



RISKS AND OPPORTUNITIES FOR SYSTEMS USING BLOCKCHAIN AND SMART CONTRACTS

May 2017



CITATION

Staples, M., Chen, S., Falamaki, S., Ponomarev, A., Rimba, P., Tran, A. B., Weber, I., Xu, X., Zhu, J., (2017) Risks and opportunities for systems using blockchain and smart contracts. Data61 (CSIRO), Sydney.

COPYRIGHT

© Commonwealth Scientific and Industrial Research Organisation 2017. To the extent permitted by law, all rights are reserved and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

IMPORTANT DISCLAIMER

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

CSIRO is committed to providing web accessible content wherever possible. If you are having difficulties with accessing this document please contact csiroenquiries@csiro.au.

ACKNOWLEDGEMENTS

This project is funded by the Australian Government's National Innovation Science Agenda, and with the assistance of The Treasury.

We thank the employees from The Treasury and from Data61 (CSIRO) who provided vital assistance in the organisation and operation of the project.

We are grateful for the participation of Emma Weston from AgriDigital for input into the preparation of the agricultural supply chain sidebar in this report.

We have received helpful feedback on earlier drafts of this report from anonymous reviewers, and from the following reviewers.

Greg Adamson, Digital Risk Innovation
Nick Cliff, Australian Payments Network
David Emery, Reserve Bank of Australia
Scott Farrell, King & Wood Mallesons
Vincent Gramoli, University of Sydney
Ralph Holz, University of Sydney
Carla Hoorweg, Financial Services Council
Zoran Milosevic, Deontik
Heshan Peiris, ANZ
Robert Porter, ANZ
Amanda Scotney, Australian Tax Office
Tilly South, CHOICE
Chris T'en, ANZ
Leo Zhang, Sydney Stock Exchange



EXECUTIVE SUMMARY

Blockchain technologies originally emerged to support new forms of digital currency, but now hold promise as a new foundation for transactions in society. A blockchain is both a database recording transactions between parties, and also a computational platform to execute small programs (called ‘smart contracts’) as transactions. A blockchain is a distributed database, replicated across many locations and operated jointly by a collective. Blockchains transactions can support services for payments, escrow, notarisation, voting, registration, and process coordination. These are key in the operation of government and industry. Conventionally, these services are provided by specific trusted third-parties such as banks, legal firms, accountancy firms, government agencies, and service providers in specific industries. With a blockchain-based system, rather than relying on third-party organisations, we could instead choose to rely on the blockchain software and on a majority of the collective that jointly operates the blockchain system.

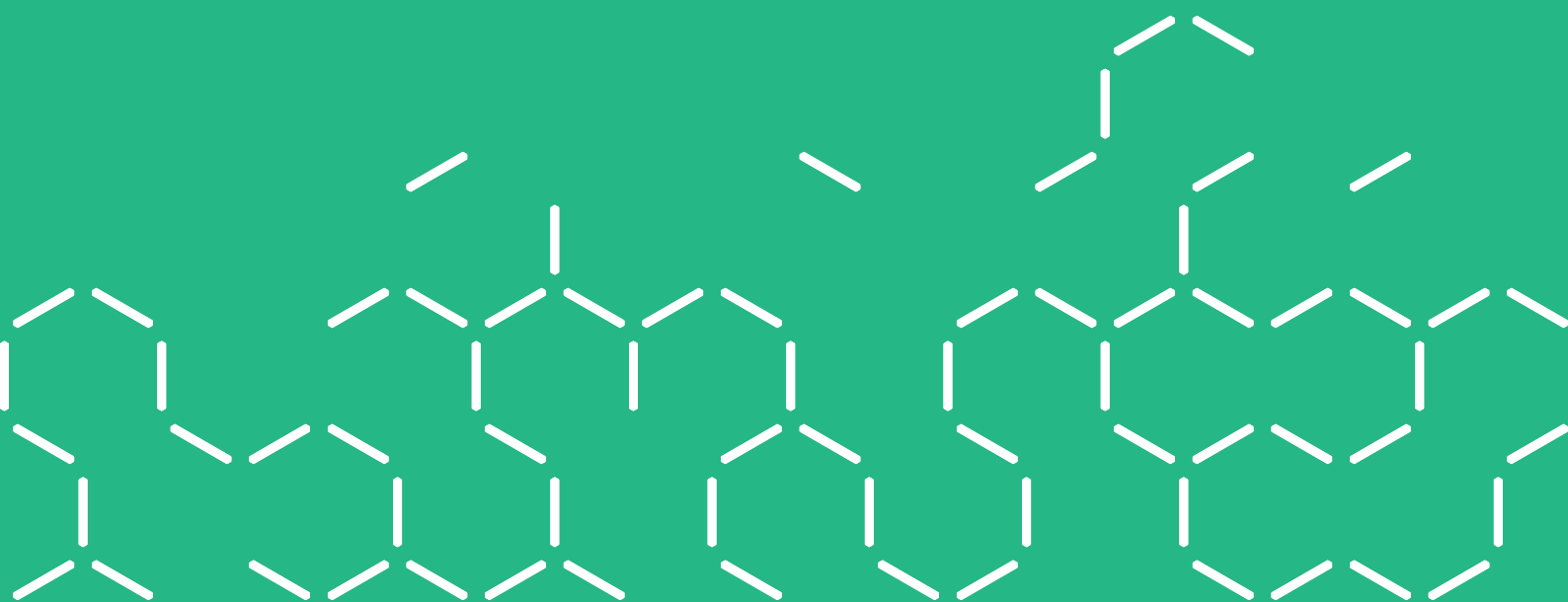
The report describes some of the technical risks and opportunities in the application of blockchain technologies within government and industry, and how to assess whether blockchain-based systems will meet critical requirements. The project explores this primarily through the description and analysis of high-level design alternatives for illustrative ‘use cases’. Three use cases have been selected after a number of initial workshops and preliminary research: remittance payments, open data registries, and agricultural supply chain. These provide reasonable coverage of various kinds of requirements and regulatory concerns, against which we can evaluate design alternatives, and in turn learn more general lessons about blockchain technologies. In addition to this design-based analysis, we also report on some empirical results from testing prototype implementations.

Compared to conventional centralised databases and computational platforms (on-premises or cloud), blockchains can reduce some counter-party and operational risks by providing neutral ground between organisations. Blockchain technologies may provide advantages for integrity and non-repudiation. However, they also currently have limitations for confidentiality, privacy, and scalability. For latency and availability, reading is improved but writing is worsened. Blockchains are also subject to a different cost model. Digital currency transfer and long-term storage of transactional data may be less expensive. However, program execution and storage of big data may be more expensive.

Public blockchains provide very low barriers to entry for new participants, which can facilitate competition, innovation, and productivity. However, they do not mandate authentication of those participants, which creates challenges for regulation of money laundering, terrorism financing, and tax avoidance. Private blockchains can impose more controls on authentication and access, which can partly address those regulatory concerns. Still, for competitors within an industry consortium, private blockchains may not be private enough to provide normal levels of commercial confidentiality for business operations, competitive position, and customer relationships.

When assessing business risk, regulatory acceptance, and assurance arguments for a blockchain-based system, we need to consider not just the blockchain, but also all of the other components that are integrated in the design of the whole system. Other components will provide user interfaces, cryptographic key management, and off-chain databases, communications, and processing. Judicious use of these other components may mitigate blockchain’s risks while still leveraging blockchain’s opportunities.

Finally, blockchains are still a rapidly evolving technology, with ongoing developments especially to improve scalability and confidentiality. Globally, governments, enterprises, and startups are exploring the technology/market fit in a wide variety of use cases and for a wide variety of requirements and regulatory demands. There is still much that is unknown about the development of trustworthy blockchain-based systems. Further research is required to improve our knowledge about how to create blockchain-based systems that work, and how to create evidence that blockchain-based systems will work as required.



CONTENTS



Executive summary	i
Part I Background	1
1 Blockchains and smart contracts	2
1.1 Blockchains.....	2
1.2 Smart contracts.....	4
1.3 Uses in industry and society	4
2 Software architecture and dependable systems.....	7
2.1 Non-functional properties and requirements.....	7
2.2 Software architecture – design and analysis.....	8
2.3 Dependable software systems.....	8
Part II Use case studies	9
3 Study objectives and approach	10
3.1 High-level approach	10
3.2 Criteria for selection of use cases	10
4 Design and analysis of blockchain-based systems.....	11
4.1 Use case 1: Supply chain	11
4.2 Use case 2: Registry	20
4.3 Use case 3: Payments.....	25
Part III Discussion.....	31
5 Risks and opportunities for blockchain-based systems	32
5.1 Blockchain myths	32
5.2 Working within blockchain’s limitations	32
5.3 Distinctive opportunities	34
6 Design and assurance of blockchain-based systems	36
6.1 Design.....	36
6.2 Assurance: Evidence and acceptance.....	38
6.3 Non-functional properties.....	40
7 Limitations of this study	45
Appendix A Findings and recommendations.....	46
Shortened forms.....	48
Glossary of terms.....	49
References.....	52





PART I BACKGROUND

Blockchains and the design of blockchain-based systems



1 BLOCKCHAINS AND SMART CONTRACTS

1.1 Blockchains

Blockchains are a digital technology that combine cryptographic, data management, networking, and incentive mechanisms to support the checking, execution, and recording of transactions between parties.

