and malicious websites and ads. Once successfully loaded, the malware installs cryptomining code onto the user computer to stealthily mine for cryptocurrencies (**figure 7.6**).

Figure 7.6—Cryptojacking Attack OWASP Mapping	
A4:2017-XML External Entities (XXE)	Blockchain nodes rely on messages communicated to neighboring peers. By masquerading as legitimate peers, malicious peers can block transactions or provide inaccurate and false transaction information.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

Mitigation

Traditional user awareness and endpoint malware management minimize the impact of cryptomining. Specifically, enterprises can consider the following:

- Implement security awareness training for organizational users with a focus on phishing prevention.
- Install ad-blocking or anti-cryptomining extensions on web browsers.
- Use only trusted browser extensions and add-ins.
- Use endpoint protection.
- Use web filtering tools.

[51] Segura, J.; "Cryptojacking in the Post-Coinhive Era," Malwarebytes Labs, 2 May 2019, https://blog.malwarebytes.com/cybercrime/2019/05/cryptojacking-in-the-post-coinhive-era/

7.3.2 Remote Manager Exploit

PoW mining software provides the capability for end users to remotely monitor the performance of the mining nodes. In most cases, an unpatched mining software or misconfigured remote manager (e.g., no password and incorrect port used) is the primary reason for the success of the exploit. This was observed with the Claymore Dual GPU miner exploit, in which the remote management interface was vulnerable to an unauthenticated format string vulnerability and an authenticated directory traversal vulnerability was exploited by issuing a specially crafted request, which allowed a remote attacker to read/write memory and arbitrary files, including updating and altering mining wallet information. See **figure 7.7**.

Anti-Pattern

The attacker leverages open-source intelligence (OSINT) tools, such as Shodan,[52] to look for patterns of vulnerable miners. When identified, the attacker attempts to send requests to the vulnerable node. If successfully sent, the attacker sends an encoded configuration update and reboot request to manipulate the node to mine for the attacker's wallet address.

Figure 7.7—Remote Manager Exploit OWASP Mapping	
A6:2017-Security Misconfiguration	The attackers exploit misconfiguration of a miner remote manager to reconfigure mining details, such as mining payout address and remote password, and take over mining operations.
A1:2017-Injection	Malicious scripts are often delivered via phishing emails or injected into malicious ads as encoded hexadecimals to obfuscate underlying instructions or scripts.

Mitigation

Possible methods to mitigate remote manager exploits vary with the mining software design and include the following:

- Avoid a remote manager by disabling port forwarding on the remote manager unless justifiable for business reasons.
- Download mining software from authoritative sources, such as the pool provider.
- Set up the mining node to run only as standard users to minimize lateral movements across the network.
- Use endpoint malware protection and provide timed exclusion for the mining software (e.g., must be manually renewed every 30 days to limit loss).
- Set scan exclusion for only the mining executable and not the mining directory.

7.4 Smart-Contract Level

One of the most compelling propositions for blockchain is the use of smart contracts on second-generation and later blockchains. In addition to the elimination of middlemen, the business logic embedded in the smart contracts enables the smart contracts to execute automatically, consistently, trustlessly and impartially. However, unlike traditional software engineering, standardized best practices for smart contracts are still evolving, and smart contracts cannot be easily updated or patched due to the immutability design of the blockchain.

[52] Shodan, https://www.shodan.io

Although blockchain properties and chosen smart-contract languages can vary significantly among different blockchain designs, security considerations and anti-patterns are generally similar. Accordingly, the attacks and mitigations discussed in this section can be broadly applied across different blockchain designs.

General mitigation methods against smart-contract attacks include:

- Keep the design simple and not overly complex.
- Use standardized and tested libraries and interfaces.
- Use automated testing that employs vulnerability assessment tools where possible to augment testing completeness.
- Conduct internal or peer code reviews.
- Conduct formal code reviews handled by third-party auditors.
- Test smart contracts thoroughly from alpha testnet through prior to production.
- Retest when new attack vectors are discovered.
- Refactor smart contracts for updates of versions, tools or libraries used as soon as possible.
- Leverage bug bounties program to crowd-source other testing perspectives.
- Understand the pitfalls of the properties of the selected blockchain.

7.4.1 Access Control

Every smart contract has at least one owner, which is typically the address that deployed the smart contract. An access-control attack attempts to seize ownership of a smart contract from its rightful owner (**figure 7.8**). When the owner address is identified, the attacker creates an attack smart contract from which it calls the vulnerable smart-contract function to take over ownership.

Anti-Pattern

The attacker examines the deployed smart contract for incorrect usage or lack of a constructor to initialize ownership (**figure 7.8**), or failure-to-check-for-ownership-prior-to-execute key functions.

Figure 7.8—Incorrect Construction

```
1   pragma solidity ^0.4.21;
2
3   contract OwnerWallet {
4           address public owner;
5
6           function initWallet() public {
7                   owner = msg.sender;
8           }
9
10          // Fallback. Collect ether.
11          function () payable {}
12
13          function withdraw() public {
14                  msg.sender.transfer(this.balance);
15          }
16  }
```

Figure 7.9—Access Control OWASP Mapping

A5:2017-Broken Access Control	Due to the transparency of smart contracts, the attackers can examine deployed codes and exploit against restrictions on authenticated users that are often not properly enforced, such as executing admin-reserved functions and fund transfer.

Mitigation

Ensure that a smart contract is properly initialized to maintain contract ownership (**figure 7.10**). Where applicable, consider allowing multiple administrators in the deployment of the smart contract. Verify that the function caller is the contract owner before the execution of functions intended for the contract owner. Users who interact with the smart contracts should be limited to functions intended for those users.

Figure 7.10—Corrected Constructor

```
1   pragma solidity ^0.4.21;
2
3   contract OwnerWallet {
4       address public owner;
5
6       //constructor to initialize ownership
7       function OwnerWallet() public {
8           owner = msg.sender;
9       }
10
11      // Fallback. Collect ether.
12      function () payable {}
13
14      function withdraw() public {
15          require(msg.sender == owner);
16          msg.sender.transfer(this.balance);
17      }
18  }
```

7.4.2 Default Visibility

A smart contract typically has multiple functions that can be externally accessed (e.g., called by an external address or another smart contract) or accessed from other functions or derived contracts from the contract itself. Improper coding or misuse of visibility modifiers exposes certain functions to manipulation by other contracts when these functions are not intended to be accessed. (See **figures 7.11** and **7.12**.)

Anti-Pattern

By default, Ethereum functions are explicitly public unless stated differently by the developers.

Figure 7.11—Incorrect Default Visibility

```
1   pragma solidity ^0.4.21;
2
3   contract HashForEther {
4
5           function withdrawWinnings() {
6                   //Winner if the last 8 hex characters od the address are 0
7                   require(uint32(msg.sender) == 0);
8                   _sendWinnings();
9           }
10
11          function _sendWinnings() {
12                  msg.sender.transfer(this.balance);
13          }
14  }
```

In the example in **figure 7.11**, another contract can bypass the function withdrawWinnings() completely by calling the function sendWinnings(), because its default visibility is not specified and thus is public.

Figure 7.12—Default Visibility OWASP Mapping

A6:2017-Security Misconfiguration	Attackers exploit incorrectly defined public function calls that are intended for internal calls.
A3:2017-Sensitive Data Exposure	Rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks or front-running transactions, or steal clear text data off the transaction queue or hidden or transacted from blockchain data.

Mitigation

Explicitly state the visibility identifier for the function (**figure 7.13**). Monitor function calls to smart contracts using applicable blockchain explorers (e.g., etherscan.io, Bitcoin explorer). Although monitoring itself is not a mitigation, it can provide useful insights to functions being tested by the attackers for possible weaknesses.

Figure 7.13—Corrected Default Visibility

```
1   pragma solidity ^0.4.21;
2
3   contract HashForEther {
4
5           function withdrawWinnings() public {
6                   //Winner if the last 8 hex characters od the address are 0
7                   require(uint32(msg.sender) == 0);
8                   _sendWinnings();
9           }
10
11          function _sendWinnings() private internal {
12                  msg.sender.transfer(this.balance);
13          }
14  }
```

Where possible, make visibility as limited as possible and consistent with the intended use of the functions being called.

- **Public**—Can be accessible internally or to other external smart contracts.
- **External**—Can be called only from other external contracts and not internally.
- **Private**—Can be accessed only from this contract.
- **Internal**—Can be accessed only from this contract and any other contracts derived from this contract. A typical use may be a derived-owner-control smart contract to transfer funds from the main contract into the owner's wallet.

7.4.3 Reentrancy

Reentrancy is a classic attack that takes over control flow of a smart contract and manipulates the data to prevent the correct updating of contract state variables. Reentrancy attacks have accounted for the loss of hundreds of millions of dollars of cryptocurrencies since they were first executed in 2016.

Anti-Pattern

Reentrancy is most exploitable when a smart contract calls an external function. Reentrancy can exist on a single function, allowing that function to be called repeatedly, before the first invocation of the function is finished to correctly set any internal state variables. (See **figures 7.14**[53] and **7.15**.)

[53] Consensys, "Ethereum Smart Contract Best Practices," https://consensys.github.io/smart-contract-best-practices/known_attacks/

Figure 7.14—Single-Function Reentrancy

```
1   // INSECURE
2   mapping (address => uint) private userBalances;
3
4   function withdrawBalance() public {
5           uint amountToWithdraw = userBalances[msg.sender];
6           require(msg.sender.call.value(amountToWithdraw)());
7           // At this point, the caller's code is executed and can call withdrawBalance() again
8           userBalances[msg.sender] = 0;
9   }
```

Reentrancy, such as cross-function reentrancy, can also be executed on two or more functions within the same smart contract, or derived smart contract provided the functions share the same state.

Figure 7.15—Reentrancy OWASP Mapping

A7:2017-Cross-Site Scripting (XSS)	Reentrancy flaws occur whenever an application includes untrusted data in a smart-contract call without proper validation or escaping, or updates contract state variables with user-supplied data using an external call.
A1:2017-Injection	Injection flaws, such as altered configuration settings, occur when untrusted data are sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Mitigation

Observe the smart-contract design rule of checks-effects-interactions. For the checks, verify that the input is acceptable. If not, then either fail early or return a false value. This check ensures that the entire transaction reverts, including all functions in all contracts that were involved. For the effects portion, update the contract to a new valid state assuming that the interactions will be successful. This protects the contract from reentrance and race conditions. For the interactions, limit interactions with any untrusted contract as much as practically possible, and when there is an interaction, check the result. (See **figure 7.16**.)

Another general rule to mitigating reentrancy is to finish all internal work, such as state changes, first and then call the external function.

Figure 7.16—Corrected Single-Function Reentrancy

```
1   mapping (address => uint) private userBalances;
2
3   function withdrawBalance() public {
4           uint amountToWithdraw = userBalances[msg.sender];
5           userBalances[msg.sender] = 0;
6           require(msg.sender.call.value(amountToWithdraw)());
7           // The user's balance is already 0 so future invocations won't withdraw anything
8   }
```

Keep in mind that a design to safeguard single-function reentrancy may not work on cross-function reentrancy. One may have to consider safeguards, such as the use of MUTEX to lock and protect against recurring calls, or to use transfer() and send() functions versus using call() where a gas limit minimizes any recurring function calls.

7.4.4 Integer Overflow/Underflow

An overflow or underflow condition occurs when an operation is performed that requires a fixed-size variable to store a number (or piece of data) that is outside the range of the data type of the variable. Attackers rely on overflow and underflow manipulations to bypass conditional checks and alter mathematical calculations, such as payment, to their benefit. These types of conditions have been a concern for some time, and there have been several incidents based on integer overflow or underflow conditions. For example, in 1996, the European Space Agency rocket Ariane 5 malfunctioned and self-destructed because of an integer overflow condition associated with the rocket software, which had been previously coded for a predecessor rocket.[54]

Anti-Pattern

Because public functions and variables are accessible to other smart contracts, they can be manipulated during external calls and data inputs to induce an overflow or underflow condition. When the condition is overflow, an unsigned integer is incremented above its maximum value and returns to zero. When the condition is underflow, an unsigned integer is decremented below zero and returns to the maximum value. The maximum value varies with the blockchain design. For example, the Ethereum maximum unsigned integer is a 256-bit integer equal to $2^{256} - 1$. Not limited to just storage variables, overflow and underflow conditions may also be triggered through calculations, particularly when calculations are performed using loops that are nondeterministic and the loop ending value depends on a specific variable that changes from run to run. (See **figure 7.17**).

Figure 7.17—Integer Overflow/Underflow OWASP Mapping

A9:2017-Using Components with Known Vulnerabilities	Components, such as libraries, frameworks and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.

Mitigation

Use either safe math implementation or predefined safe math libraries from trusted sources, such as OpenZeppelin SafeMath.sol for Ethereum blockchain. Perform input validation to ensure that user inputs have the correct data type and are within the appropriate lower- and upper-limit values. Plan for any post-input calculations that may result in an overflow or underflow condition. Check storage and calculated variables for a valid condition.

7.4.5 Timestamp Manipulation

This attack targets the blockchain timestamp function—blockchains rely on the node timestamp to establish the difficulty and ordering of the blocks. Timestamp manipulation enables the attacker to artificially deflate the difficulty by reporting a higher level of effort to mine the block than performed. Decentralized applications that derive the timestamp from the blockchain are also vulnerable to this attack, because the same vector can manipulate function conditions to the attacker's benefit. Verge Coin suffered this attack in April 2018 when a bug allowed the manipulation of block mining timestamps to create illegitimate coins.[55]

[54] Lions, J.L.; "Ariane 5 Flight 501 Failure," 19 July 1996, https://web.archive.org/web/20000815230639/www.esrin.esa.it/htdocs/tidc/Press/Press96/ariane5rep.html

[55] Suberg, W.; "Cryptocurrency Verge Responds to Hacking Claims by Launching 'Accidental Hard Fork,'" 5 April 2018, https://cointelegraph.com/news/cryptocurrency-verge-responds-to-hacking-claims-by-launching-accidental-hard-fork

Anti-Pattern

Block creators or miners can set their time to any period in the future, and the mined block may still be valid, provided the timestamp is within the required threshold. Depending on the blockchain design, the timestamp can be off from true time by between 30 seconds and several minutes in the future and may still be accepted as valid (see **figure 7.18**).

Figure 7.18—Timestamp Manipulation OWASP Mapping	
A10:2017-Insufficient Logging and Monitoring	Blockchain design relies on timestamping to accurately order and resolve conflict for transactions. Insufficient logging and monitoring allow attackers to manipulate system difficulty and transaction ordering and to tamper with, extract or alter transaction data.

Mitigation

Avoid assigning a block timestamp to a variable in a smart-contract code and instead use the block number. Decentralized applications should not rely on a blockchain timestamp as advertised but instead should rely on multiple time oracles as authoritative sources. Verify that nodes are required to synchronize date and time to a universal time provider, such as time.nist.gov, and to ensure that the mined block meets the required threshold. Any blocks that are mined outside of the tolerances should be discarded.

7.4.6 Bad Randomness

Blockchain is a deterministic system; thus it does not have a secure source of entropy from which one can create random output. Any mechanism that generates pseudorandom numbers on the blockchain must be thoroughly scrutinized for possible faulty implementation.

Anti-Pattern

The use of private variables to hold salts or hidden values does not work in a blockchain environment because state variables are set via transactions at some point in time and, therefore, are visible on the blockchain. Even when they are combined with block-specific variables, such as timestamp and block number, the algorithm is still known; therefore, the randomness can also be reconstructed (see **figure 7.19**).