A blockchain ledger is a list ('chain') of groups ('blocks') of transactions. Parties proposing a transaction may add it to a pool of transactions intended to be recorded on the ledger. Processing nodes within that blockchain community take some of those transactions, check their integrity, and record them in new blocks on the ledger. The contents of the blockchain ledger are replicated across many geographically-distributed processing nodes. These processing nodes jointly operate the blockchain system, without the central control of any single trusted third party. Nonetheless, the blockchain system ensures that all nodes eventually achieve consensus about the integrity and shared contents of the blockchain ledger.

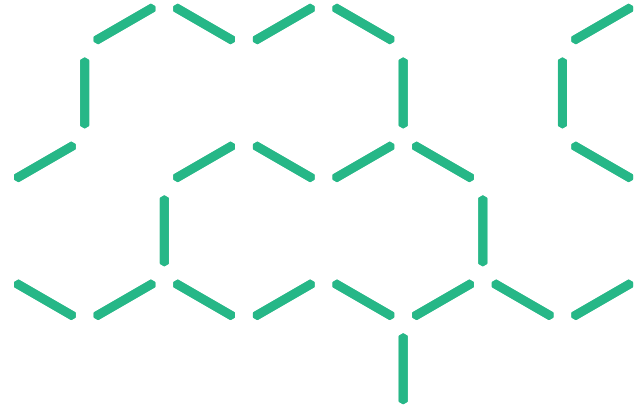
Transactions between parties such as payments, escrow, notarisation, voting, registration, and process coordination are key in the operations of government and industry. Traditionally, these transactions are supported by trusted third-parties such as government agencies, banks, legal firms, accounting firms, and service providers in specific industries. Blockchains provide a different way to support these transactions. Instead of trusting third-parties, we would trust a majority of the collective jointly operating the blockchain, and the correctness of their shared technology platform.

Blockchains were originally used for the Bitcoin [11] digital currency¹, but are now being implemented in many other platforms, and used for many other purposes. Just like a traditional database, a blockchain can in principle be used to represent transactions or information in any kind of organisation in industry or society. Nonetheless, blockchains are different from traditional databases in important ways, and the full range of technical, organisational, and societal consequences of these differences are not yet well understood.

There are several kinds of blockchains, and to provide more general insights in this project we take a broad view. For example, the Bitcoin system is a 'public blockchain', which allows unfettered public participation in both its operation and use. Other well-known systems, such as the Ethereum [16] blockchain, are similar in this regard. It is possible to use a separate instantiation of the Bitcoin or Ethereum computer programs to operate a blockchain within a private context, for example on a virtual private network. These would then be one kind of 'private blockchain'. Private networks and private computer systems allow strong access controls. This provides greater administrative control for private blockchains. However, the software for public blockchains is not always the best technical solution to use in a private setting. Many industry consortia, such as Hyperledger, R3CEV, and Ripple, are actively developing specialised private blockchain solutions. These typically support a smaller number of processing nodes than public blockchain solutions, but can provide improved security and performance. When a group of companies or organisations jointly create a private blockchain, this is sometimes called a 'consortium blockchain'.

Some authors distinguish between blockchain technology and 'distributed ledger technology' (DLT). A distributed ledger is in some ways a more abstract notion, capturing a purpose for use: the distributed replication of auditable logs of transactions, shared between parties of interest. While public or private blockchain technologies can be used to implement a distributed ledger, there are alternative technological approaches which could be used instead. For example, the Corda system [4] implements distributed ledgers between parties, but unlike most blockchain systems does not have a global ledger that is independently checkable by all processing nodes. Nonetheless in this report, unless otherwise specified, we use the term 'blockchain' to include public blockchains, private blockchains, and other kinds of DLT.

¹ What we call a digital currency is also variously known as cryptocurrency, cryptocurrency, cybercurrency, and virtual currency. A digital currency is a digitally communicable form of money which may be of a state-issued fiat currency, or a new unit created by non-state actors. Bitcoin is one example of the latter. Many blockchains implement or rely on a digital currency, but in principle digital currencies can operate without a blockchain, and some private blockchains operate without a digital currency.



The successful operation of a blockchain system relies on several key elements, including:

- appropriate integrity criteria to be checked for each transaction (and block);
- the correctness of the system's software and technical protocols;
- strong cryptographic mechanisms to identify² parties and check their authority to add new transactions; and
- a suite of incentive mechanisms to motivate processing nodes to participate in the community and to behave honestly, in its interests.

Blockchain systems can be different in various ways, including:

Admittance of processing nodes: In a *public* blockchain system, such as Bitcoin, anyone may become a processing node (sometimes called a 'miner'). In a *permissioned* (private) blockchain system, the admittance of processing nodes is controlled by its governing bodies.

Consensus mechanism: Most public blockchains use *Nakamoto consensus*, where processing nodes by convention treat the longest history of blocks as the authoritative history. The rate at which blocks can be created is limited, often by using a *proof of work* mechanism, whereby a processing node can only add a new block by demonstrating that a difficult task has been completed. Proof of work is widely used, but the auxiliary effort required to complete the difficult task can be economically inefficient. In a *proof of stake* system, the processing node that can add a new block in the next round is determined by the size of its stakeholding in the global blockchain and/or in that round. Proof of stake can be more efficient, but is more recent and has not yet been widely adopted. Other consensus mechanisms have been proposed. On private blockchains, conventional replication algorithms such as practical Byzantine fault tolerance can be used instead of Nakamoto consensus. This can provide stronger guarantees about the completion of transactions, and may be more performant, but only support a smaller number of processing nodes which must be more trusted.

Representation of transactions: A distributed ledger may record financial transactions, such as in Bitcoin. However, a distributed ledger may be thought of as a shared database, and might allow any other kind of data to be recorded. In particular, the data recorded for a transaction may be the text of a computer program, and the integrity check for that transaction may involve executing that program. This allows participants to create 'smart contracts', to be discussed below. A blockchain transaction is not appropriate for all data – because it is replicated globally, transactions should not contain very large data, nor plaintext data which must be kept confidential. So, there is a choice about what data should be stored 'on chain' inside transactions, or 'off chain', in external systems. However, even if static data is stored off-chain, the blockchain can nonetheless record a cryptographic hash of that data to allow its integrity to be checked.

The rate at which blocks can be created is limited, often by using a proof of work mechanism, whereby a processing node can only add a new block by demonstrating that a difficult task has been completed.

² 'Identity' here refers to an identifier for the authorisation of transacting participant in the blockchain system, but does not also include the authentication of that participant's real-world identities by governments or other authorities. This is discussed further below.

1.2 Smart contracts

The transactions stored on a blockchain can be more than simple records of the exchange of assets – some blockchain systems also allow computer programs to execute and be stored as part of transactions on the ledger. These are often called ‘smart contracts’, although the programs are typically not very ‘smart’, and are sometimes not used to execute or monitor legal contracts.³

The legal status of smart contracts as legal contracts is currently debated. A legal contract is an agreement between parties, and a computer program is either the text of source code or an executing physical machine. So smart contracts, as computer programs, may be the wrong category of thing to be a legal contract. Nonetheless a smart contract may provide evidence for there being a legal contract, and may be able to facilitate the execution of a legal contract. Importantly as a mechanism for the execution of provisions of a legal contract, smart contracts can carry and conditionally-transfer digital currency and other digital assets or tokens between parties. This can be done in a predictable and transparent way on the neutral ground provided by the mechanised infrastructure of a blockchain.

The Bitcoin blockchain allows very simple forms of smart contracts, but other blockchains such as Ethereum allow computer programs to be written in a ‘Turing complete’ language that is in principle as expressive as every other general purpose programming language. As a result, blockchains can be more than a simple distributed database – they can be general computational platforms. (Albeit currently with severe practical limitations on computational complexity.) This capability significantly expands the power of blockchain systems, and increases their range of use and potential for innovation. Some blockchains eschew the use of Turing-complete smart contract languages, in order to facilitate the automated verification of the correctness of smart contracts.

1.3 Uses in industry and society

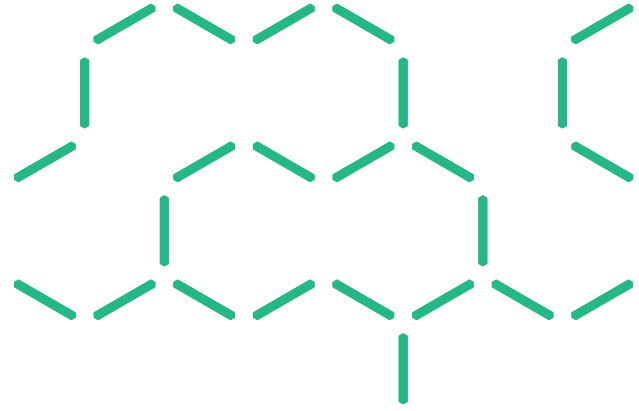
Bitcoin has been operational since 2009, and its digital currency has a total market value of about USD\$22B as of the end-of April 2017. The next-largest blockchain, Ethereum, has a value of about USD\$6.3B, and there are many other small public blockchains with their own digital currencies. Private blockchains are increasingly deployed inside large enterprises and across industry consortia. The adoption of blockchain technologies is still in its infancy. Globally, many financial services companies, governments, enterprises and startups are exploring the applicability of blockchain technologies in their domains. New businesses and business models are expected to arise, but as yet there are very few examples of significant use in production of blockchain systems within industries or government.

Blockchains, particularly public blockchains, offer opportunities for disruptive innovation. As discussed earlier, blockchains may disintermediate trusted third-party organisations, thus disrupting conventional business arrangements across society. In economies where trusted third-parties are not always trustworthy, a significant benefit of blockchain systems may be in the strong support they can provide for immutability and non-repudiation. In developed societies, trusted third-party organisations are usually trustworthy, so the benefits of using blockchain technologies would likely arise from enabling faster business model innovation, reducing the cost of establishing business relationships, and perhaps reducing the cost or risk of transactions.



The adoption of blockchain technologies is still in its infancy.

³ The concept of ‘smart contract’ is more general [13] than its use in blockchain, and there are many applications for executable transactions on blockchain other than as contracts. So, the term ‘smart contract’ is far from ideal, but is nonetheless used in this report because its usage is so widespread. Alternative proposed terminology has included ‘chain code’, and ‘automated contract tools’.



FINANCIAL SERVICES

Financial services applications using blockchain technology may include:

Digital currency: new forms of money can be implemented on blockchains, but also a foundation for incentive models that support integrity for many blockchain systems. Blockchains allow digital currency to be transferred between parties, often without those transfers being processed or recorded by banks or payment services. With smart contracts, blockchains may be able to support new kinds of ‘programmable money’, where automatically-enforced policies are attached to specific parcels of currency.

(International) payments: often via digital currency on a blockchain, with local exchanges between the digital currency and fiat currencies.

Reconciliation for correspondent banking: so that reciprocal nostro/vostro accounts held between two banks are replaced by a single shared ledger.

Securities registration, clearing and settlement: where the joint exchange of payment and security holdings are enacted as a transaction on a blockchain.

Markets: smart contracts on blockchains can provide a platform for making and accepting offers to trade assets or services. The blockchain will record the status of these trade offers. Individual smart contracts could themselves carry the digital currency required to be paid on fulfilment of these offers. This functions as a kind of escrow, without the need for a trusted third party organisation. However, blockchains are not suitable for high-frequency (low latency) market trading.

Trade finance: where the blockchain is used to evidence trade-related documents in order to reduce lending risk and improve access to finance for industry, and where smart contracts could control inter-organisational process execution, and transparently automate delayed or instalment payments.

Services such as international payments have regulatory requirements to establish the identity of participants, as part of Anti-Money Laundering (AML) and Counter-Terrorism Financing (CTF) policies. So, identity on blockchain is sometimes considered to be a key enabler for many financial services on blockchain. From a purely technical perspective, real-world identities are not necessarily required. For example on Bitcoin, transacting agents (which are not necessarily persons) are only pseudonymously identified, with a cryptographic key. Therefore international exchange of the Bitcoin digital currency can be performed without establishing real-world identity. Nonetheless, AML/CTF requirements are not obviated by the use of a blockchain. Privacy and confidentiality can be a challenge when integrating identity information into a blockchain-based system.



GOVERNMENT SERVICES

It is now widely recognised that there are many promising application areas for blockchain technology beyond financial services. Blockchains could target improved government service delivery, and private blockchains could be used to facilitate information sharing and process coordination across agencies within government. Application areas being explored in governments globally include:

Registries and identity: including the identities and attributes of persons, companies, or devices; licensing; qualifications; and certifications. Storing registry entries or cryptographic certification of registry entries on a blockchain can facilitate access to and validation against the registry. Blockchains could be used to share authenticated identifiers for individuals and companies, and these identifiers could in turn also enable many other blockchain applications. However, there are complex considerations about privacy and confidentiality.

Grants and social security: smart contracts could automate the process coordination to apply for, decide on, and distribute payments for grants and social security. With a sufficiently-sophisticated payment environment, a smart contract could automatically limit payments to approved suppliers or categories of expenses.

Quota management: Government-granted quotas, allocations, and rights to physical resources could be awarded and tracked through tokens established on a blockchain. Where policy allows, the blockchain could support an independent secondary market for these rights. The blockchain creates an ongoing immutable audit log of these rights and their use.

Taxation: ranging from automated collection of tax using smart contracts through to improved compliance by authoritative publication of taxation regulation and calculation tools as smart contracts on blockchain.



ENTERPRISE AND INDUSTRY

The full potential of blockchain technology is likely to be realised outside financial services and government. Blockchains are a foundational horizontal platform technology that could be used in any industrial sector including agriculture, utilities, mining, manufacturing, retail, transport, tourism, education, media, healthcare, and the sharing/P2P economy. Generic applications in these sectors include:

Supply chain: tracking physical assets through changes in ownership and handling can be recorded and communicated through data stored on a blockchain. This implicitly creates provenance information for goods, and provides improved logistics visibility and supply chain quality. Key events within the supply chain could be linked to automatic payments with the use of smart contracts.

Internet of Things (IoT) storage, compute, and management: devices connected to the internet can use the blockchain as a persistent and highly-available storage solution, can use smart contracts to provide a global distributed computing capability, and can rely on the blockchain as a secure channel for receiving information about software and configuration updates and dynamically-delegated access control (including physical access control, for locking devices).

Metered access to resources and services: monitoring and payment for usage of utilities or services can be provided by IoT devices and associated smart contracts.

Digital rights and IP management: a blockchain can provide a trusted registry of media assets or other intellectual property, and can provide the ability to manage, delegate, or transfer access and rights information for those assets. Note that media are not necessarily stored on the blockchain itself. Instead, cryptographic hashes, meta-data and other identifiers stored on the blockchain might be integrated with bulk off-chain storage.

Data management: a blockchain can create a metadata layer for decentralised data sharing and analytics. Although large datasets themselves are unlikely to be stored there, a blockchain can help to discover and integrate those datasets and data analytics services. Access control mechanisms implemented on a blockchain may allow public data sources to be integrated more easily with private data sets and analysis services.

Attestation and proof of existence: a blockchain can be used to record evidence of the existence of data or documents, by creating a timestamped record of a cryptographic hash of the contents of those documents. This can be combined with records of the attestation or witnessing of corresponding physical documents by trusted third parties. However, it can be significantly harder to demonstrate the uniqueness or non-existence of such document records, unless there is a widely-accepted strict normal form for their contents.

Inter-divisional accounting: multi-national companies or large enterprises with separate divisional business units, often have jurisdictional or governance needs to control their own internal accounting, and yet also sharing accounting information with other divisions. A straightforward application of blockchain technologies on a shared private network can create a shared distributed ledger of inter-divisional accounts at the interfaces between divisions. Here the role of non-repudiation is for improved audit and change management of accounting information.

Corporate affairs (board and shareholder voting and registrations): the voting authorities of board members or shareholders in companies could be recorded and proxied on a blockchain. Smart contracts on blockchains could use that record to adjudicate votes conducted on the blockchain for specific motions. (As blockchain transactions are not necessarily hidden, cryptographic mechanisms may be required to prevent potentially undesirable strategic voting behaviours.)

2 SOFTWARE ARCHITECTURE AND DEPENDABLE SYSTEMS

This project uses the disciplines of software architecture and dependable systems as an investigative framework. Key aspects of these disciplines referred to in this report are summarised in this section.

2.1 Non-functional properties and requirements

When specifying a system, software engineers often distinguish functional requirements from non-functional requirements. For a computer system, simple functional requirements characterise the relationship between observable inputs and outputs. Non-functional requirements (NFRs) are needs expressed for non-functional properties (NFPs), which are also known as ‘qualities’, or ‘-ilities’. These include characteristics such as cost, security [1,3] (confidentiality, integrity [5], availability, privacy, non-repudiation), performance (latency, throughput), modifiability, and usability.

NFRs are expressed separately from functional requirements because they are often ‘cross-cutting concerns’ that span many system functions. For example, a requirement for the scalability of system performance might constrain the resources allowed to be used to respond in a timely way to a given level of concurrent demand, up to some limit on that demand. The demand in this requirement would typically be a mix of many different kinds of system functions in normal usage.

Different use cases carry different NFRs. For example, in safety-critical industries such as medical devices or aerospace systems, NFRs for safety are paramount. In enterprise software systems, regulatory requirements often constrain NFPs such as privacy and data integrity.