Figure 7.19—Bad Randomness OWASP Mapping	
A3:2017-Sensitive Data Exposure	Rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks or front-running transactions, or steal clear text data off the transaction queue or hidden or transacted from blockchain data.
A9:2017-Using Components with Known Vulnerabilities	Pseudonumber generation on blockchains is predictable; thus, if a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.

Mitigation

The best random generators are those derived from external random oracles or services, such as random.org or provables.xyz. Care should be taken to ensure that the random value is safeguarded and used when obtained from the oracle. In certain blockchain designs, using hash from a future block may be a reasonable seed for a pseudorandom generator.

7.4.7 Front Running

Every blockchain transaction pays a gas fee as a mechanism to deter spamming of the blockchain network and to pay for the cost of keeping the network secure (e.g., as compensation to the miners for validating and maintaining transactions). Mining nodes pool transactions and form them into blocks. Typically, the miner selects transactions to be pooled based on the gas fees and the address nonce. In a front-running attack, the attacker duplicates the unmined transaction and pays higher gas fees to preempt the original transaction. In 2018, there were several examples of front running of blockchain transactions on the Ethereum blockchain network.[56]

Anti-Pattern

In a front-running attack, the attacker watches the transaction pool for transactions that may contain the solution to a specific problem (e.g., correct answer to a game puzzle in a DApp), and then modifies the solver's permissions or change state in a contract to the detriment of the solver, causing a delay or lag via DDoS. Concurrently, the attacker can get the data from this transaction and create a transaction for the attacker with a higher gas fee, so the transaction is included in a block before the original solution (see **figure 7.20**).

Figure 7.20—Front-Running OWASP Mapping	
A3:2017-Sensitive Data Exposure	Rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks or front-running transactions, or steal clear text data off the transaction queue or hidden or transacted from blockchain data.

Mitigation

Observe the importance of transaction ordering or time in the design of a DApp. Where possible, use a commit-reveal approach to mitigate an observer's ability to front-run the transaction. In the commit phase, the parties submit their cryptographically protected secrets to the smart contract. In the reveal phase, the parties announce cleartext seeds, and the smart contract verifies that they are correct. Last, stipulate an acceptable range of gas fees on the transaction, thereby negating front running.

7.4.8 Denial of Service

In a denial-of-service (DoS) attack, the attacker manipulates one or more conditions, causing the smart contract to fail and essentially disabling the smart contract. Because contracts are immutable, failed contracts represent significant risk, which was the case with the 2019 Parity Multi-Signature Library hack, where a multisignature library was set to self-destruct by an attacker, resulting in the complete lockup of funds used by smart contracts that required the use of the same library.

Anti-Pattern

Several attack vectors relating to DoS follow (**figure 7.21**):

- **Blocking access to the contract owner account**—Preventing the owner from executing owner-designated functions.
- **Locked functions**—Manipulating through a fallback function that causes the transaction to revert before completing the loop.

[56] Czernik, M.; "On Blockchain Frontrunning," Medium, 9 February 2018, https://medium.com/@matt.czernik/on-blockchain-frontrunning-part-i-cut-the-line-or-make-a-new-one-b33850663b55

- **Exceeding gas limit**—Causing the transaction to run out of gas, thereby reverting the transaction. This may also exist with transactions that run over unbounded operations.

Figure 7.21—Denial of Service (SmartContract) OWASP Mapping	
A4:2017-XML External Entities (XXE)	The application accepts messages directly through interactions with its peers, unknowingly from untrusted sources, which provides inaccurate and false transaction information.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

Mitigation

The best mitigation for DoS is the use of expert code review and user testing in a test network with performance evaluation or gas optimization.

7.4.9 Unchecked Return/Unhandled Exceptions

Results from function calls in a smart contract must be verified before additional processing. This is more significant when the results derived from a call to an external function leads to a failure to verify the low-level function state after the call, which may result in incorrect variable states.

Anti-Pattern

Low-level functions can vary with different blockchain designs. On Ethereum, examples of low-level functions are call(), callcode(), delegatecall() and send(), and similar functions such as invoke() on Hyperledger Fabric chain code. These unique functions typically return a Boolean value rather than executing a transaction revert, inherent to a blockchain design. Failure to trap or catch for the return values may place state variables out of sync with the intended programming design and produce unpredictable and undesirable behavior (see **figure 7.22**).

Figure 7.22—Unchecked Return/Unhandled Exceptions OWASP Mapping	
A6:2017-Security Misconfiguration	Attackers often attempt to exploit unpatched flaws or access default accounts, unused pages, unprotected files and directories, etc. to gain unauthorized access or knowledge of the system.

Mitigation

When using low-level functions, always check the return value to confirm the success of the execution. If not successful, force an exception so that the transaction immediately rolls back.

7.4.10 Missing Input Validation

Behaviors of smart contracts are deterministic, based on the range of the data inputs. When input validation is not performed, values fall outside of the expected range and can lead to many issues, including buffer overflow, short-address attack and loss of access control.[57]

[57] Kaiser, T.; "Lack of Consensus: Programming Patterns That Prevent Consensus in Your Hyperledger Smart Contracts," Medium, 19 July 2018, https://medium.com/chainsecurity/lack-of-consensus-programming-patterns-that-prevent-consensus-in-your-hyperledger-smart-contracts-52134ca74bc0

Anti-Pattern

Integer values are not bound to known minimum and maximum values. Bytes are not verified to meet the required length. Solidity addresses do not contain the required number of characters, resulting in padding with leading zeros (see **figure 7.23**).

Figure 7.23—Missing Input Validation OWASP Mapping	
A6:2017-Security Misconfiguration	Attackers often attempt to exploit the lack of input validation to pass in out-of-bound or incorrect data types in attempts to induce application errors to escalate their access.

Mitigation

Ensure that data types are expected before incorporating their use into the smart-contract functions.

7.4.11 Read After Write/Bad Handling of Asynchronous Operation

A blockchain network such as Ethereum supports the sequential execution of transactions on smart contracts with a consensus mechanism.[58] In a sequential execution, the requests to the smart-contract invocations are ordered by the consensus method. Then the smart contracts are executed in the same order on all the nodes. This method has many performance limitations and drawbacks in blockchain-based applications.

Anti-Pattern

An attacker can craft a smart contract that requires a significant amount of time to complete execution, causing network delay. A smart-contract function may execute a write function to update a state variable but read the updated value without having achieved an asynchronous update.

Mitigation

Ensure that global state variables are updated prior to use.

7.4.12 Arbitrage Attack

As blockchains and smart contracts become more mainstream, new ecosystems, such as decentralized finance (DeFi), emerge with new use cases, such as DeFi lending, decentralized exchanges (DEXs) and flash loans. DeFi lending allows participants to obtain fiat loans using cryptoassets as collateral. A DEX enables traders to swap cryptoassets, without a central authority's permission, by creating a liquidity pool for specific tokens and having the orders executed algorithmically on the Ethereum blockchain. Flash loans allow traders to take out uncollateralized loans to increase the payout of a singular trade, provided the loan is paid back in the same transaction in which it was taken out. An arbitrage attack focuses on the execution of a timed series of transactions across one or more DEXs, to drive up the price of a specific asset, from which the attacker can profit at the expense of one or more participating service providers and take advantage of the disparity in the price of the cryptoasset required to settle the transactions. The first two known cases of arbitrage attack took place against a bZx decentralized finance project, costing the organization almost US$1 million.

[58] Kaiser, T.; "Chaincode Scanner: Automated Security Analysis of Chaincode," Chainsecurity, https://static.sched.com/hosted_files/hgf18/18/GlobalForum-tobias-kaiser.pdf

Anti-Pattern

For the attack to be successful, one or more of the underlying service providers must rely on a single price feed from an oracle that generally operates as an onchain decentralized oracle. Targeting the onchain decentralized oracle is important because its pricing information is likely to be lower than the true price, thus causing the service provider to obtain insufficient collateral for the pending transaction (see **figure 7.24**).

Another observation entails monitoring the dependencies of the manipulation across the participating third-party providers and understanding what spot price truly means for the involved parties. For example, an accurate rate for a DEX means that a trade can be made using that rate, but an accurate rate for a DeFi project means that it is close to or equal to the fair market value of the asset.

Figure 7.24—Arbitrage Attack OWASP Mapping	
A9:2017-Using Components with Known Vulnerabilities	DEX and DeFi rely on the transparency of audited smart contracts. Accordingly, any exposed vulnerabilities can also be exploited. In the case of bZx, possible slippage vulnerabilities were highlighted by a researcher in the month before the attack.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the lack of monitoring and timely response to achieve their goals without being detected. For bZx, they targeted the initial attack on Valentine's Day, a Friday night, and during ETHDenver, when the core bZx team was out.

Mitigation

Using multiple data feeds from multiple oracle sources for decision making can significantly reduce risk. Place sanity bounds on the oracle returned value and fails if the value exceeds a certain minimum or maximum threshold. Understand the implications of introducing a dependency on a third-party project. Consider whether the project has been audited and whether the project specifications and threat model align with its intended use.

7.5 User Level

Users interact with a blockchain application through a front end using a software wallet, such as MetaMask or MyEtherWallet, which holds the user private keys in one or more wallet addresses. The recovery phrases, or seed words, and key storage files for these wallets are prime targets in phishing attempts to seize ownership of the user addresses. The selection of the wallet should be based on usage frequency, ease of use, amount of assets to be protected and appetite for loss of assets. Most wallets are noncustodial, thus the liability for loss likely resides with the users; however, some wallet designers, such as MetaMask, may release the core code review and audit for possible weaknesses.

7.5.1 Fake Cryptocurrency Exchange, Wallet, Airdrop and Hard Fork Scams

Since decrypting stolen encrypted key storage files requires significant time and resources, this type of attack focuses on human greed and emotions to get the users to volunteer their private keys. These attacks include using fake cryptocurrency exchange, wallet download or giveaways due to forking or airdrops.

Anti-Pattern

These scams entice victims through sophisticated websites with quality roadmaps, white papers, and a seemingly impressive management team. They may also coincide with legitimate planned blockchain hard forks and extensive promotions of airdrops and incentives using fake Twitter accounts.

In most cases, the scams require the users to adopt the scammer's online exchange or wallet or to download the scammer's fake wallet to claim airdrops and to provide existing user private keys required for snapshot or proof of ownership. After the private keys are obtained, the scammer transfers the user funds into a wallet controlled by the scammer (see **figure 7.25**).

Figure 7.25—Fake Cryptocurrency Components OWASP Mapping	
A3:2017-Sensitive Data Exposure	Most airdrops and hard fork scams focus on stealing user private keys by getting the user to divulge the keys in exchange for the drops, through subversive approaches, including use of malware.
A10:2017-Insufficient Logging and Monitoring	The attackers rely on the greed of recipients and the lack of monitoring of recipient infrastructure to exfiltrate the private keys.

Mitigation

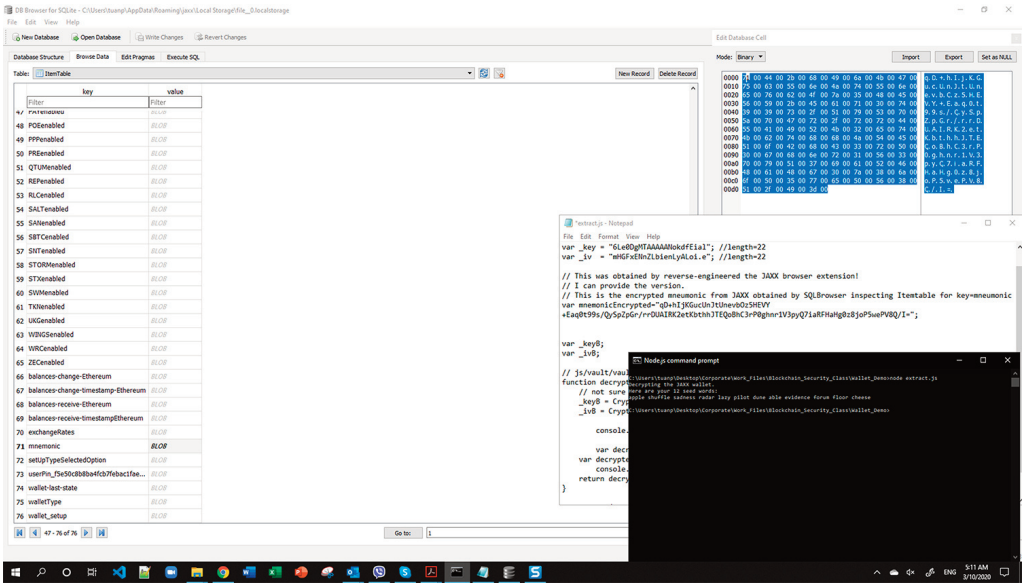
Research thoroughly for all downloads, airdrops and hard forks. In many cases, legitimate airdrops and hard forks require little action from the user side. Be careful when private keys are being asked for. If a download is requested, make sure it does not contain malware by scanning with an antimalware product. Where applicable, download from authoritative sources, such as Google Play, Apple Store or the developer's GitHub repository, and verify the file hash before installation. When infeasible, download into a sandbox and check installation and runtime behavior with Wireshark or some other open-sourced network protocol analyzer. If actual testing is the last resort, segregate from the environment that is holding the true wallet and test with a small amount of cryptocurrency. Activate multifactor authentication on user accounts. Although not 100 percent secure, multifactor authentication is much harder to bypass and is effective in protecting user funds, even if login credentials are phished.

7.5.2 Wallet Exploits

The noncustodial wallet may be vulnerable due to design flaws, as in the case of Jaxx wallet, where the encryption algorithm has been broken; misconfiguration by the user (e.g., forgetting to encrypt the key storage files); or key storage files that were left locally after the software was removed.

In the case of Jaxx wallet, the key storage mnemonic can be extracted, as shown in **figure 7.26**, and the original seed words can then be obtained using a decryptor.

Figure 7.26—Decrypting Seed Words in Jaxx Wallet



Wallet developers do not delete key storage files during the wallet software uninstallation. This rule is designed to prevent accidental loss of user private key information (see **figure 7.26**).

These files are not visible to the normal users so if they are unmanaged, these files can be stolen from compromised user devices for attempted decryption (**figure 7.27**).

Figure 7.27—Wallet Exploits OWASP Mapping

A9:2017-Using Components with Known Vulnerabilities	Jaxx users are fearful of losing access to their funds if they upgrade the software, so they choose not to update the software to the later version–Jaxx Liberty. Accordingly, Jaxx storage files are vulnerable and can be exploited if a hacker gains access.

Mitigation

Search authority sources to ensure that the installed wallet version has no known vulnerability. Enable local storage encryption as soon as the wallet is installed. Check and back up the seed words or recovery phrases before sending or receiving funds. Test by sending and receiving small amounts at the start. Wait for confirmations to verify effectiveness before sending additional transactions. When a wallet is uninstalled, make sure to use a shredder tool to remove the key storage files.

7.5.3 SIM Swapping

Mobile phones are becoming increasingly popular as multifactor authentication devices for user login to web wallets, such as those provided by Coinbase and Binance.us. Attackers seek to gain access to such devices through SIM swapping, by relying on social engineering methods to trick mobile phone operators into issuing a new SIM card to the attackers. SIM swapping attacks require no participation from the victims and allow the attackers to bypass multifactors that rely on the mobile number. A well-known SIM swap attack took place in May 2018, in which a major cryptoinvestor was hacked for US$23.8 million in cryptocurrency.[59]

[59] Kapilkov, M.; "Crypto Investor Sues New York Teen for $71.4 Million in SIM-Swap Saga," 8 May 2020, https://cointelegraph.com/news/crypto-investor-sues-new-york-teen-for-714-million-in-sim-swap-saga

Anti-Pattern

Attackers attempt to run these attacks when the victims are occupied (e.g., in meetings or at the movies) or otherwise not accessible, such as inflight. The only indicator to the victims is the loss of mobile usage, which requires the victims to contact the operator or, in certain cases, to visit a store location.