In regulated industries, legislation or regulation can provide constraints on minimum standards for critical NFPs within the industry. These constraints may be mandated to provide consumer protections, or to manage systemic risks or negative economic externalities within the industry.

NFPs are also important in understanding innovation. NFPs are quality or performance dimensions for technology, and technological progress pushes out the frontiers of performance on these various dimensions. Orders of magnitude improvements in performance on NFP dimensions open up possibilities for new markets and new business models using that technology innovation.


2.2 Software architecture – design and analysis

The software architecture [3] of a software-based system is the high-level structure of relationships between software elements (components and connectors) in the system. In the creation of a software architecture there are many possible options for these structures, and the choices between these options are important design decisions. A key idea in the discipline of software architecture is that these design decisions have a critical impact on a system's ability to meet NFRs. Given a design candidate for a software system, software engineers may use qualitative, analytical, or simulation-based tools to evaluate the design for its predicted ability to achieve a NFR.

To achieve a NFR, the right design decisions must be made, and each design decision will impact a number of NFRs, either positively or negatively. Often this will lead to conflicts between NFRs, so it is important to manage trade-offs between these when designing a system. An important part of software architecture as a practice is to document the design for a system, including the rationale for why specific design options were chosen.

2.3 Dependable software systems

A dependable software system [2] is one that must not fail because it is safety-critical, security-critical, or business-critical. There is increasing interest in using blockchains as part of dependable software systems, in domains such as health records, banking, voting, and personal identity. The *trustworthiness* of a system is orthogonal to its specific NFRs. In the field of dependable software systems, a *trusted system* is one a user has chosen to rely on for a purpose – if that system fails, the user will suffer some related harm or loss. A *trustworthy system* is one where we have justified assurance (evidence) that the system will not fail for that purpose. These failures may be of either functional or non-functional requirements for a system. In later sections we begin to explore how we may provide preliminary evidence to support assurances about designs of blockchain-based systems, using various kinds of design analyses. This is an open area of research, but is critical for facilitating adoption of blockchain technologies in highly regulated industries that need dependable blockchain-based systems.



There is increasing interest in using blockchains as part of dependable software systems, in domains such as health records, banking, voting, and personal identity.



PART II USE CASE STUDIES

Designs and design analyses for three use cases

This section describes the objectives, method and results of a study of design alternatives for illustrative three use cases.



3 STUDY OBJECTIVES AND APPROACH



3.1 High-level approach

Much of the innovation activity surrounding a new technology such as blockchain is an exploration of its suitability for various use cases. Details about the context and requirements for specific use cases are important in understanding the consequences of using this new technology and its ability to support new markets and business models. The overall objective of this study is to better understand the technical risks and opportunities of blockchain technologies, through an examination of high-level design alternatives for a variety of representative use cases.

The study was broadly structured in three phases:

1. Identify a range of use cases through desk research, workshops, and discussions with companies and government agencies. From a list of initial candidates, we selected three use cases which provide reasonable coverage of representative blockchain use cases.
2. Elaborate some high-level requirements for these use cases, and create or document candidate solution architectures for them, using blockchain and conventional technologies. These designs are then evaluated against the use case requirements.
3. Develop laboratory-based 'proof of concepts' for some design elements, for demonstration and laboratory-based testing.

After phase 1 was completed, phases 2 and 3 were conducted concurrently.

3.2 Criteria for selection of use cases

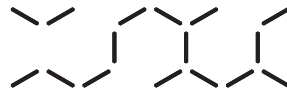
The criteria below used for the selection of use cases for the study were intended to support learning opportunities:

- Be potentially technically suitable for use of blockchain technology.
- Cover a range of non-functional properties.
- Cover prominent categories of use cases (e.g. variety of industry sector, blockchain function).
- Availability of informed stakeholders to inform use case definition.

The first criterion, of when blockchain technology is a good technical design choice for a particular use case, is not a settled question. Indeed, this project is intended to improve our understanding of the question. Ultimately it might only be answered through trial and error in the global innovation ecosystem. Nonetheless some drivers likely to be important are:

- Multi-party processes across organisations or industries, potentially where there are intermediaries acting within current systems.
- Need for greater transparency, provenance and visibility of transaction history by a wider range of stakeholders.
- Reducing cost and inconsistency from redundant data management across existing systems, providing better ways of sharing data between stakeholders.
- Need for improved and more flexible ways of supporting data integrity and access control.
- Business opportunities arising from inefficiencies, cost, or impediments to innovation within current systems.

4 DESIGN AND ANALYSIS OF BLOCKCHAIN-BASED SYSTEMS



From an initial list of candidate use cases, we selected three using the criteria listed previously. The selected use cases are:



1. SUPPLY CHAIN – in an agricultural supply chain, recording events from various participants in the neutral ground provided by a blockchain, in order to improve supply chain visibility and efficiency, and to track provenance.



2. REGISTRY – using blockchain to host government registries of open government data, for improved access to facilitate interoperation with registries of commercial data.



3. PAYMENTS – approaches for using blockchain to support improvements to efficiencies of international remittance payments.

These are discussed in this section.

The designs discussed for these use cases do not exhaust the possible solution space, and are not expected to be optimal. They are intended to be reasonable but simple high-level designs, revealing some aspects of the technical risks and opportunities for blockchain technologies.

4.1 Use case 1: Supply chain



In manufacturing, retail, and agricultural industries, supply chains are critical in the movement of goods and services across organisational boundaries. Supply chain contracts are complex, dynamic, multi-party arrangements, with regulatory and logistical constraints. They often cross jurisdictional boundaries. The information exchange in a supply chain is as important as the physical exchange of goods. For example, customs inspections would not start until both the physical goods and the information about those goods are present. Confidence in supply chain documentation can expedite customs and biosecurity processes, reduce risk and insurance costs, and be used as leverage in trade finance. Payments are made between parties at many points in the supply chain.

For food products, being able to tell where ingredients were grown, and how products were processed and distributed can be important in establishing confidence in food safety, creating and building high-quality brands, help in preventing fraud, and improving supply chain efficiency. These benefits may be felt by consumers globally and by producers domestically, although producers and logistics agents are likely to bear most of the total costs of deploying and operating the blockchain infrastructure.

Supply chains are a highly promising area for the application of blockchain technologies. The neutral ground provided by a blockchain is expected to help integrate the disparate participants in a supply chain, and the integrity and audit trail in a blockchain ledger is expected to improve transparency and confidence across the supply chain. Smart contracts and digital currencies on a blockchain can enact payments when linked to key supply chain events, as e.g. described in the sidebar on AgriDigital.

4.1.1 STAKEHOLDERS AND HIGH-LEVEL SUPPLY CHAIN VIEW

There are many stakeholders in an agricultural supply chain, ranging from producers, to transport providers, sorting/processing facilities, wholesalers, distributors, retailers, and consumers. In international supply chains there are also stakeholders related to customs and biosecurity regulation. A simplified configuration of stakeholders and functions is shown in Figure 1 for illustrative purposes. Note that, in this figure, we abstract from transport providers, which would be involved at numerous stages of the supply chain, as well as interactions with customs or biosecurity or international partners.

The information systems supporting supply chains reside at the individual supply chain participants, and are integrated to varying degrees, i.e., from no digital integration with machine-readable barcodes that can be understood by a number of participants to full system integration with digital message exchanges.

The identification of stakeholders participating on a blockchain is critical, and can be a difficult challenge for public blockchains. In this section we assume the operation of a private consortium blockchain, and the management of identity information through off-chain governance mechanisms for that consortium.

4.1.2 KEY NON-FUNCTIONAL REQUIREMENTS

It is unlikely that a single system will address the entirety of industry's supply chain. However, supply chain systems will normally have requirements on the following NFPs:

- **Interoperability:** A huge challenge in logistics is to coordinate information exchange across the many different kinds of goods, modes of transport, and information systems. Individual shipments can be aggregated into larger consignments, which means tracing information about the status of goods can require integration of different interlinked information sources.
- **Latency:** The exchange of physical goods must sometimes wait upon exchange of documentation associated with the delivery. This should not introduce significant additional delays.
- **Integrity:** Supply chain quality and provenance require that information about goods and supply chain events cannot be falsified or created without proper authority.
- **Confidentiality:** some information in supply chain documentation should be held commercial-in-confidence. Although the meta-data about any particular shipment is rarely highly confidential, information about aggregate trade flows can be commercially sensitive. Because of long supply chains and the use of subcontractors, parties' interests in information about supply chain events may extend beyond the parties directly involved in that event.

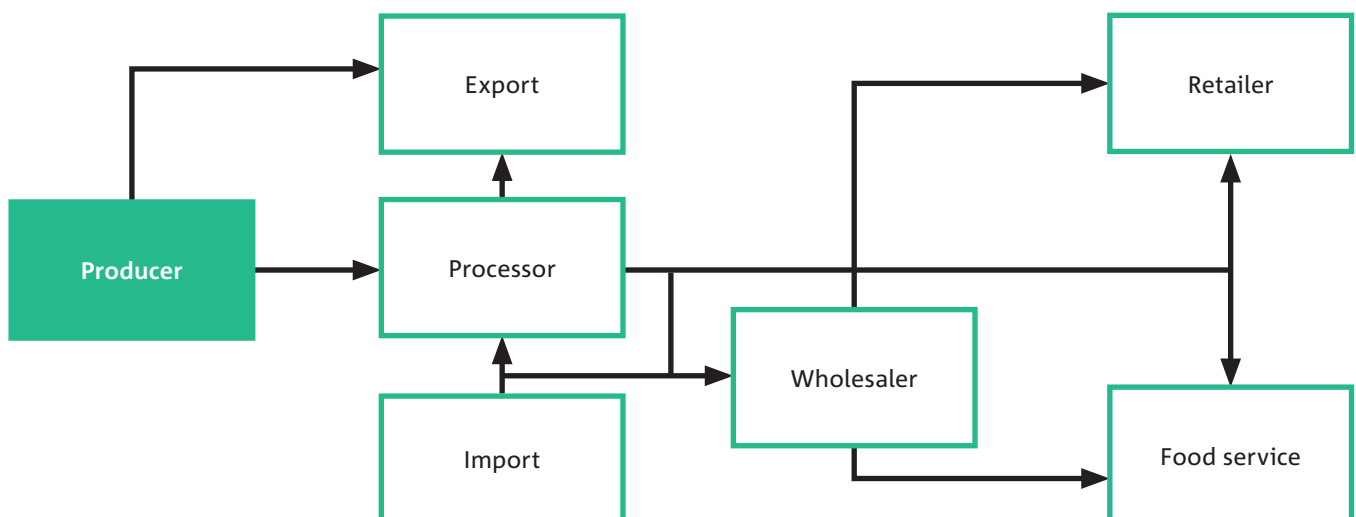
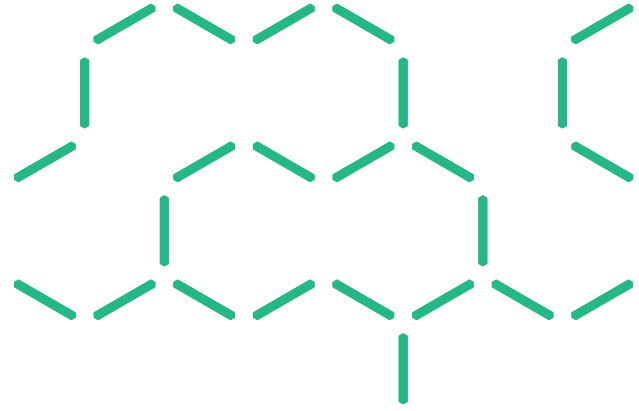


Figure 1 Stakeholders in a simple agricultural supply chain



- Scalability: There are many supply chain processes in progress at any time across a large number of different parties. Each process instance creates a large number of events, although not all events are relevant to all participants. A system must scale to handle the total throughput of transactions, with parties using resources in proportion to their level of involvement in the process.

4.1.3 DESIGN OPTIONS

We describe three design options for a system to record supply chain events: one reflecting existing piecemeal solutions, one using a consortium blockchain for event tracking, and one using a consortium blockchain smart contracts for executing supply chain processes.

Design 1: Conventional technology

Traditionally, supply chain information is recorded separately by each entity in the chain, and different players in a supply chain are privy only to the information involving themselves. As supply chain systems have become more digitised, and supply chain audits have become more commonplace, information sharing across different systems has become more common. The GS1 standards organisation has introduced the 'EPCIS' format for recording supply chain events. This allows different parties in a supply chain to record and exchange information in a standard format.

Figure 2 depicts a design for a supply chain system using EPCIS and other data with conventional technologies. All EPCIS data is sent to a central aggregation server for an agreed portion of the supply chain. A group of supply chain participants agree on a trusted party to operate and control access to the aggregation server. Note that this design would be an advance over many current supply chain systems, but has been implemented in few industrial settings, because of trust considerations and the competitive tensions between parties. The central server is trusted not to reveal commercial-in-confidence information to unauthorised parties. For availability and integrity, the centralised server creates a risk as a single point of failure, either from an operational failure or from the possibility that it might manipulate the data on the server.

Supply chain events are not the only data that needs to be exchanged. Other documents may include Bills of Lading, Booking Confirmations, Arrival Notices, Container releases, Terminal Load List, Delivery Orders, Tax Invoices, and so on. Agricultural products and foods may have different requirements than, say, consumer goods. Many industries have no single accepted standard, especially internationally. These other types of data are kept local to the systems of the different supply chain participants, and exchanged directly with point-to-point integration between parties. So a supply chain participant's system needs to be separately with each new participant.

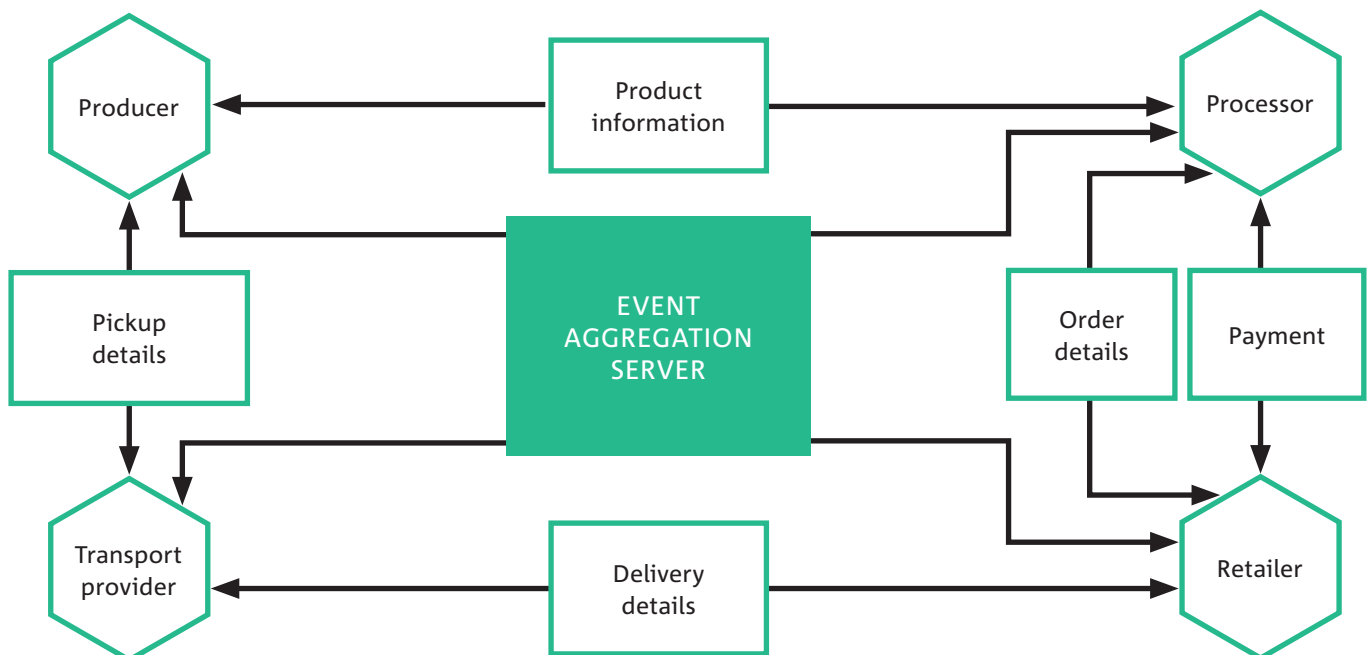


Figure 2 Model of supply chain using conventional event aggregation server and point-to-point integration

Design 2: Supply chain event tracking on a blockchain

We could consider an alternative design for a blockchain-based system that records and shares supply chain events. In contrast to the conventional design presented above, all events are recorded on a consortium blockchain for a specific industry sector. With this approach, we replace the central event aggregation server with a blockchain network for exchanging supply chain events. All non-event data is still exchanged in a point-to-point manner between participants. This may reduce some of the barriers to adoption noted above for the centralised system, but some competitive tensions remain, as discussed below.

The design is depicted in Figure 3. Note that there may be thousands of companies in each of these identified roles, and they may interact in complex chains of transport and supply relationships that are not shown in this diagram.

Although using a blockchain instead of a conventional central server, we can still utilise the GS1 EPCIS standard as a format for events. GS1 EPCIS events have 4 dimensions:

- *What* (an identifier of the relevant object)
- *Why* (business reason for this event, e.g., ‘received’, from a predefined vocabulary)
- *Where* (an identifier of the location), and
- *When* (a timestamp, including time zone).

For our blockchain-based solution, we complement that with a *Who* dimension, designating the originator of the event.