Mitigation

Arrange with mobile phone providers to protect the account. Require additional layers of verification, such as being physically present with identification at the store to make the request. Do not reveal cryptoholding or personally identifying information on social media to avoid being targeted by cybercriminals. Do not use the mobile phone number for SMS as multifactors. Instead, use apps such as Google Authenticator or Authy to secure user accounts.

7.5.4 Dusting Attack

A dusting attack is a social engineering technique where by malicious attackers attempt to break the privacy of cryptocurrency holders by sending small amounts of coins or tokens to the user wallets.

Anti-Pattern

Attackers send small amounts of cryptocurrency to the user wallets and monitor the movement of user funds to subsequent addresses to identify the users or owners of the wallets.

Mitigation

Dusting attacks rely on a combined analysis of multiple addresses. If a dust fund is not moved by the user, the attackers cannot make the connections required to de-anonymize the wallets.

7.5.5 Privacy Considerations

Data stored on the blockchain are immutable and are visible to nodes and node operators. Accordingly, sensitive data must not be stored on public blockchains.

Regulations such as the General Data Protection Regulation 2016/679 (GDPR) and the California Consumer Privacy Act (CCPA) also stress the need for blockchain solutions to provide data protection and privacy for all individual users of the solutions. The nature of blockchain, including its distributed aspects, makes such requirements difficult, and much research is under way to derive GDPR- and CCPA-compliant blockchain solutions.

Enterprises must consider the tradeoff and plan for data accuracy and correction versus immutability of information. Hard forking of codes to address data issues on public blockchains is impractical due to a combination of challenges from governance, execution, cost of chain reorganization and other collateral risk that may arise from the event. Although private blockchains may have more flexibility in the management of hard forks using prebuilt transaction rollback mechanisms to correct faulty or incorrect transactions, rigid check-and-balance oversight must be in place to deter malicious tampering of previous legitimate transactions, particularly for blockchains that manage physical or financial assets.

Due to its inherent distributed nature, blockchain implementation must consider the rights of individuals to protect and erase their private information, particularly financial and health information. Accordingly, instead of using actual data, better privacy implementations should use cryptographic hash for evidence on the chain. Other privacy

implementations may be provided by the blockchain platform. For example, Hyperledger Fabric allows the use of channels for private information exchanges between selected members within a larger network, as noted in chapter 4. However, there are some instances when this may not be compliant. For example, a hash of personally identifiable information may not be complaint under GDPR. Readers are encouraged to research and confirm the compliance of a blockchain or smart-contract solution with legal counsel or relevant regulatory bodies.

Other implementation possibilities include the obfuscation of transaction data, additional safeguards to limit access control for the nodes and the participants, or the use of zero-knowledge proofs or succinct noninteractive arguments of knowledge (SNARKs). SNARKs offer the greatest possibility for safeguarding privacy because they programmatically verify hidden inputs known only to the user to derive publicly known output that affirms the user without revealing any other information.

Page intentionally left blank

Chapter 8
Digital Asset/Token Requirements

8.1 Introduction

Chapter 8	Digital Asset / Token Requirements
Description	This chapter provides an overview of cryptotokens and digital assets encompassing: • Definition of cryptotoken • Key features of cryptotokens • Cryptotoken use cases • Types of cryptotokens • Cryptotoken technology standards • How cryptotokens are issued • How cryptotokens are traded • Token economies • Security concerns • Regulatory considerations • Cryptotoken compliance
Key questions enterprises need to answer	• Why do we need to issue or interact with cryptotokens? What business purpose are we enabling? • Are the relevant tokens regulated as securities or not? Does it matter to us if they are? • How do the relevant cryptotoken regulatory regimes in the relevant jurisdictions impact us? • How are we managing risk, including security risk, fraud risk, fat finger risk, liquidity risk, blockchain risk and market/exchange rate risk? • How are we managing compliance requirements? • How are we managing wallets, private keys and/or token custody? • How are we integrating and aligning across procurement, treasury, accounting and tax?
Stakeholders	Stakeholders include, but are not limited to, the board of directors, executive management, corporate treasurers, business unit managers, IT managers/practitioners, assurance providers, risk management personnel, compliance personnel, procurement personnel, traders, broker/dealers, investors, regulators, and business or vendor partners.
Resources	• SEC Cyber Security Enforcement Actions: www.sec.gov/spotlight/cybersecurity-enforcement-actions • New York Department of Financial Services: www.dfs.ny.gov/apps_and_licensing/virtual_currency_businesses

Any framework related to blockchain technology implementation must address the rising usage and use cases for digital assets and tokens, which have their reasons for being inextricably linked to blockchain technology. In the context of a blockchain framework, there are two dominant reasons why digital assets or tokens may be relevant:

- **Use of tokens as part of the blockchain ecosystem**—Tokens can be important to the operations of a blockchain. The bitcoin cryptotoken is used as a mechanism to incentivize miners to carry out the complicated processing necessary to validate transactions on the blockchain. Likewise, many other public blockchains

leverage the token model to power their blockchain operations. Understanding the issuance of cryptotokens as part of a framework in this context is important.

- **Issuance of tokens to fund a new business or new platform**—In addition to enabling public blockchains to operate, tokens have been used to raise funds for the development or launch of new businesses or platforms. Although no longer focusing on the now widely discredited initial coin offering (ICO) model, enterprises around the world are listing security tokens on exchanges as a method of raising capital. It is important for any blockchain framework to offer a fundamental understanding of this evolution in capital raising.

Digital assets and tokens are an important component of the rapidly evolving landscape of blockchain technology. As multiple public and private blockchains evolve, and as more global enterprises not only launch blockchain platforms but also interoperate and integrate with other platforms, it is critical for users of this framework to be aware of these uses and the potential legal and regulatory risk associated with them. The broad definitions of tokens and risk are covered in more detail later.

8.2 Definition of Cryptotoken

The terms cryptotoken, cryptocurrency, altcoin, utility token, security token and digital asset are used—sometimes interchangeably—to describe a digital unit of value, utility or asset on a blockchain. These tokens may or may not be securities, or they may exist in a regulatory gray area. There are examples of tokens that possibly started out as securities and then became nonsecurities over time. Some tokens are natively digital, meaning there is no underlying offchain or real-world asset the tokens represent. Other tokens are tokenized offchain assets, in which an asset in the real world (e.g., stock certificate, piece of real estate, work of art or rights to IP royalty payments) is represented as a token on a blockchain. For this framework, cryptotoken is used as the catch-all term.

Bitcoin, first mined in 2009, is the original blockchain-based cryptotoken. Since then, tens of thousands of different cryptotokens have come into existence on multiple different blockchains. Many of these tokens trade freely, with holders able to trade these tokens for another token or a fiat currency based on floating market prices. The total current market capitalization of traded cryptotokens is around US$250 billion, with Bitcoin still by far the largest with a market cap of around US$170 billion (based on 5 May 2020 estimates on coinmarketcap.com).

Cryptotokens are used as an investable asset class, or a store of value, to make electronic payments, to access and use blockchain-based applications and services, for distributed governance and voting, to fund and create security for blockchains and blockchain-based apps, and to create tradable and fractionalizable digital securities and nonsecurity tangible and intangible assets. Like traditional assets, there are cryptotoken trading venues, indices and derivatives.

Some industries are already interacting with cryptotokens. For example, some ecommerce sites accept bitcoin for payment. A small amount of real estate has been tokenized. A few security tokens have been issued on blockchains. Less regulated financial services enterprises, such as hedge funds and family offices, are investing in cryptotokens. Supporting services, such as custodians and data vendors, are starting to service the cryptotoken needs of those businesses.

As more business activity moves to blockchains and as more assets are tokenized on blockchains, more businesses and individuals will buy, hold, sell and use cryptotokens as an essential part of their economic and business activity. They will need to do so securely, within risk tolerances and while meeting compliance requirements.

8.3 Key Features of Cryptotokens

Most cryptotokens share the following features, although there are exceptions, limitations and many additional potential features.

- **Fungibility**—Whether a token is fungible is one of its most fundamental characteristics. Fungibility is the concept that one unit is completely interchangeable with any other unit of the same thing. Using the US dollar as an example, a one-dollar bill is completely interchangeable with any other one-dollar bill—they are exactly the same in value and utility. Many tokens also work this way, with each token being completely interchangeable with any other of the same token. Each paper US dollar has a unique serial number, but this uniqueness is rarely considered—no one pays much attention to the serial number, and there is no differentiation in value, dollar to dollar. But in edge cases, such as a bank robbery, the serial numbers can be used to track criminals and criminal activity. Tokens also have unique identifiers—even fungible tokens. These can be used to track individual tokens across trades and to grant or restrict specific utility to a token. These unique identifiers can also be used to give certain otherwise fungible tokens specific utility. There are also nonfungible tokens (NFTs),[60] in which the uniqueness of each token is the point—for example, a token that represents a real estate title or ownership of a work of art. There are also various hybrid models that combine fungible and nonfungible characteristics. An example is a token representing a general admission concert ticket for a particular tour date. This token is not interchangeable with a token for a different date, but it is fungible with any other general admission token for the same date.
- **Tradability**—Tokens are typically held in digital wallets on or connected to a blockchain or multiple blockchains. From this wallet, tokens can be sent to another wallet, smart contract or blockchain address, often in exchange for something of value (another token, a fiat currency or a service). Because of how blockchain transactions work, these trades settle with each block, creating a highly efficient medium of exchange. There are transaction and/or other processing fees associated with blockchains and sometimes wallets. Although settlement is nearly real time, settlement finality can be difficult to ascertain to the degree required for some governance, risk, compliance or regulatory purposes on some blockchains. This is due to the low but nonzero risk that the specific chain of blocks that includes the trade can be superseded by another chain that does not include the settled trade. In these blockchains, settlement finality is probabilistic.[61] In addition, trading of certain cryptotokens may be restricted—either programmatically or by legal and regulatory regimes—by the token itself, the wallet or relevant laws and regulations.
- **Fractionalization**—Many cryptotokens can be subdivided—often to many decimal points. The US dollar can be fractionalized to two decimals: cents. That unit of value cannot be further divided. Bitcoin can be fractionalized to eight decimals. The smallest unit (0.00000001) of a bitcoin is called a satoshi, after the pseudonymous inventor of bitcoin; at a bitcoin price of US$10,000, a satoshi is worth US$0.0001. Cryptotoken fractionalization is defined programmatically at the time the token is created. In practice, tokenized assets can be fractionalized, and fractionalized units can be traded and managed, more easily and with less friction than most offchain assets. Typically, fungible tokens can be fractionalized, but NFTs often cannot.

8.4 Cryptotoken Use Cases

There are several reasons that an enterprise may decide to issue or interact with cryptotokens. Different types of tokens are more appropriate for different use cases. The following are common use cases for tokens.

- **Payments and settlement**—The original bitcoin use case is electronic payments. Payments remain the most straightforward use of cryptocurrencies, stablecoins and future digital fiat. In the business-to-consumer and consumer-to-consumer payment spaces, accepting bitcoin or other cryptotoken payments can be an alternative to existing electronic payment sources, with faster settlement and possibly lower fees. In the business-to-business space, cryptopayments can be a way to make large cash payments with same-day settlement and possibly the elimination of crossborder and other fees. For example, JP Morgan created JPM Coin to make large interbank payments at lower costs and with faster settlement times.[62]
- **Investing**—Cryptotokens represent new investable asset classes and a way to hold, trade and settle investments in securities. Cryptotokens exist in distributed ledgers, eliminating the need for central depositories, and

[60] Blenkinsop, C.; "Nonfungible Tokens, Explained," Cointelegraph, 26 July 2018, https://cointelegraph.com/explained/non-fungible-tokens-explained
[61] Buterin, V.; "On Settlement Finality," Ethereum Blog, 9 May 2016, https://blog.ethereum.org/2016/05/09/on-settlement-finality/
[62] J.P. Morgan, "J.P. Morgan Creates Digital Coin for Payments," https://www.jpmorgan.com/global/news/digital-coin-payments

settlement is in near real time, eliminating the need for clearinghouses and other central counterparties. All this can reduce cost, risk, friction and time.

- **Raising capital**—By creating and selling a cryptotoken, enterprises can raise capital. These tokens can take the form of digital versions of traditional securities, such as debt or equity; the tokenization of other types of assets that can then be fractionalized and sold, or possibly through an initial coin offering of a nonsecurity consumer utility token, although the regulatory environment for raising capital through nonsecurity tokens is currently extremely unclear.
- **Applications and services**—An enterprise can create a token that is intrinsic to how its product or service functions with users, using the token for payment, membership, rewards and/or governance. A well-designed utility token can theoretically create a powerful network effect in which customers and/or suppliers and the enterprise benefit from the value collectively built around the token.
- **Governance**—A token can be used by token holders to collectively make governance decisions, decentralizing and democratizing key decision making, much like how proxy voting allows equity holders to have a say in how a corporation is governed.

8.5 Types of Cryptotokens

There are many different types of cryptotokens serving different purposes. There are also many classification schemas to organize the universe of cryptotokens. Moreover, there are hybrids and edge cases that do not fit neatly into a single classification. Following are many of the most prominent types of cryptotokens.

- **Cryptocurrencies**—A natively onchain, stateless currency. Bitcoin is the most well-known example, although there are many others. Many cryptocurrencies have an algorithmic monetary policy hard-coded to the token governing rates and triggering circumstances of inflation and deflation and possibly a fixed money supply. For example, bitcoin has a hard-coded predefined inflation rate that is permanently capped at 21 million bitcoins.

  Cryptocurrencies are sometimes referred to as cryptocommodities (as a metaphor, not as a regulatory policy). Bitcoin is sometimes called digital gold because of its predictable supply, its relative scarcity and its use as a store of value. Ether, the native cryptocurrency for the Ethereum blockchain, is sometimes called digital oil because it is used not just as a payment currency and a store of value but also to fuel the storage, processing and delivery of the smart contract–based business and economic ecosystems built on that blockchain.
- **Security token**—A token that represents an equity, debt, derivative or other regulated security. These can be private or public securities. The security can be natively digital or a tokenized representation of a security that exists offchain. Issuers and sellers of these securities are required to follow the securities regulations for relevant jurisdictions.
- **Tokenized tangible and intangible assets**—A token that represents another asset, such as a piece of real estate or art (tangible) or intellectual property rights (intangible). These may or may not be considered securities.
- **Central bank digital currency (CBDC)**—A token that represents a government-issued currency that usually also exists in an offchain form (e.g., the US dollar). Sometimes referred to as digital fiat. Typically, these are issued by the central bank, government treasury or monetary authority that issues the offchain currency. CBDC can also enable programmable money, in which governance is encoded in the token and enforceable programmatically.[63, 64]
- **Stablecoins**—Cryptotokens whose value is pegged to another asset (usually a fiat currency) or basket of assets. Even the largest market cap cryptotokens are highly volatile compared to major currencies, which creates significant exchange-rate risk. Many businesses prefer to engage with the cryptouniverse on a dollar (or other fiat) basis. For investing or hedging purposes, businesses may want exposure to assets (dollar, gold, etc.), but

[63] Velissarios, J.; "The (R)evolution of Money II: Blockchain Empowered CBDC," Accenture, 15 January 2020, https://www.accenture.com/us-en/insights/blockchain/evolution-money

[64] Barontini, C.; H. Holden; "Proceeding With Caution—A Survey on Central Bank Digital Currency," BIS, 8 January 2019, https://www.bis.org/publ/bppdf/bispap101.htm

with the benefits of blockchain-based tokens. Finally, having a pegged onchain token allows the movement of value from highly volatile cryptotokens into the relative safety of the US dollar, the euro or a highly liquid commodity.[65] Stablecoins can maintain their peg through a variety of methods:

- **Fiat collateralized**—A user deposits a unit of fiat currency into a bank account and, in return, gets a unit of the stablecoin for that currency. The deposit is locked up until the token is returned in exchange for the deposited fiat.
- **Offchain asset collateralized**—Similar to fiat collateralized, except instead of fiat currency, another asset (e.g., gold, silver, diamonds) is deposited in exchange for a unit of the stablecoin.
- **Digital asset collateralized**—Like fiat collateralized, except another cryptotoken is deposited in exchange for the stablecoin. This can be done to create a price-pegged token on a different blockchain (e.g., Wrapped Bitcoin on Ethereum, with the price pegged to bitcoin on the Bitcoin blockchain) or to algorithmically create a fiat currency peg (e.g., Maker DAO, which uses algorithmically controlled deposits of ether and other cryptotokens on Ethereum to create a token pegged to the US dollar).
- **Algorithmic**—Stablecoins in which the peg is maintained by algorithms controlling supply, demand and reserves—currently more theoretical than in practice.
- **Hybrid**—Stablecoins in which the peg is maintained by a hybrid of the previous approaches.