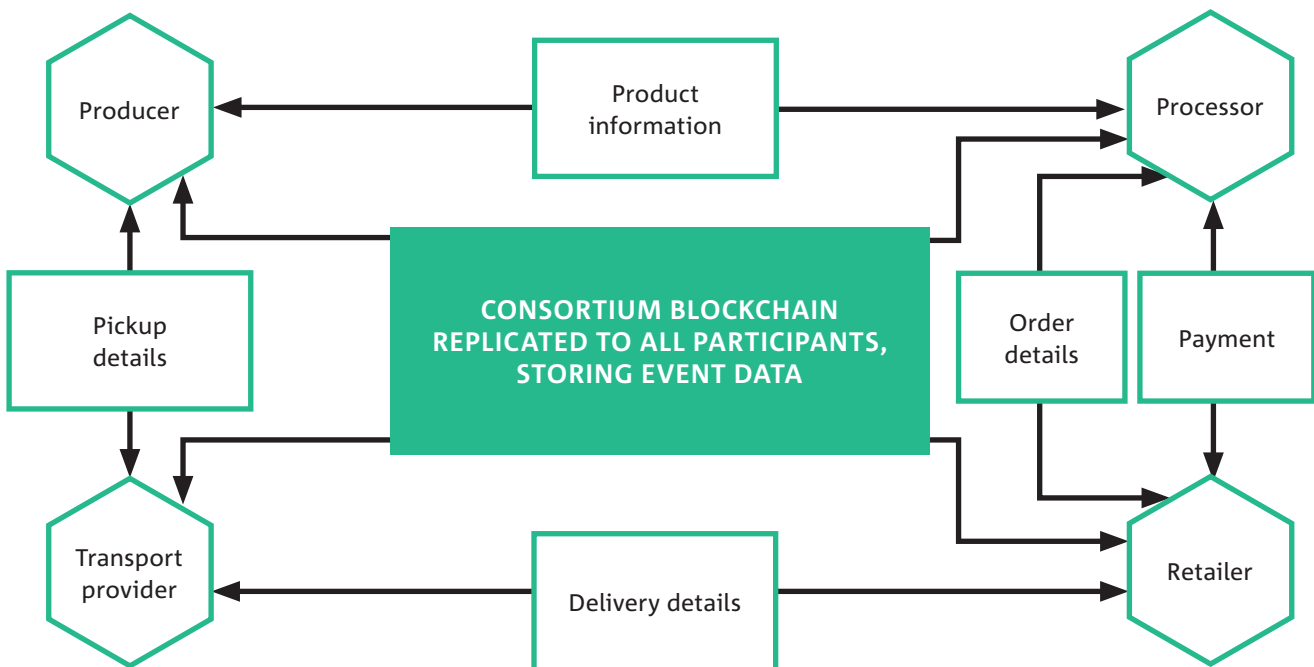
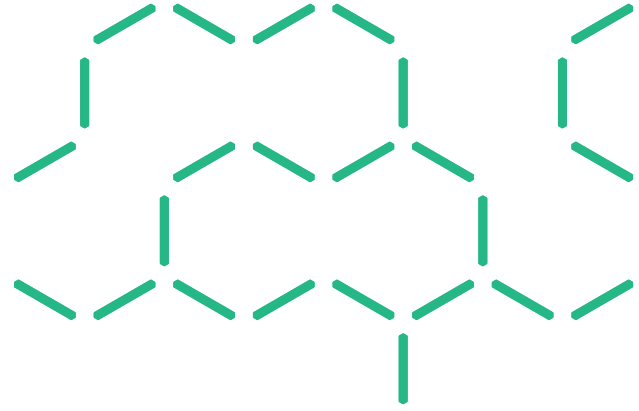


Figure 3 Model of supply chain using blockchain for event data and point-to-point integration



As the information on a consortium blockchain is readable by anyone permitted to join the blockchain network, sensitive information may be viewable by a participant's competitors. To limit this, we could encrypt all fields of GS1 events except for the *What*. Participants of a particular supply chain process would be required to share the cryptographic keys offline, so that only they can decrypt the information. By keeping the *What* data unencrypted, the participants can query the blockchain data using the object identifiers. Other data fields would not be readable except by those holding the keys. (For long-term storage of highly confidential data, some parties may not even consider encryption to be adequate protection, and may want to preserve physical isolation of stored data; in which case a globally-distributed ledger would not be viable at all.) Nonetheless even if encryption is considered to be adequate protection for the details of the data itself, participants on the consortium blockchain will still be able to see aggregate volumes about trade flows and may be able to perform data mining to re-identify competitors and customers from the blockchain history.

When the GS1 data is stored and committed on the blockchain, it is replicated to all participants of the consortium blockchain. The participants can query the blockchain like a conventional database, or they can actively analyse the new data as it arrives. Note that the visibility of data is limited by the above-described encryption mechanism.

The consortium blockchain here is envisaged as being for an industry vertical, in order to facilitate representation of information relevant to that industry. However, no industry vertical is really isolated from others – there are inherently links created when manufacturers combine goods from multiple industries. Here we do not discuss or investigate whether this would be best resolved by integrating multiple industrial blockchains, or by consolidating on one.

Design 3: Implementing supply chain processes on blockchain as smart contracts

Design 2 above uses blockchain in a flexible manner and records events with integrity guarantees – as such it enables provenance. In the third design, we follow another approach: supply chain process design, implementation, and enforcement on blockchain. That is, a group of participants that want to implement a supply chain process agree on a design for the collaborative process that regulates how interactions should take place.

Consider a simplified exemplary process: export of a single container of wine from a rural Australian producer. This starts when the producer initiates a shipment, and ends (for illustrative purposes) when the container is on a ship. Figure 4 shows the process model.

This kind of collaborative process can be implemented using smart contracts running on blockchain. Moreover, these smart contracts can be generated automatically from the process model [15]. In the resulting system, the supply chain participants interact with each other by sending messages through the blockchain. To facilitate interaction through blockchain, part of our approach is a so-called trigger component which acts as a bridge between the blockchain and the enterprise application worlds. The trigger can translate conventional service calls to blockchain transactions, and vice versa. Thus, implementation cost for this solution can be kept relatively low.

The smart contract can enforce the process as follows. First, it can reject messages if they arrive at the wrong point in the process. Second, messages are only accepted from the participant who is authorised to send them. For instance, customs clearing can only be granted by customs. Third, conditions can be specified on the process model level and executed in smart contract code directly, so that e.g. a particular process branch automatically gets activated when certain conditions are met or certain events are observed.

In terms of standardisation of messages, we can leverage the same standards as in the previous design. Similarly, encryption is handled as before, such that only minimal information is visible to all blockchain participants. A consortium blockchain will provide some additional access control protections.

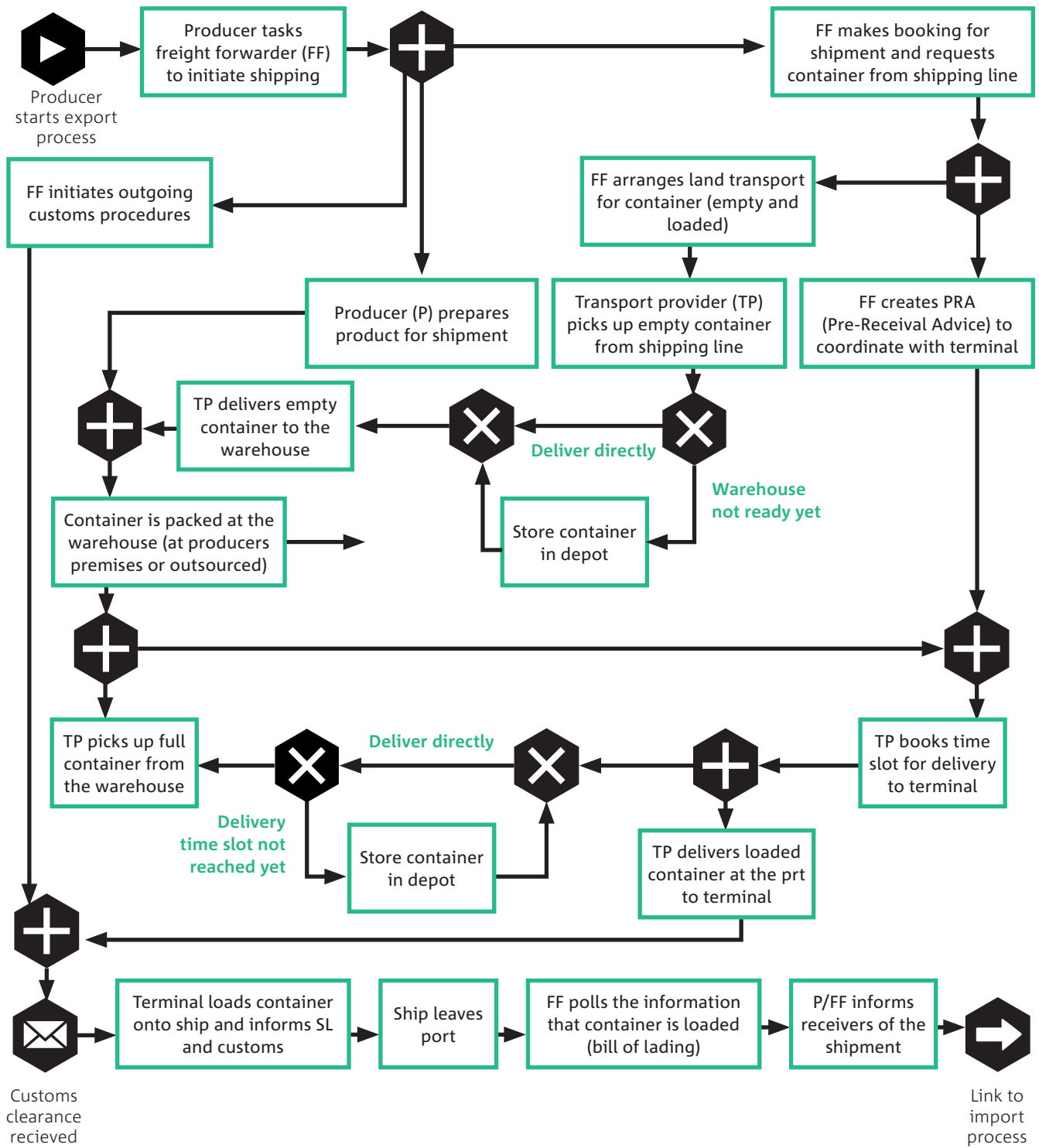
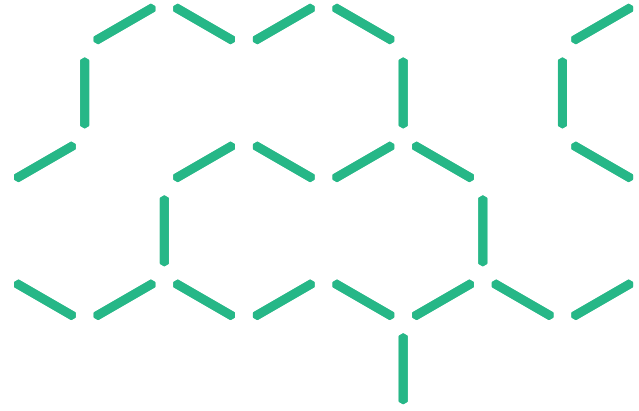