- **Consumer utility tokens**—Sometimes referred to as consumer tokens or utility tokens, these are generally considered nonsecurity tokens (although this proposition has not been fully tested with regulators) that are used with blockchain-based applications and services. These tokens may grant access to the product, serve as a security deposit, be granted as rewards or incentives for certain behavior, be used for governance and voting, or provide discounts. Many of these tokens were sold in ICOs as a way to fund the development of the application/service and then used in the operation of that application/service.
- **Nonfungible tokens (NFTs)**—A token that, because of its uniqueness, is not interchangeable with another of the same token. If the metaphor for a fungible token is a dollar bill, a nonfungible token can be compared to a sports trading card, where each card has distinct properties and value. Nonfungible tokens typically cannot be fractionalized. Typical NFT use cases include ownership title (e.g., real estate, auto), art object ownership and tickets (e.g., events, airplane).

  A subcategory of NFTs are digital collectibles. The most famous example is CryptoKitties, a blockchain game where each user purchases and collects unique digital cats.

8.6 Cryptotoken Technology Standards

There are many different programming languages and codebases used to create cryptotokens, complicating how systems interact with tokens. Each blockchain platform operates separately from every other, meaning that a token on one blockchain cannot (easily) be traded or sent to another blockchain. There are some common technology approaches and emerging technology standards that help to simplify integration and interoperability.

8.6.1 Native Tokens

Native tokens exist at the protocol level of a blockchain (most frequently public blockchains), are unique to that blockchain, are first created at the genesis of the blockchain and are typically intrinsic to the functioning of that blockchain. Typically, native tokens are used to reward the successful processing of transactions or the creation of blockchain blocks (e.g., mining), and they are the native unit of payment on the blockchain. Examples of native tokens include bitcoin (native to the Bitcoin blockchain) and ether (native to the Ethereum blockchain). There are

[65] Bullmann, D.; J. Klemm; A. Pinna; "In Search for Stability in Crypto-Assets: Are Stablecoins the Solution?" European Central Bank, August 2019, https://www.ecb.europa.eu/pub/pdf/scpops/ecb.op230~d57946be3b.en.pdf

many others, including Dogecoin, LiteCoin, EOS and Cardano. Native tokens are typically described as cryptocurrencies.

8.6.2 Layer 2 Tokens

Layer 2 tokens are created by smart contracts on existing blockchains and are not intrinsic to the functioning of the underlying blockchain. The underlying blockchain can have a native token (e.g., Ethereum) or exist on a blockchain with no underlying native token (e.g., blockchains based on Fabric or Corda). Currently, the token types described, other than cryptocurrencies, are layer 2 tokens.

Because layer 2 tokens are created through smart contracts, a proliferation of tokens can make it very hard for users, wallets and smart contracts on a blockchain to interact with all the tokens. To mitigate this problem, token standards make sure that tokens can function seamlessly across the blockchain. Many of these standards are Ethereum Request for Comments (ERCs) on the Ethereum blockchain, which has the largest number of layer 2 tokens. The most common ERC token standards include:

- **ERC-20**[66]—The most common Ethereum token standard and the token type used by most ICOs and consumer utility tokens.
- **ERC-721**[67]—A standard for NFTs. CryptoKitties used the ERC-721 standard.
- **ERC-1400**[68]—A standard for security tokens.

Layer 2 tokens exist on other blockchains as well. One example is a creative approach to creating layer 2 tokens on Bitcoin:

- **Bitcoin colored coins**—The Bitcoin blockchain has much more limited programming capabilities than Ethereum and other blockchains that use smart contracts. Bitcoin's much more limited scripting language does allow a small amount of metadata to be attached to the smallest unit of a bitcoin token—a satoshi. These metadata can describe a specific asset, ownership right or other property affixed to that specific token or set of tokens—coloring those tokens to make them distinct from normal satoshis that represent only the value of that token. In this way, layer 2 tokens can be created on the Bitcoin blockchain.

8.7 How Tokens Are Issued

Minting is the process of creating tokens on a blockchain. Tokens are originally minted either as part of the genesis block of a blockchain (native tokens) or by deploying a smart contract that creates the tokens. The code that mints the tokens contains several parameters, typically including:

- The number of tokens created
- Whether the token is fungible or not fungible
- Whether the token can be fractionalized, and to how many decimal points
- Trading restrictions on the token
- Whether additional tokens can be minted in the future

After tokens are minted, they can then be issued, meaning they can be sent to users, smart contracts or addresses. There are several ways in which tokens are issued, many of which have been addressed previously, including:

- **Mining**—Tokens are sent to users as a reward for doing work that benefits the blockchain or DApps. For example, bitcoin miners compete to create each Bitcoin blockchain block. The winner is rewarded with a

[66] Ethereum Improvement Proposals, "EIP-20: ERC-20 Token Standard," https://eips.ethereum.org/EIPS/eip-20
[67] Ethereum Improvement Proposals, "EIP-721: ERC-721 Non-Fungible Token Standard," https://eips.ethereum.org/EIPS/eip-721
[68] GitHub, "ERC 1400: Security Token Standard #1411," https://github.com/ethereum/eips/issues/1411

specific number of bitcoins newly minted with that block. For Bitcoin, the size of the mining reward is algorithmically governed at a declining rate that is hard-coded into the Bitcoin blockchain—the rate halves about every four years until the predetermined maximum number of total bitcoins has been minted somewhere around year 2140, at which point no more bitcoins can be minted. The bitcoin mining reward was halved to 6.25 bitcoins per block in May 2020.[69]

- **Security token offering (STO)**—The issuing of a digital security by selling to investors. This is the onchain analog to the initial sale of an offchain security, such as an initial public offering (IPO) of an equity. STOs need to follow securities laws and regulations in the relevant jurisdictions, typically where the issuer is domiciled and where the security is sold. Just as with offchain security sales, service providers who facilitate the sale are also regulated (e.g., broker/dealers).
- **Initial coin offering (ICO)**—The term is most frequently used to describe the initial sale of a nonsecurity consumer utility token. ICOs use many different sale mechanisms, including selling at a fixed price and various auction models that allow the market to determine the auction clearing price. Many ICOs took place in 2017 and 2018, but due to the crash of cryptoprices in late 2018 and the increased scrutiny of regulators, ICOs have become much less common since early 2019. ICOs in 2017 and 2018 were also challenged by fraudulent actors. Sometimes the token-issuing project itself was fraudulent. At other times, scammers unrelated to the project found ways to misdirect investors to send money to them rather than to the ICO.
- **Airdrop**—The issuer sends tokens, usually for free, to specific wallets or addresses. This is typically used to start building a token community for utility tokens, increase or decentralize the base of token holders, create liquidity, market tokens, and get tokens to key influencers. Airdrops are sometimes scams or marketing schemes for token sales. These may run afoul of securities laws and regulations.

8.8 How Tokens Are Traded

A key feature of tradable tokens is that tokens traded on blockchains are settled in near real time (T0), with no settlement risk or need to post collateral against unsettled trades. The most basic way to trade a token is for one wallet or blockchain address that is holding a token to send that token to another wallet or address. This method does not guarantee that the sender receives anything back in return. To facilitate true trading between counterparties, many platforms have been launched. These can be grouped into three categories:

- **Over-the-counter (OTC)**—Similar to OTC trading in offchain capital markets, OTC token trades take place directly between two counterparties. Advantages are that the bid and offer can remain private until after the trade is completed, which can be helpful for large or strategically sensitive trades. OTC may be the only way to practically trade an asset with low trade volumes where there are no ready liquidity pools. The disadvantages include a lack of competition to obtain the best price. OTC trades are sometimes facilitated by brokers who help connect buyers and sellers. OTC trades can be crypto-to-crypto or fiat-to-crypto.[70]
- **Centralized exchanges (CEXs)**—Similar to offchain stock exchanges, CEXs are transaction venues that enable the posting of bids and offers for a set of listed tokens and a matching engine to match buyers and sellers. CEXs are distinct from DEXs, in that there is a central intermediary operating the exchange venue. CEXs are typically custodial, meaning that traders must deposit what they are trading into an account controlled by the CEX. This model can create faster and more liquid trading, but it also creates a point of security weakness that can be exploited by hackers. CEX operators are typically for-profit enterprises that capture fees from traders. CEX trades can be crypto-to-crypto or fiat-to-crypto.
- **Decentralized exchanges (DEXs)**—A trading model available only on blockchains where a decentralized set of smart contracts algorithmically matches bids and offers without using a central intermediary. Once launched, the DEX is not controlled by anyone or operated by a for-profit enterprise. Importantly, DEXs are typically

[69] Zhao, W.; "Bitcoin Halving Arrives: Mining Rewards Drop for Third Time in History," Coindesk, 11 May 2020, www.coindesk.com/bitcoin-halving-arrives-mining-rewards-drop-for-third-time-in-history

[70] Dempsey, C.; "How Does Crypto OTC Actually Work?" Medium, 25 March 2019, https://medium.com/circle-research/how-does-crypto-otc-actually-work-e2215c4bb13

noncustodial, meaning traders control their tokens in their own wallets right up to the moment they execute the trade, which they approve using their private keys. This makes DEXs slower than CEXs, but also more secure. DEX trades can only be crypto-to-crypto.[71]

8.9 Security Concerns

Tokens are units of value; therefore, token holders need to protect themselves from the risk of losing tokens. Here is a rundown of typical security risk factors for tokens.

- **Blockchain risk**—The actual blockchain accounts ledgers for Bitcoin and Ethereum—by far the two largest blockchains by market capitalization—have never been successfully hacked since their respective launches in 2009 and 2015, despite constant attempts. That is a remarkable cybersecurity track record. That is not to say that tokens on those blockchains have not been stolen—just that the breaches were not the result of attacking the blockchains themselves. There are ways of attacking blockchains that may be successful in the future. The primary way to steal tokens on public blockchains that use PoW or PoS consensus is through a 51% attack,[72] in which the attacker controls over 50 percent of the total computer processing power (in PoW) or staked value (in PoS). In a decentralized blockchain with many miners/stakers, this is very difficult to do—nearly impossible in the cases of bitcoin and ether, at least so far. In other blockchains with greater centralization, however, the possibility of collusion leading to a successful 51% attack is higher. Private blockchains typically use other consensus protocols, such as Byzantine Fault Tolerance. The primary protection against theft is that the validating nodes are known. If they attempt to change the blockchain history, they can be caught and punished.
- **Cryptography**—The public/private key pairings and the linking of blocks in a blockchain are enabled through cryptography. The form of cryptography used by most blockchains is elliptic curve cryptography, which is regarded as globally secure—essentially unbreakable. Quantum computing, a technology in its infancy, may create computers powerful enough to break this cryptography. This is a problem that blockchains share with most other computer encryption and security systems. There are new quantum-resistant encryption technologies being explored; for example, the National Institute of Standards and Technology (NIST) is working on a post-quantum cryptography standard. Lattice-based cryptography approaches are among those being considered as possible final candidates for cryptographic standards. The security afforded by cryptography is only as good as the users' private keys.
- **Smart-contract risk**—Smart contracts are code, written by fallible humans, that may contain mistakes, errors or holes. Because smart contracts can be used to store and transfer tokens, a flaw in the smart-contract code may be exploited to transfer tokens in a way that was not intended by the programmer.

One of the more notorious examples is the Decentralized Autonomous Organization (DAO) project, which was hacked in 2016. The DAO was a fully decentralized venture fund in which people sent ether to the DAO smart contract, which then locked up those funds and released them only to projects the DAO contributors voted on to fund. The DAO raised over US$150 million in ether through this process. All the processes of collecting, storing, voting and distributing funds were controlled by the decentralized smart-contract code—not a person or enterprise. Hackers discovered an unintended flaw in the code and were able to siphon US$60 million in ether. The hackers even posted an open letter stating that because the action to move the US$60 million in ether was allowed by the code (even if unintentionally), it should be permitted.

Ultimately, the Ethereum Foundation made the decision to hard-fork the network to basically go back in time and undo the theft of ether. This resulted in a couple of big impacts. First, the theft caused a sharp drop in the value of ether. It is possible the hackers' real aim was to earn money off a large short position on ether, and they never intended to be able to keep the stolen DAO ether. (Due to the design of the DAO and the traceability of specific ether

[71] Deme, B.; " Decentralized vs. Centralized Exchanges," Medium, 24 January 2018, https://medium.com/herdius/decentralized-vs-centralized-exchanges-bdcda191f767

[72] Van Valkenburgh, P.; "51% Attack," Coin Center, 22 August 2018, https://coincenter.org/entry/what-is-a-51-attack-and-what-can-a-successful-attacker-do-1

tokens, it seems unlikely that the hackers could have used the stolen ether. However, the DAO and the people who supplied the stolen ether could not access it either, so it would have been lost to them too.) The other big impact was that some of the Ethereum community disagreed with the decision to hard-fork the network and kept participating in the old (not hard forked) chain. This chain is known as Ethereum Classic.[73]

Several smart-contract audit services have emerged to perform security audits on smart-contract code. This can give enterprises and their users more confidence in the security of the codebase. Audit certification can also help meet compliance and risk requirements for both token issuers and token investors.

- **Fraud**—One of the current challenges with crypto is verifying the identity of who controls the wallet or address where the tokens are sent. Uncertainty exists as to whether they are who they say they are and even what country they are in. This creates significant risk of fraud. During the ICO boom in 2017, when there was little regulatory oversight, several types of fraud were pursued. Some projects raising money by ICO never intended to use the funds for the stated purpose of building a business that might generate a return for investors, but simply disappeared with the money. Other fraudsters spoofed legitimate ICO marketing material, keeping all the information the same but changing the blockchain address where investors sent the money. This remains a challenge, particularly for public blockchains. There are several services that tag known good and fraudulent addresses and take other measures to give users more confidence that they are sending tokens to legitimate addresses.
- **Social engineering**—As the weakest part of digital security, humans can be exploited through phishing or other low-tech means. These types of attacks have been made on people who control private keys to token wallets or access to private blockchain networks. Hackers have called cell phone enterprises pretending to be someone known to have a large value of cryptotokens, telling the phone company they need to transfer their service from an old phone to a new phone. They then use the phone to access online accounts (potentially including cryptowallets) where the phone is the security backup and even use the phone for two-factor authorization.
- **Human error**—When keying in information to a computer, people sometimes make mistakes. For cryptotokens, this may be a mistake in the address the token is sent to or the number of tokens sent. Blockchain addresses—long strings of letters and numbers—make it hard to catch a typing error. In offchain transactions, these errors can usually be reversed because there are central intermediaries executing the trade and the trade settlement, and there are usually a few days between the trade and the settlement, allowing mistakes to be caught and reversed before the assets have actually changed hands. In blockchain transactions, there may not be a central intermediary, settlement is nearly instantaneous and transactions are irreversible.
- **Key management and custody**—Private keys give users the ability to send tokens from wallets. This means that having a private key is essentially owning the tokens in a wallet. As a result, managing these keys is a critical part of token security. There are several options and factors to consider for key management:
  - **Custodial wallets**—Cryptowallets serve as an access point to send and receive tokens and as on ramps/off ramps between crypto and fiat currency. The Bitcoin blockchain has never been successfully hacked. But wallets have been—with CEX custodial wallets being a target in a few high-profile thefts. In a custodial wallet, users put their private keys into a wallet controlled by an enterprise, typically a CEX. There are several advantages to custodial wallets, but they also present a centralized point of attack for hackers.[74]

    In 2013, Mt. Gox was the largest bitcoin exchange in the world. Then hackers stole a Mt. Gox master private key, gained access to all Mt. Gox customers' custodial wallets and siphoned off funds until early 2014. Mt. Gox became insolvent, with the hackers stealing US$450 million in bitcoin. The value today would be in the billions of dollars.[75]

[73] Gazi Güçlütürk, O.; "The DAO Hack Explained: Unfortunate Take-off of Smart Contracts," https://medium.com/@ogucluturk/the-dao-hack-explained-unfortunate-take-off-of-smart-contracts-2bd8c8db3562
[74] HTC Exodus, "A Crypto Dilemma: Custodial vs. Non-Custodial Wallets—HTC EXODUS," Medium, 6 August 2019, https://medium.com/@HTC.EXODUS/crypto-dilemma-custodial-non-custodial-wallets-htc-exodus-dba63eb6d4e6
[75] Norry, A.; "The History of the Mt Gox Hack: Bitcoin's Biggest Heist," Blockonomi, 31 March 2020, https://blockonomi.com/mt-gox-hack/

- **Single key versus multisig wallets**—In a single-key wallet, there is one private key that controls the sending of tokens. This creates two significant security issues: If that private key is stolen, the thief is able to transfer all the tokens out of the wallet. If a noncustodial wallet owner loses or forgets the private key or if the owner becomes incapacitated or dies, and no one else knows the private key, the tokens in that wallet can never be accessed. Cryptoanalytics enterprise Chainalysis estimated in 2017 that between 2.2 million and 3.7 million (US$20 billion to US$30 billion) bitcoins are permanently out of circulation due to lost private keys.[76]

  Multisig wallets are a solution to this risk. In a multisig wallet, two or more private keys are required to enable a token transaction. If one is stolen, the funds are still secure. It is also possible to create a two-of-three private key requirement so that if one private key is lost, the remaining private keys can still access the tokens.[77]

- **Cold storage**—Private keys can be stored offline in servers unconnected to the Internet. This prevents hackers or malware from accessing the private keys, so it can be a highly secure way to store large amounts of cryptotokens. There are downsides. One is convenience—the token owners cannot quickly and easily access, trade or use their tokens. The cold storage site also creates a centralized point of attack—either by breaking into the cold storage facility or by destroying or erasing the computers storing the keys.

  Large-scale cold storage providers try to limit this security risk. For example, Xapo built a cold storage server farm in a decommissioned Swiss military bunker in the Alps, with extremely high levels of physical security even from an electromagnetic pulse attack or a nuclear bomb.[78]

- **Hardware wallets**—Physical electronic devices store private keys and can then be disconnected from the computer and the Internet. There is often a PIN access code to provide a second layer of security. The device can then be physically secured in a safe or in a bank safety deposit box. It is ironic that tokens based on decentralized technology and cryptography are stored in traditional banks.
- **Cryptocustody**—In traditional capital markets, custodians are banks or other large financial services institutions that hold securities and other assets for safekeeping on behalf of their clients. Because these assets can total in the billions of dollars, custodians are typically large firms with strong reputations. In addition to holding the securities, custodians typically provide asset servicing and access to the asset when their clients need it. Cryptocustody works the same way: a large trusted enterprise holds private keys for primarily institutional clients, while giving clients access to the tokens when needed and providing value-added services.

8.10 Regulatory Considerations

Cryptotokens may be, or may be akin to, money, securities, commodities or none of these. Sometimes, they can seem like more than one of these items at the same time or like different ones at different times. As a result, issuing, trading, holding and/or investing in cryptotokens may fall under regulatory regimes for one or more of these offchain analogs. It is still in the early days for cryptotokens, so regulators in different jurisdictions have come to different conclusions about how to regulate different types of tokens, and in many cases, they have not yet reached conclusions about how to regulate them.

- **Money**—Cryptotoken transaction platforms and exchanges may need a money transmitter license (MTL) to operate. In the United States, each state has its own MTL.
- **Commodities**—Cryptoderivatives, swaps and futures may be regulated under commodities laws. Transaction venues for swaps may need to be licensed as a swap execution facility (SEF). In the United States, commodities, derivatives, swaps and futures are primarily regulated by the Commodity Futures Trading Commission (CFTC).

[76] Insights, "Bitcoin's $30 Billion Sell-Off," 8 June 2018, https://blog.chainalysis.com/reports/money-supply
[77] Binance Academy, "What Is a Multisig Wallet?" https://www.binance.vision/security/what-is-a-multisig-wallet
[78] Quartz, "Photos: The Secret Swiss Mountain Bunker Where Millionaires Stash Their Bitcoins," https://qz.com/1103310/photos-the-secret-swiss-mountain-bunker-where-millionaires-stash-their-bitcoins/

- **Securities**—Tokens may be securities that are governed under securities laws. In the United States, securities are primarily regulated by the Securities and Exchange Commission (SEC).

In addition to relevant regulations originally written for offchain activity that may apply to onchain activity and cryptotokens, there are cryptospecific regulations; a few examples include:

- New York State Virtual Currency Business Activity (BitLicense)[79]
- Wyoming blockchain and crypto laws—including banking and custodian licenses

Questions relating to cryptotokens and securities law are among the biggest issues surrounding cryptotokens since 2017. Two primary questions are:

- How are registered security tokens properly issued under existing law?
- What types of tokens are not securities and do not need to follow securities laws?

The potential issuance of tokens as securities continues to be an ongoing challenge for both issuers and regulators. The US Securities and Exchange Commission has become increasingly active in the policing of issuers and in launching investigations and enforcement actions, sometimes on the order of millions of US dollars.[80]

Likewise, regulation outside of the United States has proved to be diverse across multiple jurisdictions. Some notable, though not exhaustive, examples include:

- **Cryptohavens**—Several countries have set themselves up as attractive places for cryptotoken issuers and the cryptotoken trading and investing ecosystem to domicile and operate, often through a combination of low taxes, clear cryptoregulations and a regulatory environment in which it is easy to operate (e.g., where tokens need not be regulated as securities). These include Singapore, Malta and Gibraltar.
- **Switzerland**—Switzerland has been amending its robust financial regulations to work with cryptotokens and distributed ledger technology. Switzerland has become one of the main global centers of cryptotoken business activity, startups and innovation in what is being called Crypto Valley.[81]
- **China**—China has been a leader in cryptoactivity since the early days of bitcoin. Most of the world's bitcoin mining takes place in China—attributable to the availability of inexpensive electricity to certain mining operators, making mining more profitable in China than elsewhere. However, starting in 2017, China began cracking down on bitcoin and other cryptotoken trading and investing activities and on ICOs.[82] In November 2019, China issued a new series of warnings, which caused many China-based cryptoexchanges to cease operation and the price of bitcoin to plummet.[83] Despite these crackdowns, there are reports that China is investigating issuing its own national cryptocurrency or digital fiat.[84]

Users of this framework are strongly encouraged to keep up to date on evolving regulations and legislation, depending on their locality, and they should keep these in mind when considering blockchain adoption or integration for their enterprises.

[79] New York State Department of Financial Services, "Virtual Currency Business Activity," www.dfs.ny.gov/apps_and_licensing/virtual_currency_businesses
[80] Public information about these regulatory actions can be found at www.sec.gov/spotlight/cybersecurity-enforcement-actions.
[81] "China's Cryptocurrency Plan Has a Powerful Partner: Big Brother," Law Reviews, https://thelawreviews.co.uk/edition/the-virtual-currency-regulation-review-edition-2/1197602/switzerland
[82] Kharif, O.; "Cryptocurrency Exchanges Across China Halt Services Amid Crackdown," *Japan Times*, 28 November 2019, www.japantimes.co.jp/news/2019/11/28/business/chinas-crackdown-cryptocurrencies-claims-first-victims/
[83] Li, Y.; "Bitcoin Sinks to Lowest Level Since May, Falling $3,000 in a Month as China Accelerates Crackdown," CNBC, 25 November 2019, www.cnbc.com/2019/11/25/bitcoin-sinks-to-a-6-month-low-as-china-accelerates-crackdown.html
[84] "China's Cryptocurrency Plan Has a Powerful Partner: Big Brother," *New York Times*, 18 October 2019, www.nytimes.com/2019/10/18/technology/china-cryptocurrency-facebook-libra.html

8.11 Governance, Risk and Compliance (GRC)

Enterprises that interact with cryptotokens need to consider their governance, risk and compliance requirements beyond the regulatory issues described previously. This section focuses on two large areas of compliance:

- **Anti-money Laundering (AML), know your customer (KYC), counter financing of terrorism (CFT)**—Rules set up by regulators to ensure that the business activity of an enterprise is not supporting crime, terrorism or embargoed countries. Key to remaining compliant is knowing who one is doing business with and monitoring those transactions. In the United States, the Financial Crimes Enforcement Network (FinCEN) sets requirements for financial institutions. Different industries and countries may have different requirements. FinCEN four core elements of customer due diligence (CDD) for financial institutions are:[85]
  - Customer identification and verification
  - Beneficial ownership identification and verification
  - Understanding the nature and purpose of customer relationships to develop a customer risk profile
  - Ongoing monitoring for reporting suspicious transactions and, on a risk basis, maintaining and updating customer information
- **Enterprise-specific governance, risk and compliance (GRC)**—In addition to regulatory compliance requirements, enterprises may establish other compliance rules to meet their own internal or external requirements. These can include not doing with business with certain types of enterprises or industries (e.g., gambling) or adhering to a core mission statement or promise to customers or investors (e.g., divesting from certain industries or adhering to an ecological mandate). In addition, enterprises can put in place operational risk controls to protect the enterprise from unwanted losses due to employer or system actions.

In the offchain world, there are many third-party GRC data services that help enterprises stay compliant with regulations and internal policies. These data services include databases of people, enterprises and countries that are known to be high risk or out of compliance; transaction monitoring; and compliance reporting.

Enterprises interacting with cryptotokens have some distinct needs and challenges when it comes to meeting these compliance requirements. The biggest challenge is that enterprises may not know with confidence who controls the wallet or address with which they are sending or receiving tokens. Some transaction venues have KYC requirements to participate in the venue, which may provide sufficient confidence about who the counterparty is. But some transaction venues do not have KYC requirements, and direct peer-to-peer trading may not have KYC requirements either. Blockchain-based transactions have advantages in that tokens may be tracked across transactions and times to see how wallets and addresses are connected in a way that is better than current electronic money transfers (e.g., via SWIFT) and far superior to paper cash. Note that some emerging blockchain transaction privacy technologies, such as zero-knowledge proofs,[86] have legitimate use cases and can also be used to make cryptotoken transactions impossible to track or monitor.

There is an emerging market for cryptotoken compliance solutions to help enterprises meet their compliance requirements. These include the capability to:

- Tag known addresses and flag those that are high risk or out of compliance
- Monitor transactions and track tokens from address to address to see if the business activity of an enterprise may be connected to out-of-compliance actors or activity
- Provide risk assessments, dashboards and reporting capabilities

[85] Federal Register, "Customer Due Diligence Requirements for Financial Institutions," www.federalregister.gov/documents/2016/05/11/2016-10567/customer-due-diligence-requirements-for-financial-institutions

[86] Hay Newman, L.; "Hacker Lexicon: What Are Zero-Knowledge Proofs?" Wired, 14 September 2019, https://www.wired.com/story/zero-knowledge-proofs/

8.12 Resources

Following are resources for blockchain controls:

- COBIT® 2019
- NIST Cybersecurity Framework V1.1
- ISO/IEC 27001 Information Technology—Security Techniques—Information Security Management Systems—Requirements

Page intentionally left blank

APPENDIX A
Blockchain Controls

This appendix provides a complete set of high-level blockchain control objectives, derived from common organizational objectives, to be considered by management. These control objectives ensure the security, availability and processing integrity of blockchain networks and their larger ecosystem, and they are based on the following five key risk domains that are unique to blockchains:

- Governance
- Infrastructure (includes all data-independent functionality of a blockchain)
- Data
- Key management
- Smart contracts

Key Questions Enterprises Need to Answer

Following are some questions that enterprises need to answer regarding blockchain control objectives:

- Do the proposed blockchain control objective domains adequately cover risk vectors and business process objectives for the organization transacting in this space?
- Have we identified the relevant stakeholders of blockchain control objectives?
- Do we understand our engagement with the larger blockchain ecosystem in evaluating risk and control objectives?

Stakeholders

Stakeholders for blockchain controls include:

- Board of directors
- Executive management
- Business unit managers
- IT managers/practitioners
- Cybersecurity/information security
- Assurance providers
- Risk management team
- Regulators
- Business or vendor partners
- Miners/node owners
- Regulators

Control Objectives

The following table presents a complete set of high-level requirements to be considered by management for effective control of each information and technology (I&T) blockchain process. The control objectives are organized within

the five blockchain domains. The table also provides the preexisting frameworks and controls for established I&T processes.

Note: Whenever possible, control objectives should be addressed on blockchain/distributed ledger technology networks with control activities that leverage closed-loop, automated, continuous-control analytics.

Blockchain: Governance—Regulatory and management oversight guidelines/requirements of the blockchain/distributed ledger technology (BC/DLT) networks policies and procedures.