Figure 4 Process model of an agricultural export supply chain process. This process model can be used to generate smart contracts to implement coordination for this process on blockchain.



NON-FUNCTIONAL PROPERTIES

Scalability

In all three designs, each party has to deal with the scalability of their own enterprise applications, not discussed here. We consider scalability of the components shared by all parties. In design 1, this is the central aggregation server. If all participants publish all event data for item movements, this might become a bottleneck. There are two options to address that: filtering to only publish events that are relevant for other parties, or limiting the range of participants that can use a particular aggregation server. In the latter case, it may be subsequently possible to define a way to federate data access across multiple aggregation servers.

In designs 2 and 3, the component shared by all participants is the blockchain. Scalability of reading from the blockchain can be good, since each participant can hold their own full copy of the blockchain. For writing new transactions (and for design 3, smart contract method calls), scalability is currently limited on public blockchains. We propose using a consortium blockchain, where transaction volumes can be controlled, and where other technical options for block formation and consensus are available to improve performance. As with design 1, only relevant events should be stored on-chain. In design 3, communication is also limited to the messages exchanged as part of the collaborative process execution.

In laboratory experiments using design 3, we observed that we can scale to around 1,000 parallel process instances, using commodity blockchain software without further optimisation⁴. Globally, real-world supply chains would have orders of magnitude more than 1000 process instances actively executing at any time. However, consortia operating private blockchains could control the demand on any deployed blockchain instance. Furthermore, we expect that the throughput scalability can be increased by multiples with careful design and performance tuning. Other types of blockchains, not using Nakamoto consensus, have been specifically designed for private or consortium blockchains, and are expected to offer significant performance gains.

Interoperability

Designs 1 and 2 use the GS1 EPCIS standard for events, but require point-to-point integration between any two participants for the other documents. Extending the supply chain to a new participant requires integration of that participant's system with all participants that need to exchange documents directly with the new participant.

The collaborative process in design 3 requires the same amount of integration initially: the data formats used during its execution need to be agreed upfront. However, any new participant needs to integrate their systems with the given process, and thus the integration burden for the remaining participants is reduced. This methodology may as such also increase the uptake of standards for supply chain documents, since there is a central medium, the collaborative process, which makes the adoption of standards particularly beneficial.

Latency

Supply chains typically involve the physical movements of goods, so many latency requirements on information transfer are usually on the order of minutes to hours. Neither of the designs should suffer from latency exceeding these timeframes. However, at points of handover, there may be low latency requirement for confirmation of receipt of goods. Blockchain commit times are likely to be too long for this, but it may be possible to instead provide cryptographically-signed receipts off-chain, with the delivery agent able to lodge those to the blockchain at a later time.

⁴ To analyse the performance of this design, we set up an experimental environment where we translated the process model from Figure 4 to a smart contract and executed it on a private Ethereum blockchain. Specifically, we executed 4,000 instances of the process with a total of 80,000 events. The limiting factor in our setup is the complexity permitted *per block*, which comprises (i) size of the data attached to transactions and (ii) computation required for smart contract calls. We used the default configuration and initial limit per block from the standard Ethereum client *geth* (<https://github.com/ethereum/go-ethereum/wiki/geth>), where a new block is created approx. every 13.3 seconds. Each of these blocks can theoretically be filled to the limit, although there are some influences in practice which slightly reduce this capacity. In summary, it took 435 blocks to execute all 4,000 process instances; in a standard Ethereum blockchain that equates to approx. 1h 36min. After a ramp-up phase in the beginning and until the cool-down phase at the end, typically between 900 and 1,000 process instances were active in parallel. The ramp-up and cool-down phases observed are artefacts of our experimental setup.

Integrity

Design 1 relies on a trusted party to operate the aggregation server, and is subject to the possibility of manipulation with a low chance of detection.

Integrity is a strong inherent feature of blockchains: information captured as part of committed transactions would be exceedingly hard to change. This is the strong suit of designs 2 and 3. It may be desirable to store large blocks of data off-chain. In this case it is standard practice to store a cryptographic hash of this data on-chain. As such, it becomes trivial to detect alterations or corruption of the off-chain data.


Confidentiality

Confidentiality requirements for supply chain data are not the same across industries or participants. This affects all three designs: for a specific supply chain and a specific set of participants, the confidentiality requirements need to be formulated and analysed, and potentially the design needs to be adapted accordingly. The main trade-off is between the benefits of sharing data within the group of collaborators – visibility and cross-party optimisations are impossible without that – and retaining confidentiality towards competitors where needed. Supply chain information can be commercial-in-confidence. This may include the identities of participants, trade volume, prices, and delivery times.

It is possible to restrict access to the aggregation server in design 1 and the consortium blockchains in designs 2 and 3. Nonetheless, multiple competing participants might gain access to the same system, perhaps by playing multiple roles in the market. Even a private blockchain does not protect commercial-in-confidence information. Unless the supply chain is entirely vertically-integrated within one organisation, competitors will be sharing access to information on the blockchain. The only way to prevent that is by setting up a separate aggregation server or blockchain for each group of parties. That is, switching transport providers would require setting up a separate system, which would not only be tedious and resource-intensive, but would also severely hamper the analysis of supply chain data across specific instances.

Data stored on a blockchain is readable to all participants of that blockchain. Confidential data can be encrypted, and keys can be exchanged between supply chain participants so that only the ‘right’ group of participants can decrypt that data. However, this requires off-chain key exchanges and diligent handling of keys. Furthermore, encrypted data can itself not be processed by the blockchain or its smart contracts. Thus, transfer of assets that are managed by the blockchain cannot be encrypted; and encrypted data cannot be transformed or actioned by smart contracts.

Another confidentiality concern is the amount of interactions between parties. It is possible to create new addresses for each process instance, but the flow of assets may still be used to infer relationships between addresses. Reidentification attacks may still be possible, and aggregate trade volumes might be inferred. Dummy transactions may be used to attempt to hide this. Such protection mechanisms can help, but erode the benefit of using a blockchain at all. These trade-offs require careful consideration.



The rate at which blocks can be created is limited, often by using a proof of work mechanism, whereby a processing node can only add a new block by demonstrating that a difficult task has been completed.

SIDEBAR: AGRIDIGITAL'S BLOCKCHAIN TRIAL IN A GRAIN SUPPLY CHAIN

AgriDigital, a Sydney-based startup company, is developing a product to support food supply chains. As part of the first phase of their product development, they focus on the early stages of grain supply chains, i.e., when growers transport grain to an initial buyer of the grain, and the ownership of the grain is transferred in return for payment. Quoting from AgriDigital's press release: "AgriDigital has successfully executed the world's first-ever live settlement of a physical commodity on a blockchain between a grower and a buyer."