G-1: Assure/review regulatory compliance with all jurisdictions impacted by the network.	**Related Framework Guidance**
G-1.01: Assure the BC/DLT technology is an approved, or at least not a prohibited, technology in all sovereign jurisdictions where the network operates or may operate.	COBIT 2019 MEA03.01, MEA03.04 NIST CSF ID.GV-3 ISO/IEC 27001:2013 A.18.1
G-1.02: Assure via location stamps or other automated means that the network does not/cannot operate/transact in high-risk or sanctioned regions.	COBIT 2019 APO13.01, BAI03.02, DSS05.03 NIST CSF PR.IP-5, PR.PT-1 ISO/IEC 27001:2013 A.18.1
G-1.03: Perceived or confirmed regulatory conflicts between jurisdictions should be fully disclosed and documented with input from all stakeholders. If unable to resolve to the mutual satisfaction of both parties, the network should cease to operate in the impacted jurisdictions.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST ID.GV-3 ISO/IEC 27001:2013 A.18.1
G-1.04: Assure compliance with privacy standards in the respective jurisdictions where the network operates, e.g., General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA). Hard encryption, data segregation, cybersecurity and compliant notification/response plans must all be in place and tested.	COBIT 2019 APO01.01, APO01.09, APO13.02, BAI01.07, BAI03.03, BAI03.07, BAI03.08, DSS06.02, MEA03.01 NIST CSF ID.GV-3 ISO/IEC 27001 A.18.1

G-2: Assure all assets (e.g., cryptocurrencies and tokens) transacting on the BC/DLT network are sanctioned, or at least not prohibited, assets in all sovereign jurisdictions where the network operates or could operate.	**Related Framework Guidance**
G-2.01: Assure via location stamps and asset/token identification or other automated means that assets transacted on the network are not/cannot transact in high-risk or sanctioned regions.	COBIT 2019 APO13.01, BAI03.02, DSS05.03 NIST CSF PR.AC-3, PR.AC-6 ISO/IEC 27001:2013 A.9.1.2
G-2.02: Perceived or confirmed regulatory conflicts regarding assets transacted between jurisdictions should be fully disclosed and documented with input from all stakeholders. If unable to resolve to the mutual satisfaction of both parties, the network should cease to operate in the impacted jurisdictions.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST CSF ID.GV3 ISO/IEC 27001:2013 A.18.1

G-3: Configuration of management/protocol code.	**Related Framework Guidance**
G-3.01: Assure the network is designed and operating effectively to meet the objectives of the public or private network nodes/network participants.	COBIT 2019 BAI04.05, DSS05.02, DSS05.03, MEA02.01, MEA04.07 NIST CSF ID.BE-3

G-3: Configuration of management/protocol code.	Related Framework Guidance
G-3.02: Provide an ongoing independent review of the current and future capability and security of the protocol code in support of the network objectives.	COBIT 2019 APO11.05, BAI03.10, MEA04.01, MEA04.06 NIST CSF PR.IP-2, PR.IP-7 ISO/IEC:2013 A.12.1 A.14.2
G-3.03: Assure that approved changes/updates to protocol code, smart contracts, consensus, etc. follow a secure change management process and are appropriately approved by the consensus mechanism or other change management process supporting the network.	COBIT 2019 BAI03.09, BAI03.10, BAI07.01, BAI07.02 NIST CSF PR.IP-2, PR.IP-3 ISO/IEC:2013 A.12.1 A.14.2
G-3.04: Ensure appropriate approvals via the consensus mechanism of the network for all assets that are disposed or retired from the blockchain, which may lead to invalid or fraudulent asset deletions.	COBIT 2019 APO14.08, APO14.09, BAI09.03 NIST CSF ID.AM-4 ISO/IEC:2013 A.11.2
G-3.05: Ensure that data and platform formats align with standards (voluntary or required) to enable desired network interoperability.	COBIT 2019 APO14.02, BAI03.03, BAI03.08

G-4: Manage capacity/sustainability of the network.	Related Framework Guidance
G-4.01: Conduct regular analysis/simulation of the network's current and anticipated future capacity. This must be conducted to determine whether the network is capable of scaling to meet current and future volume and hashing demands. Assure that the speed of the consensus process facilitates a sufficient transaction throughput to achieve the objectives of the network.	COBIT 2019 APO08.01, APO08.02, APO09.01, BAI04.01, BAI04.03, BAI04.04 NIST CSF PR.DS-4 ISO/IEC:2013 A.17
G-4.02: Assure that block size, block interval time and block depth are sufficient to meet current and anticipated resource demands of the network and to meet objectives.	COBIT 2019 APO08.01, APO08.02, APO09.01, BAI04.01, BAI04.03, BAI04.04 NIST CSF PR.DS-4 ISO/IEC:2013 A.17
G-4.03: Monitor transmission size of the ledger to ensure it can be efficiently distributed to miners/nodes currently and as the network scales (e.g., preventing bloat). Take appropriate actions to compress or archive the network.	COBIT 2019 APO08.01, APO08.02, APO09.01, BAI04.01, BAI04.03, BAI04.04 NIST CSF PR.DS-4 ISO/IEC:2013 A.17

G-5: Ensure node/network access, validation.	Related Framework Guidance
G-5.01: Protect the network from access by rogue or unauthorized nodes in permissioned and public BC/DLT networks.	COBIT 2019 APO13.01, DSS01.04, DSS05.03 NIST CSF PR.MA-2, DE.CM-7, PR.AC-3 ISO/IEC 27001:2013 A.13.1, A.13.2
G-5.02: Employ secure, preventative analytics secured by IT general controls to mitigate algorithm access vulnerabilities on the blockchain, preventing unauthorized and/or malicious use of the blockchain network for hostile aggregation (e.g., 51% attacks, hash collisions, consensus compromises).	COBIT 2019 APO13.01, DSS05.02, DSS05.03, DSS06.05, DSS06.06 NIST CSF PR.DS-1, PR.DS-2, PR.DS-5, PR.DS-6 ISO/IEC 27001:2013 A.14.1, A.14.2
G-5.03: Assure appropriate know-your-customer (KYC) and anti–money laundering (AML) screening on exchanges and other on ramps to the BC/DLT network.	COBIT 2019 DSS06.02, MEA03.04 NIST CSF PR.DS-6, DE.DP-2, ID.GV-3 ISO/IEC 27001:2013 A.18
G-5.04: Assure that inactive or terminated users are not able to view or access any transactions on the network.	COBIT 2019 DSS05.04, DSS06.03 NIST CSF PR.AC-1 ISO/IEC 27001:2013 A.9.2.1

G-6: Provide for comprehensive audit and monitoring of the BC/DLT network.	Related Framework Guidance
G-6.01: Provide a system of record or an on-chain automated, continuous, closed-loop-based audit/monitoring process addressing IT general controls for the BC/DLT network.	COBIT 2019 APO14.02, APO14.05, MEA01.01, MEA02.01 NIST CSF PR.PT-1 ISO/IEC 27001:2013 A.12.4, A.12.7
G-6.02: Establish and confirm that there is an integrated audit/monitoring process in place addressing the larger BC/DLT ecosystem supported by service organization control (SOC) reviews. Examples: Exchanges, oracles, banks, etc.	COBIT 2019 EDM01.03, MEA01.01, MEA02.01, NIST CSF PR.PT-1 ISO/IEC 27001:2013 A.12.4, A.12.7

G-7: Threats of centralization and concentration of power.	Related Framework Guidance
G-7.01: Establish and monitor to assure that no single node or group of nodes, miners/stakers/etc., or mining pools controls an inappropriate percentage of the hashing, DoS-gas, nodes, consensus or network resources at any time.	COBIT 2019 APO14.05, APO14.06, BAI03.02 NIST CSF ID.AM-4, DE.CM-6 ISO/IEC 27001:2013 A.14.2
G-7.02: Programmatically ensure that the network (permissioned, unpermissioned, hybrid, etc.) always follows the approved consensus mechanism before any transactions or changes are submitted to the network.	COBIT 2019 BAI03.02, DSS06.02 NIST CSF PR.DS-6 ISO/IEC 27001:2013 A.14.1, A.14.2
G-7.03: Prevent privileged nodes/persons (miners, stakers, compliance nodes, administrative nodes, etc.) from viewing or participating in potential insider trading or otherwise front-running transactions awaiting submission to the blockchain.	COBIT 2019 BAI02.01, DSS05.04, DSS05.07, DSS06.03, MEA03.01 NIST CSF ID.GV-3 ISO/IEC 27001:2013 A.18

Blockchain: Infrastructure—Any functionality or capability of a blockchain independent of the data transacting on the blockchain or between blockchains.

I-1: Protocol Code: Ensure that the blockchain protocol design and code appropriately support the objectives of the blockchain use case, providing adequate security, availability, processing integrity, confidentiality and privacy. The protocol should mitigate the potential for misuse and unauthorized disclosure by an attacker.	Related Framework Guidance
I-1.01: Monitor the protocol to assure sufficient liveness (speed of the consensus process that facilitates a sufficient transaction throughput) to achieve performance objectives of the user organizations.	COBIT 2019 BAI04.04 NIST CSF DE.CM-1 ISO/IEC 27001:2013 A.13.1.1
I-1.02: Ensure that the protocol is secure and accurate (the ability to make decisions of correctness) and in line with the risk appetite and risk tolerance of the user organizations.	COBIT 2019 BAI03.08, DSS06.01 NIST CSF PR.AC-7, PR.DS-2, PR.DS-5, ISO/IEC 27001:2013 A.14.2
I-1.03: Ensure that new node synchronization and node block requests are managed, to prevent blockchain malfunctions due to hardware errors within the system, resulting in corrupt data or latency of service.	COBIT 2019 BAI09.01, BAI09.03, DSS03.01 NIST CSF PR.DS-8, PR.IP-3, ID.AM-4 ISO/IEC 27001:2013 A.14.2.2

I-1: Protocol Code: Ensure that the blockchain protocol design and code appropriately support the objectives of the blockchain use case, providing adequate security, availability, processing integrity, confidentiality and privacy. The protocol should mitigate the potential for misuse and unauthorized disclosure by an attacker.	**Related Framework Guidance**
I-1.04: Preventive controls should ensure that the sum of all transactions in a block does not exceed the allowable block size. Transaction sizes larger or smaller in bytes than allowed by agreed-on parameters may result in compromise to the network performance or integrity.	COBIT 2019 APO11.05, BAI11.05, DSS05.07, DSS06.01, MEA02.01 NIST CSF PR.PT-1, PR.IP-7 ISO/IEC 27001:2013 A.12.6.1
I-1.05: Ensure that critical transaction sequencing or timing is carefully followed on the blockchain design or protocol. Failure to do so may result in compromise to the network performance or integrity.	COBIT 2019 APO11.05, BAI11.05, DSS05.07, DSS06.01, MEA01.04, MEA02.01 NIST CSF PR.PT-1, PR.IP-7, PR.DS-ALL ISO/IEC 27001:2013 A.12.6.1
I-1.06: Control activities should ensure ongoing performance by providing appropriate structural constraints for the block height before any type of forks or archiving. Mechanisms must be in place to ensure that the block height does not exceed a specified height, causing structural violations that could negatively impact the stability or integrity of the chain.	COBIT 2019 APO11.05, BAI11.05, DSS05.07, DSS06.01, MEA01.04, MEA02.01 NIST CSF PR.PT-1, PR.IP-7, PR.DS-ALL ISO/IEC 27001:2013 A.12.6.1
I-1.07: Ensure that block height management, for which the consensus mechanism should be currently reviewing or approving is the same for all approving nodes. Consensus nodes approving wrong or different blocks negatively impact the stability and integrity of the blockchain.	COBIT 2019 APO11.05, BAI11.05, DSS05.07, DSS06.01, MEA01.04, MEA02.01 NIST CSF PR.PT-1, PR.IP-7, PR.DS-ALL ISO/IEC 27001:2013 A.12.6.1
I-1.08: Control activities should ensure an appropriately long block depth for higher-risk transactions so that the stability and integrity of the blockchain are not compromised.	COBIT 2019 APO11.05, BAI11.05, DSS05.07, DSS06.01, MEA01.04, MEA02.01 NIST CSF PR.PT-1, PR.IP-7, PR.DS-ALL ISO/IEC 27001:2013 A.12.6.1
I-1.09: Ensure the appropriate verification of timestamps, so that block timestamps are no more than a certain number of seconds/minutes in the future of the current system time, based on a configuration parameter. Timestamps that exceed the limit may result in fraud or abuse on the network.	COBIT 2019 APO11.05, BAI11.05, DSS05.07, DSS06.01, MEA01.04, MEA02.01 NIST CSF PR.PT-1, PR.IP-7, PR.DS-ALL ISO/IEC 27001:2013 A.12.6.1
I-1.10: Ensure appropriate automated monitoring of segregation of duties (SoD), where certain nodes have defined roles resulting from the agreed baseline blockchain protocol, contract, consensus mechanism or other participation agreements that determine which nodes have read/write/delete access to protocol code, smart contracts and transactions. Risk is increased if certain nodes have too much power or access, resulting in negative influence and the ability to cause harm to the network.	COBIT 2019 DSS05.04, DSS06.01, DSS06.02, DSS06.03 NIST CSF PR.AC-4, PR.AC-6, PR.AC-7 ISO/IEC 27001:2013 A.6
I-1.11: Ensure that automated preventive monitoring/alerting is in place to prevent unauthorized soft or hard forks of the network (e.g., unauthorized consensus-approved forks) by internal or external actors, resulting in damage to the network.	COBIT 2019 DSS05.02, DSS05.03, DSS05.04 NIST CSF DE.CM-1, DE.CM-3, DE.CM-4 ISO/IEC 27001:2013 A.12.4

I-2: Consensus Mechanism: Ensure that the blockchain consensus mechanism is designed and operating effectively to meet the objectives of the network. The consensus should be monitored to mitigate the potential for fraud or abuse on the network.	Related Framework Guidance
I-2.01: Verify signers, ensuring that the number of signature operations (SIGOPS) for authorized block signers contained in the transactions is less than the SIGOPS limit, which prevents block signers from signing multiple blocks at the same height and enabling participants to use the inconsistent signatures to construct fraud proofs.	COBIT 2019 BAI03.05, DSS05.04, DSS06.01, DSS06.03 NIST CSF PR.AC-1, PR.AC-7 ISO/IEC 27001:2013 A.14.1.1
I-2.02: Verify that block signers do not sign multiple blocks at the same height, resulting in inconsistent signatures to construct fraud proofs. Signers signing multiple blocks at the same height may enable network participants or the signers themselves to perpetrate a fraud in the consensus.	COBIT 2019 BAI03.05, DSS05.04, DSS06.01, DSS06.03 NIST CSF PR.AC-1, PR.AC-7 ISO/IEC 27001:2013 A.14.1.1
I-2.03: Ensure that if a block signer signs multiple blocks at the same height, participants can use the inconsistent signatures to construct fraud proofs to warn other nodes or provide evidence for enforcement out-of-band. Failure to ensure that block signers do not sign multiple blocks at the same height may enable fraud in the consensus.	COBIT 2019 BAI03.05, DSS05.04, DSS06.01, DSS06.03 NIST CSF PR.AC-1, PR.AC-7 ISO/IEC 27001:2013 A.14.1.1
I-2.04: Ensure that miners cannot manipulate a block timestamp—consider all direct and indirect uses of the timestamp. Timestamp manipulation may result in fraud or abuse on a blockchain by a bad actor.	COBIT 2019 APO01.07, BAI03.05, DSS05.03, DSS06.02 NIST CSF PR.DS-6 ISO/IEC 27001:2013 A.14.2
I-2.05: Ensure that when validator nodes on private blockchains are added to or deleted from the network voting protocol, thresholds are updated to ensure that enough validator nodes are present on the network to complete the consensus mechanism and prevent consensus failures.	COBIT 2019 APO01.07, BAI03.05, DSS05.03, DSS06.02 NIST CSF PR.DS-6 ISO/IEC 27001:2013 A.14.2

I-3: Third-Party/Integration Vulnerabilities.	Related Framework Guidance
I-3.01: Ensure that monitoring and analytic controls are in place to prevent sybil attacks, in which the attacker attempts to corrupt a peer-to-peer network by forming fake identities/nodes that participate and manipulate transactions, which may compromise individual nodes and the entire blockchain.	COBIT 2019 DSS01.03, DSS03.05, DSS05.07 NIST CSF PR.AC-1, DE.AE-1, DE.CM-1 ISO/IEC 27001:2013 A.13.1.2
I-3.02: Ensure monitoring and analytic controls are in place to prevent anti-pattern attacks similar to sybil attacks, in which attackers establish several fake nodes that appear to be genuine to their peers. These fake nodes take part in corrupting the network to validate unauthorized transactions and to alter valid transactions. The attackers can use several devices, virtual machines or Internet protocol (IP) addresses as fake nodes for the attack to out vote legitimate nodes and favor malicious transactions.	COBIT 2019 DSS01.03, DSS03.05, DSS05.07 NIST CSF PR.AC-1, DE.AE-1, DE.CM-1 ISO/IEC 27001:2013 A.13.1.2
I-3.03: Ensure monitoring and analytic controls are in place to prevent a border gateway protocol (BGP) hijacking, i.e., routing attack, which is an exploit against BGP protocol, where the Internet service provider (ISP) makes false announcements over the routing system to divert traffic. This attack typically requires a combination of partitioning the networks and then diverting traffic to delay transaction processing.	COBIT 2019 DSS01.03, DSS03.05, DSS05.07 NIST CSF PR.AC-1, DE.AE-1, DE.CM-1 ISO/IEC 27001:2013 A.13.1.2

I-4: Algorithm Vulnerabilities: Assure blockchains use specific algorithms and sometimes specific hardware to complete the blockchain lifecycle process. These processes can be vulnerable depending on their design and dependencies.	**Related Framework Guidance**
I-4.01: Ensure there is not and cannot be an inappropriate concentration of node control or hashing power by any authorized node or network participant to enable a compromise of the integrity of the network.	COBIT 2019 DSS05.01, DSS05.02, DSS05.04, DSS05.07 NIST CSF ID.RA-1,ID.RA-5, PR.AC-4, PR.AC-5 ISO/IEC 27001:2013 A.12.6.1
I-4.02: Ensure there are not unauthorized/or malicious use of the blockchain network for hostile aggregation of 51%-attack capabilities. Monitor and ensure that the hash rate of the network is maximized and is geographically and noncentrally distributed so that no single entity or entities can collude and seize control. Leverage limiting safeguards, such as checkpointing and notarizing of blocks.	COBIT 2019 DSS05.01, DSS05.02, DSS05.04, DSS05.07 NIST CSF ID.RA-1,ID.RA-5, PR.AC-4, PR.AC-5 ISO/IEC 27001:2013 A.12.6.1
I-4.03: Ensure appropriate cybersecurity monitoring and controls are in place to prevent cryptojacking attacks—an emerging attack that leverages malware to install PoW mining code on the user devices to steal their computing resources such as CPU or GPU to mine for cryptocurrencies for the malware owner.	COBIT 2019 DSS05.01, DSS05.02, DSS05.04, DSS05.07 NIST CSF ID.RA-1,ID.RA-5, PR.AC-4, PR.AC-5, DE.CM-4 ISO/IEC 27001:2013 A.12.6.1

I-5: Hash Collision Attacks: Employ preventive automated monitoring controls to identify and remediate potential hash collision attacks where two input strings of a hash function produce the same hash result. Controls objectives should cover master data; collision exploitation, transaction data; double spend, unauthorized entry, validity, inaccurate data, missing data, duplicate data, etc.	**Related Framework Guidance**
I-5.01: Ensure processes/procedures are in place to prevent and detect eclipse attacks where blockchain nodes connect to neighboring peers via a peer-to-peer communication protocol to process transactions and maintain the blockchain ledger. The attacker seeks to isolate the victim node by blocking its access to the adjacent peers and then trick the victim node with false transaction information, thereby compromising the attacked node and, potentially, the network.	COBIT 2019 DSS05.01, DSS05.02, DSS05.04, DSS05.07 NIST CSF ID.RA-1,ID.RA-5, PR.AC-4, PR.AC-5, DE.CM-4 ISO/IEC 27001:2013 A.12.6.1

Blockchain: Data—Offchain information stored and/or transmitted in a computer-readable format that is used to transact or interact on a blockchain network, or data that is sourced from a blockchain network and treated as a source of truth for a business purpose.

D-1: Data Integrity: BC/DLT networks must be designed to ensure data integrity and nonrepudiation, mitigating risk involved with this immutable type of data storage.	**Related Framework Guidance**
D-1.01: Ensure all input transactions have a corresponding output transaction—preventing orphan transactions on the network.	COBIT 2019 APO14.08 NIST CSF PR.DS-1, PR.DS-2 ISO/IEC 27001:2013 A.8.2.3

D-1: Data Integrity: BC/DLT networks must be designed to ensure data integrity and nonrepudiation, mitigating risk involved with this immutable type of data storage.	**Related Framework Guidance**
D-1.02: Ensure that orphan transactions are added to the orphan transaction pool (if applicable). If the matching transaction is not already in the pool, this may indicate a matching transaction problem, which may adversely affect the integrity of the network and may be an indicator of fraud or abuse.	COBIT 2019 APO14.08, DSS05.02 NIST CSF PR.DS-1, PR.DS-2, PR.AC-5 ISO/IEC 27001:2013 A.13
D-1.03: Ensure procedures are in place such that multi-sig/approval requirements for specific transactions are confirmed before a requested transaction can proceed.	COBIT 2019 APO14.08, DSS05.02 NIST CSF PR.DS-1, PR.DS-2, PR.AC-5 ISO/IEC 27001:2013 A.13
D-1.04: Ensure controls are in place so that miners/stakers cannot manipulate data/transaction timestamps resulting in fraud or abuse on a blockchain by a bad actor/node.	COBIT 2019 APO14.08, DSS05.02 NIST CSF PR.DS-1, PR.DS-2, PR.AC-5 ISO/IEC 27001:2013 A.13
D-1.05: Ensure that nonstandard transactions are not submitted or approved on the network, which may compromise the network and facilitate fraud.	COBIT 2019 APO14.08, DSS05.02 NIST CSF PR.DS-1, PR.DS-2, PR.AC-5 ISO/IEC 27001:2013 A.13

Blockchain: Key Management—Management of public and private keys.

KM-1: Insufficient Entropy (lacking appropriate complexity, vulnerable to compromise/hacking): Ensure that keys are designed/created with appropriate complexity and hard encryption to mitigate risk of compromise.	**Related Framework Guidance**
KM-1.01: Ensure key/seed is created with sufficient required entropy, mitigating risk that keys can be brute forced or guessed, resulting in a loss of assets. All future use of the keys for storing and transacting in cryptoassets is at risk if sufficient entropy is not used in the beginning in key creation.	COBIT 2019 APO13.01, BAI02.01, BAI03.05, DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6 ISO/IEC 27001:2013 A.10.1.2
KM-1.02: Ensure key/seed creation methodology is validated prior to creation, preventing generating keys with low entropy.	COBIT 2019 APO13.01, BAI02.01, BAI03.05, DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6 ISO/IEC 27001:2013 A.10.1.2
KM-1.03: Ensure key/seed was not created by the operator or owner. If a third party is relied on to generate the key/seed, appropriate independent accreditation should be ensured to mitigate key manipulation risk.	COBIT 2019 DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6 ISO/IEC 27001:2013 A.10.1.2

KM-2: Key Security: Ensuring keys are stored and accessed in a safe and sustainable manner.	**Related Framework Guidance**
KM-2.01: Ensure a secure key/seed backup exists. If the primary key is the only key in existence, then the loss of that key results in a complete loss of cryptoassets. If proper access controls for primary and backup keys are not implemented, an unauthorized party may gain control of keys and compromise funds.	COBIT 2019 APO14.10, DSS04.07, DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6, PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.10.1.2, A.12.3.1

KM-2: Key Security: Ensuring keys are stored and accessed in a safe and sustainable manner.	**Related Framework Guidance**
KM-2.02: Ensure a key compromise protocol (KCP) is in place to prevent and respond to key compromise or potential compromise. KCP should include an inventory of keys, processes and procedures, knowledgeable personnel and authenticated communication channels, assuring an organization is able to respond to a compromise in a timely manner to mitigate the potential loss of assets.	COBIT 2019 BAI03.05, DSS04.07, DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6, PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.10.1.2, A.12.3.1
KM-2.03: Ensure proper keyholder grant-and-revoke policies and procedures are created and implemented. Without these, the onboarding and especially offboarding of staff can result in someone having improper access to keys, resulting in unauthorized movement of assets.	COBIT 2019 DSS04.07, DSS05.04, DSS05.06, DSS06.03, DSS06.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6, PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.10.1.2, A.12.3.1
KM-2.04: Ensure SMS messaging related to key management using two-factor authentications preventing swapping attacks. Often phone carriers do not have good security practices for customer accounts.	COBIT 2019 DSS04.07, DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6, PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.9.2.3, A.10.1.2, A.12.3.1
KM-2.05: Ensure a multisignature scheme is created in an enterprise environment when appropriate. An M of N scheme, when N > M, and two or more signers should be used to move cryptoassets for internal control purposes. If this scheme is not deployed, then custody and authorization can reside with one party, allowing them to potentially misappropriate or inadvertently move cryptoassets.	COBIT 2019 DSS04.07, DSS05.04, DSS05.06 NIST CSF PR.AC-1, PR.AC-3, PR.AC-4, PR.AC-6, PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.9.2.3, A.10.1.2, A.12.3.1

KM-3: Key Storage: Ensuring keys are stored and accessed in a safe and sustainable manner.	**Related Framework Guidance**
KM-3.01: Ensure that key storage and key backups are encrypted when not in use; unencrypted cryptoassets can be compromised/stolen.	COBIT 2019 APO14.10, BAI03.02, DSS01.01, DSS04.07 NIST CSF PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.9.2.3, A.10.1.2, A.12.3.1, A.17.1
KM-3.02: Ensure key/seed backups are stored in a geographically separate location from the primary key. If the primary key and the backup are stored in the same location. An "act of god" or other incidents could result in the loss of both sets of keys at the same time resulting in the loss of cryptoassets.	COBIT 2019 DSS01.01, DSS01.04, DSS01.05, DSS04.07 NIST CSF PR.IP-4, PR.IP-5, RC.IM-2 ISO/IEC 27001:2013 A.9.2.3, A.11.1, A.10.1.2, A.12.3.1, A.17.1
KM-3.03: Ensure that backup key/seed is protected against environmental risk, such as fire, flood, theft and other "acts of god." If the backup is stored on paper rather than stainless steel or titanium, then the key could be lost due to water damage, degradation of the paper and fire, resulting in a loss of assets if the primary key is also lost or compromised.	COBIT 2019 DSS01.01, DSS01.04, DSS01.05, DSS04.07 NIST CSF PR.IP-4, PR.IP-5, RC.IM-2 ISO/IEC 27001:2013 A.9.2.3, A.11.1, A.10.1.2, A.12.3.1, A.17.1
KM-3.04: Ensure that hardware wallets are not deployed in a cold storage scenario with one keyholder, i.e., no multi-sig scheme. Cryptoassets held in cold storage are more susceptible to hack, theft or loss when a hardware wallet is not deployed.	COBIT 2019 DSS01.01, DSS04.07 NIST CSF PR.IP-4, RC.IM-2 ISO/IEC 27001:2013 A.9.2.3, A.10.1.2, A.12.3.1, A.17.1

Blockchain: Smart Contract (SC)—Includes impacted DLT networks, oracle calls and integration code between blockchain recording of state and other transaction data.

SC-1: Governance of Smart Contracts (SC): Governance control objectives are potential regulatory or legal risk inherent in the design and operation of any SC, which may result in compliance or legal actions against the network designers/participants.	**Related Framework Guidance**
SC-1.01: Ensure that the contractual obligations hard coded in the SC code and approved by the consensus mechanism of the network do not violate contract law/regulation in any of the jurisdictions in which the SC operates or may operate—this may include conflicting contract law in multiple jurisdictions.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST CSF ID.GV-3, ID.GV-4 ISO/IEC 27001:2013 A.18.1
SC-1.02: Ensure that cryptocurrencies or token assets transacted by the SC are recognized as legal tender or are not sanctioned in one or more of the jurisdictions in which the smart/contract operates.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST CSF ID.GV-3, ID.GV-4 ISO/IEC 27001:2013 A.18.1
SC-1.03: Establish and document, given the international nature of the SC/network, that it is clear where the home regulatory and legal jurisdiction of the SC/network is and what the required reporting (tax and financial) and regulatory compliance issues are in the home jurisdictions and others in which the network transacts.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST CSF ID.GV-3, ID.GV-4 ISO/IEC 27001:2013 A.18.1
SC-1.04: Establish and document, given the decentralized nature of a private or public network on which SCs operate, who (entity or individual) is responsible for reporting or regulatory compliance/violations.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST CSF ID.GV-3, ID.GV-4 ISO/IEC 27001:2013 A.18.1
SC-1.05: Ensure that oracles used by SCs to providing critical third-party external inputs are secure and appropriately accredited (e.g., via a service organization control [SOC] review of IT general controls) to help assure accuracy and security of the oracle supporting the SC.	COBIT 2019 BAI02.01, MEA03.01, MEA03.04 NIST CSF ID.GV-3, ID.GV-4 ISO/IEC 27001:2013 A.18.1
SC-1.06: Ensure that the SC transactions or outputs to the network are appropriately protect-and-secure and that personally identifiable information (PII) that is regulated under privacy laws around the world (e.g., GDPR, CCPA, etc.) conforms with all privacy requirements under applicable law for all jurisdictions where the blockchain/DLT network resides and participants transact.	COBIT 2019 APO13.03, BAI02.01, DSS05.03, MEA03.01, MEA03.04 NIST CSF ID.GV-3, ID.GV-4 ISO/IEC 27001:2013 A.18.1

SC-2: Design Risk is potential ubiquitous risk inherent in the design of any SC, which may result in compromises of the intended SC function and may include risk applicable to any software code.	**Related Framework Guidance**
SC-2.01: Ensure all SCs are operating on the network and test that smart contracts have adequate circuit breaker/self-destruct capabilities when things go wrong, resulting in large or undetected losses/liability.	COBIT 2019 EDM03.03, APO04.06, APO12.03, BAI02.01, DSS06.01, MEA03.01 NIST CSF ID.GV-4 ISO/IEC 27001:2013 A.17.1

SC-2: Design Risk is potential ubiquitous risk inherent in the design of any SC, which may result in compromises of the intended SC function and may include risk applicable to any software code.	**Related Framework Guidance**
SC-2.02: Ensure all SCs that are approved by the consensus mechanism and operating on the network have adequate transaction limits (high or low) to help prevent the potential of large or undetected losses/liability.	COBIT 2019 APO04.06, BAI02.01, DSS06.01, MEA03.01 NIST CSF ID.GV-4 ISO/IEC 27001:2013 A.17.1
SC-2.03: Ensure that the consensus-approved change management process for adding or upgrading smart contracts prevents wrong or malicious code or sequencing of the SC, resulting in contract performance or network liability damage to the nodes or network participants.	COBIT 2019 DSS05.01 NIST CSF DE.CM-4 ISO/IEC 27001:2013 A.12.2
SC-2.04: Ensure via continuous independent review/validation that there are not design errors in time lock/lock time coding or execution, which may result in financial loss or liability due to the premature or latent transactions.	COBIT 2019 EDM02.01, APO11.04, BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1

SC-3: External/Integration Dependencies of Smart Contracts: This is risk where the SC interaction with other SCs (including itself being run before the first version finishes), data, oracles or other processes or network features creates unforeseen vulnerabilities, resulting in a compromise of the SC and the network (e.g., reentrancy and delegate call injection).	**Related Framework Guidance**
SC-3.01: Mitigate reentrancy risk, which may result from many types of program calls in blockchains. A key vulnerability involves functions that can be called repeatedly, before the first invocation of the function is finished. This may cause the different invocations of the function to interact in destructive ways, creating smart contract vulnerabilities, which may be exploited by internal or external bad actors.	COBIT 2019 BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1
SC-3.02: Mitigate race conditions, risk form of reentrancy which creates risks when independent smart contracts (code) which may be secure running alone creating unforeseen anomalies when run simultaneously.	COBIT 2019 BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1
SC-3.03: Ensure that delegate-call injections cannot occur – this is where a malicious callee contract can directly modify (or manipulate) the state variables of the caller contract. This vulnerability is caused by the ability of a state variable of a caller contract updating the bytecode of a callee contract.	COBIT 2019 BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1
SC-3.04: Ensure that DoS with unexpected revert cannot occur when a transaction is reverted due to a caller contract encountering a failure in an external call, or the callee contract deliberately performs the revert operation to disrupt the execution of the caller contract. This vulnerability is caused by the execution of a caller contract being reverted by a callee contract, resulting in value loss or nonperformance of a contract.	COBIT 2019 BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1

SC-3: External/Integration Dependencies of Smart Contracts: This is risk where the SC interaction with other SCs (including itself being run before the first version finishes), data, oracles or other processes or network features creates unforeseen vulnerabilities, resulting in a compromise of the SC and the network (e.g., reentrancy and delegate call injection).	**Related Framework Guidance**
SC-3.05: Ensure that a return value is always validated and checked. This vulnerability may happen when call exceptions are mishandled (unintentionally or maliciously) resulting in erroneous or unintended transactions, causing contract performance or financial loss to the network or users.	COBIT 2019 BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1
SC-3.06: Ensure appropriate access controls are in place such that the smart contract author/owner's address cannot be compromised by calling the vulnerable smart contract function to take over the ownership.	COBIT 2019 BAI03.03, BAI03.08, BAI03.10 NIST CSF DE.CM-8 ISO/IEC 27001:2013 A.14.1

SC-4: Manipulation/Denial of Service related to smart contracts. These may include any potential SC vulnerabilities originating from within or outside the network that might manipulate data on the network or restrict a node or user access to the network.	**Related Framework Guidance**
SC-4.01: Ensure that all SCs gas (ether) are deposited to their account on the network to transact with cryptoassets. This legitimate resource for running smart contracts should never be frozen (i.e., frozen ether). There should be adequate functionality, security, processes and controls in place to transact frictionless, without their resources on the network instead relying on the money-spending function of another contract (as a library) resulting in accidental or deliberate inability to perform on a contract or obligation.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS05.02, MEA03.01 NIST CSF ID.GV-3, PR.AC-5, PR.DS-3, PR.DS-4 ISO/IEC 27001:2013 A.8.2.3, A.18.1
SC-4.02: Prevent balance manipulation when ether/gas are forced to smart contracts when a contract's control-flow decision relies on the value of a particular balance or address, which can be leveraged by an attacker to cause (for example) that only the attacker can obtain the money, because the contract's balance is in a condition check the attacker sees, resulting in fraud or abuse.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS05.02, MEA03.01 NIST CSF ID.GV-3, ID.AM-3, PR.AC-5, PR.DS-3, PR.DS-4 ISO/IEC 27001:2013 A.8.2.3, A.13.2, A.18.1
SC-4.03: Ensure that the visibility of a function is correctly specified (e.g., secured as private or internal with appropriate encryption) preventing unauthorized access. Some blockchain languages make all functions public by default, which allows attackers to directly call these improperly specified functions.	COBIT 2019 APO13.01, BAI02.01,BAI04.04, BAI09.03, DSS01.04, MEA03.01 NIST CSF ID.GV-3,PR.AC-3, PR.AC-5, PR.DS-3, PR.DS-4 ISO/IEC 27001:2013 A.8.2.3, A.18.1
SC-4.04: Ensure that a smart contract's funds can only be withdrawn by an authorized caller, who deposited funds to the contract. This may be caused by a failure in checking a caller's identity when the caller invokes a function to send ether to an arbitrary address.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS05.02, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.AC-1, PR.AC-5 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1

SC-4: Manipulation/Denial of Service related to smart contracts. These may include any potential SC vulnerabilities originating from within or outside the network that might manipulate data on the network or restrict a node or user access to the network.	**Related Framework Guidance**
SC-4.05: Processes should be in place to prevent suppression attacks where an attacker sends multiple transactions with a high gas price and gas limit to custom smart contracts that assert (or use other means) to consume all the gas and fill up the block's gas limit — this stops or delays others from running transactions, resulting in potential loss or abuse.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS01.04, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.IP-1, PR.IP-5, PR.IP-7 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1
SC-4.06: Ensure that in any type of bidding or group payment transactions, a malicious bidder cannot throw or take over the transactions resulting in and incomplete transaction with no one getting paid, refunded or winning a bid.	COBIT 2019 APO13.01, BAI02.01 BAI04.04, BAI09.03, DSS01.04, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.IP-1, PR.IP-5, PR.IP-7 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1
SC-4.07: Ensure procedures are in place to mitigate DoS Gas limit manipulation abuse. Each block has an upper bound on the amount of gas that can be spent, and thus the amount computation that can be done. If the gas spent exceeds this limit, the transaction will fail. This leads to possible Denial of Service vectors: Also, excessive use of gas by a single node or user to executing smart contracts may be an indication of fraud or abuse on the network.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS01.04, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.IP-1, PR.IP-5, PR.IP-7 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1
SC-4.08: Ensure that variants related to transactions or smart contracts where transactions/contracts are trying to perform competing transactions (one is canceling an order while another is trying to fulfill the same order - asymmetric) or one transaction/contact is running a large bulk transaction while another is trying to procure a small part of that bulk processing. The results of these conflicts can be delayed, canceled or incomplete contracts.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS01.04, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.IP-1, PR.IP-5, PR.IP-7 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1
SC-4.09: Ensure privacy on blockchains with control activities that prevent transaction combinations or smart contracts containing nefarious information (i.e., PII or other privacy information) is never able to be put on blockchain resulting in violation of compliance standards (i.e., GDPR, CA Privacy, etc.) resulting in damage or the destruction of the network due to the need to hard fork to remove the information.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS01.04, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.IP-1, PR.IP-5, PR.IP-7 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1
SC-4.10: Ensure processes are in place to prevent frontrunning on blockchains/smart contracts by miners or administrators having insider knowledge of block or/transactions before they are submitted to the ledger. This can be troublesome for things like decentralized markets, where a transaction to buy some tokens can be seen, and a market order implemented before the other transaction gets included, resulting in losses related to insider trading type activities.	COBIT 2019 APO13.01, BAI02.01, BAI04.04, BAI09.03, DSS01.04, DSS05.04, MEA03.01 NIST CSF ID.GV-3, PR.IP-1, PR.IP-5, PR.IP-7 ISO/IEC 27001:2013 A.8.2.3, A.9.2, A.18.1

Page intentionally left blank

APPENDIX B
Glossary

TERM	DEFINITION
Altcoin	Have no formal definition but are widely considered to be alternative digital currencies; can also be all cryptocurrencies other than bitcoin
Application programming interface (API)	A set of routines, protocols and tools referred to as building blocks used in business application software development
Atomic	A condition of smart contracts in that one or more conditions defined by the smart contract must all be met for the transaction to execute in its entirety
Atomic swaps	Peer-to-peer exchange of assets across separate blockchains triggered by predetermined rules, without the use of a third-party service, through the use of self-enforced smart contracts. Requires an exchange of assets on both sides or transaction will not occur
Blockchain	A distributed, protected journaling and ledger system. Use of blockchain technologies can enable anything from digital currency (e.g., Bitcoin) to any other value-bearing transaction
Channels	Also known as ledger conduits, are private channels in a permissioned blockchain network, in which two or more nodes perform private transactions
Checkpointing	The process of storing a block in the history of the blockchain at intervals and refusing to accept divergent blockchain without these blocks
Confirmation	The number of blocks added to the blockchain after the network has accepted that a particular transaction has been executed
Consensus mechanism	A fault-tolerant mechanism used in blockchain/distributed ledger systems to achieve the necessary agreement on data values or the state of the network among distributed processes or multiagent systems
Contract account	The account (or address) created when a smart contract is deployed by the smart contract owner. Contract account contains the runtime virtual machine bytecode for a contract.
Cross chain	Interoperability between two independent blockchains; allows for blockchains to speak to each another; accomplished mainly by an asset swap or asset transfer

TERM	DEFINITION
Cryptocurrency	A digital asset designed and created to function as a unit of account and payment method within its particular ecosystem. Cryptocurrency transactions usually take place within a peer-to-peer network and use cryptography to secure transaction records.
Cryptotoken	A cryptotoken, which can also be considered a cryptoasset, is the unit for any blockchain ecosystem that is used for any function not related to payments within that blockchain; for example, as a function of a decentralized application or a smart contract. Security tokens or utility tokens are examples of cryptotokens.
Digital asset	Any token, whether created in a peer-to-peer and/or cryptographic environment, that exists in a digital format and comes with the ability and right of the token holder to use or transfer the digital asset; all cryptocurrencies and cryptotokens are subsets of digital assets.
Distributed denial-of-service attack (DDoS)	A denial-of-service (DoS) assault from multiple sources.
Double spending	A potential flaw in blockchain where the native digital token or currency can be spent more than once.
Externally owned account	Externally owned account (EOA) is an address that is generated from a user's public key. EOA is typically owned by an individual.
Gas	A unit/fee that measures the amount of computational effort required to execute certain operations related to a function or smart contract on a blockchain. Best known in relation to the Ethereum blockchain/network.
Gas fee	Relevant to the Ethereum blockchain in particular, gas references the cost required to process a transaction on the network. Miners in this instance can set the price of gas and can decline to process a transaction if it does not meet a price threshold that they determine.
Hard fork	A change to blockchain software, such that any nodes validating according to the old software, will see all blocks produced subsequent to the new software as invalid. For blockchain nodes to work in alignment with the new software, each will be required to upgrade. If a group of nodes do not upgrade and perpetuate use of the old version of software, a permanent split in the blockchain can occur.
Hash power	The individual hash power contributed by a single miner or worker to the PoW hash rate
Hash rate	PoW blockchain network measures the security profile using the total hash rate provided by all full nodes in supporting consensus algorithm. Generally, the higher the total hash rate the more secure the PoW blockchain network.

TERM	DEFINITION
Hybrid blockchain	A blockchain that attempts to use optimal parts of private and public blockchain solutions; hybrid blockchains are not open to all parties, but still maintain immutability, transparency and integrity features of public chains.
Interoperability	The ability to exchange, access and make use of information across different systems and/or networks without the need for intermediaries. The capacity to transfer an asset between two or more networks or systems without changing the state of the asset.
Mutex	A lock that sets by the smart contract code before using a shared resource or function, and release that after using it. When locked, the lock prevents no other threads can gain access to the locked region of the code.
Network Topology	The basic configuration and architecture of a set of interconnected nodes
Network Interoperability	The ability for networks comprised of different topologies, configurations and functionalities to send and receive data between each other.
Node	Point at which terminals are given access to a network.
Offchain	Offchain refers to any blockchain actions that require data outside of the blockchain network.
Onchain	Onchain transactions refer to those cryptoasset or token transactions which occur on and within the data records of a blockchain and are perpetually dependent on the state of that blockchain for their validity.
Oracle	A relational-database programming system incorporating the SQL programming language. A registered trademark of the Oracle Corp.
Oracle problem	The oracle problem describes a paradoxical situation where the oracle can become the central point of failure for the smart contract due to decreasing security and centralization.

TERM	DEFINITION
Practical byzantine fault tolerance (pBFT)	Consensus mechanism in which all nodes are ordered in sequence with one node being primary node or leader, and all others referred to as backup nodes. All nodes in pBFT systems communicate with one another with the goal being that all honest nodes will come to an agreement of the state of the system using a majority rule. Nodes communicate for two reasons: to prove that messages came from a specific peer node, and to confirm that that message was not modified during transmission; pBFT can be used for private and public blockchains and allows for instant transaction finality; however, such methodology requires a great number of messages between nodes, hence making a large blockchain network challenging.
Private blockchain	A blockchain system in which all physical and digital assets are owned by one entity, group or permissioned participants
Proof of elapsed time (PoET)	PoET is a consensus mechanism algorithm that is often used on the permissioned blockchain networks to randomly decide the next block publisher. The selection process works as follow: • Each publisher waits for a random amount of time. • The first publisher to finish the waiting gets to be the leader for the new block. Intel Software Guard Extensions (SGX). SGX allows applications to run trusted code in a protected environment. For PoET, the trusted code is what ensures that these two requirements are satisfied—keeping the lottery fair.
Proof of stake (PoS)	Proof of stake is a type of consensus algorithm by which a cryptocurrency blockchain network aims to achieve distributed consensus. In PoS consensus, the creator of the next block of data is chosen via several combinations of random selection and wealth or age (i.e., the stake) within the blockchain; With PoS, miners can mine or validate block transactions based on amount of cryptocurrency a miner holds; was created as an alternative to PoW, which requires large amounts of energy; PoS gives mining power based on the percentage of cryptocurrency held by a miner; seen as less risky in terms of network attacks and security and used only for public blockchains.
Proof of Work (PoW)	PoW is conducted through miners (participants who keep the blockchain running by providing computing resources), who are competing to solve a cryptographic problem (hash puzzle). The PoW algorithm is used to confirm transactions and produce new blocks which are added to the chain. With PoW, miners compete against each other to complete transactions on the network and get rewarded. The computational work required to accomplish this is fairly (and usually increasingly) difficult for miners to perform, but easy for the network to verify. As difficulty increases over time, the amount of computational power, and, hence, energy consumption, grows. Bitcoin is the first widespread application use of PoW. PoW is applicable to public blockchains.

TERM	DEFINITION
Protocol code	Cryptographically secure code prescribing strict adherence to the design and functioning of blockchains/distributed networks. This code can only be expanded or modified with approval from the network consensus mechanism.
Public blockchain	A blockchain system in which physical and digital assets are decentralized, zero-trust based, and hosted/maintained on ephemeral networks and nodes
Security token	Digital assets or tokens created to represent a quantity of a specified investment, including rights to ownership, payment of a specific sum under a contract, entitlement to future profits, etc.
Smart contract	Software (computer code) that automatically executes transactions and/or enforces agreements based on the fulfillment of the terms of the agreement by leveraging decentralized ledger technology that uses public validation to ensure correct and reliable performance according to agreed rules.
Soft fork	A software upgrade that is backward compatible with previous versions of the blockchain software. Thus, a soft fork does not require all blockchain nodes to upgrade to maintain functionality.
Turing-complete	A computational term meant to describe a system that can successfully be used as a Turing Machine, i.e., a system whose programming language can simulate what another programming language can accomplish
Utility token	Digital assets or tokens created and utilized to finance creation of a network by providing its buyers with a pledge of being able to use some of the network ecosystem or products; do not give any legal or economic right of ownership over the developer nor any part of the ecosystem
Wallet	An application or other service that gives holders of cryptocurrency the ability to store and retrieve their digital assets. Such wallets come in many forms, including hot wallets (any wallet application or service connected to the Internet), or cold wallets (or cold storage, which are often hardware devices that can be disconnected from the Internet or other electronic services).
Zero-knowledge proof	A critical aspect of cryptography, this is a method by which one party (Party A) is able to prove to another party (Party B) that Party A is aware of the value of a specific variable, without conveying any additional information about that variable, other than that they know its value

Page intentionally left blank