Trial setup. During the harvest in late 2016, AgriDigital ran a trial where a blockchain-based system was deployed in parallel to a traditional backend. The blockchain operated in shadow mode as follows. Both backends, i.e., the Blockchain-based and the traditional backend, received the same requests from AgriDigital's frontend system, and the traditional backend served the responses that were used productively. In parallel, the blockchain-based backend computed a response, which allowed AgriDigital to cross-check its outputs. So the blockchain-based solution was tested live, but without relying on it in production.

Procedure. The trial concerns the part of the supply chain starting with the grain being loaded onto a truck. When the truck arrives at the buyer's site, it passes a first weighbridge and a sampling station. The information from the weighbridge is the gross weight, i.e. the weight of the grain as well as the truck and trailers. The sampling station picks a sample of grain, which is processed in an adjacent lab to assess the quality of the grain. The quality of the grain determines the price per ton; together with the gross weight, an upper bound of the price can be calculated. The data (gross weight, quality, price) is sent to the AgriDigital frontend, which creates a blockchain transaction containing this information. This transaction is supplied with the respective amount of digital currency to cover the price. For the purposes of this trial, AgriDigital minted the AgriCoin, where 1 AgriCoin corresponds to 1 AUD. The transaction invokes a function of a grain supply chain smart contract, which in turn confirms the price calculation, verifies a sufficient amount of AgriCoin has been supplied, and stores the values into its local data storage. The AgriCoin supplied is kept in the smart contract's own account, acting as escrow.

Then the truck physically unloads the grain into the buyer's silo. Subsequently, upon leaving the buyer's site, the truck passes a second weighbridge. Here the weight of the empty truck, the *tare weight*, is measured. The second weighbridge forwards that information to the AgriDigital frontend, which in turn creates another blockchain transaction with that data. Invoked by the second transaction, the smart contract calculates the *net weight*, i.e., gross weight minus tare weight. The price for the grain is then recalculated as net weight times price per ton for the grain's quality, and a title for the grain with its net weight and quality is created. The final price is transferred to the grower, and the grain ownership title is transferred to the buyer. Additionally, the GRDC levy is paid, and any royalties for the grain, e.g., to respective plant breeders' rights-holders, are deducted. Therefore, the ownership of AgriCoins and grain titles is updated by the time the truck leaves the buyer's site. For settlement in traditional systems, AgriDigital's system generates bank messages for payment and a receipt for the grain title.

The main goal of the trial was to show that the truck's appearance on the weighbridges triggered all system interactions, which was achieved. Steps that are yet to be automated are: (i) establishing that the weighbridges fulfil the conditions (having been inspected by authorities within the past 12 months and not recalibrated), and (ii) automated generation of the quality assessment message, which is currently entered manually by a technician in the sampling station's lab.

Technical implementation. AgriDigital set up a private network of the Ethereum blockchain with three nodes, simulating the situation where AgriDigital, the buyer, and a third party like the regulator each operate a full blockchain node. The private blockchain has been configured to mine approximately one block per second, where each block may or may not contain transactions. The electronic weighbridges automatically created messages with their measurements, which they sent to the AgriDigital frontend. As mentioned above, the data entry from the sampling station was done manually into the AgriDigital frontend. Ethereum is currently limited in its handling of decimal values, and thus some rounding error occurred as expected.

The information presented in this sidebar was collected through an interview with AgriDigital at their premises, and based on their press release on the topic. AgriDigital gave feedback on a draft version. Data61 representatives did not review the actual systems or data involved.

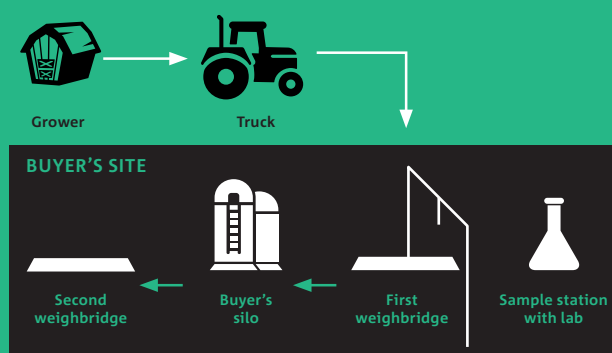


Figure 5 Physical setup of AgriDigital's trial

4.2 Use case 2: Registry



Registries are authoritative collections of information, often managed by government agencies. A registry holds information about a class of entities. Examples of such entities include individuals, businesses, species and organisations. In Australia, familiar registries include the immunisation registry, the business name registry, and land title registries. There are also well-known international registries such as the Domain Name Service (DNS). Some government registries are described as ‘public’, and can be queried by individuals. However, query access to these registries may be limited to prevent attempts at re-publishing or mining the data. Unfettered data mining could threaten commercial or personal privacy, and is often restricted using regulatory policies, and technically with query rate limiters and user access controls.

Some government registries contain periodically published, open data. In Australia, these are published through data.gov.au. In this case study, we specifically consider the use of blockchains for managing an open data registry of data sets, data sources, and data analytics services. So, we do not consider confidentiality or privacy issues for this use case. Blockchains provide transparency about their entire transaction history to all processing nodes. In a public blockchain, this means that the information is openly published. It is possible to run a private blockchain hidden behind a web service or other interfaces. This could limit access to the registry in a way that satisfies an appropriate access policy. However, many of the benefits of using a blockchain would be foregone in such an architecture. Private blockchains may provide a way to integrate registries across multiple government agencies, but this is not explored further below.

Although here we discuss open government data, we note that there are also non-government open data sets of national importance. This can include scientific data from universities, and data from non-profit institutions (including industry associations and consumer organisations). These data sets are not included in sites such as data.gov.au, but a blockchain of open data could provide neutral ground to federate references to all of these data sets. Also, instead of storing the open data directly in the blockchain, only metadata is stored. This provides a federated index to the data which are kept in the source repositories independently-managed by their governing bodies.

4.2.1 STAKEHOLDERS

For open data, the major stakeholders are data providers, data consumers, and the data registry. Data providers may include government agencies, research institutes, universities, and companies. Data providers record metadata about their datasets on the data registry, and make their data available on their websites. Data consumers query to discover datasets in the data registry based on the metadata. They can then download the datasets from the data providers for analysis.

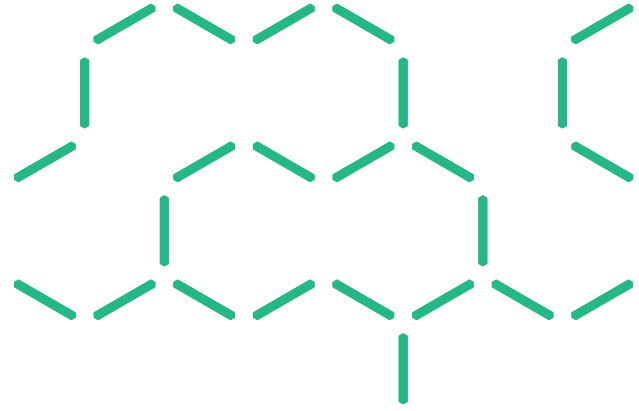
4.2.2 KEY NON-FUNCTIONAL REQUIREMENTS

Some of the key Non-Functional Requirements (NFRs) for open data registries include:

- Integrity: each data provider should only be able to create and change registry entries for their own data sets.
- Availability: there should be high likelihood of being able to access the registry when desired, for both data providers and data consumers. This particularly applies to national public registries, which form the basis for many other services that utilise the data from the registries.
- Read latency: data consumers may need to repeatedly query the registry while browsing and searching for relevant data sets. This may be done programmatically from a graphical user interface and so should have low latency.
- Interoperability: A registry may reference other registries to reduce duplication and errors.
- Ease of integrating new data providers: to grow the network effects of the registry as a data portal, it is important to have low barriers (time, cost, and administrative burden) to add new data providers to the registry.

4.2.3 DESIGN OPTIONS

We provide three illustrative design options for such a registry. These are: conventional technologies operated by a single agency, a shared private blockchain operated by data providers, and a public blockchain.



Design 1: Conventional technology

Data portals such as data.gov.au implement a dataset registry using conventional technologies such as CKAN⁵. The CKAN software is run and managed by a single government agency. Data consumers interact with the registry to discover datasets, but retrieve datasets directly from data providers. The data providers may perform some permission management for data access independently. An illustrative high-level design is shown in Figure 6.

In the ecosystem of CKAN, the datasets in different CKAN repositories refer to each other through importing metadata from the referred repository to the primary repository and transferring it to the format used by the primary repository with possible customer-defined fields.

Design 2: Data registry on consortium blockchain across data providers

One design alternative using blockchain is to replace the backend of a conventional registry implementation with a consortium blockchain across data providers. Not all data is stored on the blockchain. For example, the registry may maintain a separate database for administrative purposes for permission management. As above, the data providers perform some permission management for data access independently. Instead of integrating with the registry's web service as in the conventional approach, data providers must instead integrate with the shared blockchain. Data consumers access the registry through an open data portal, which is hosted by a government agency. An illustrative high level design is shown in Figure 7.

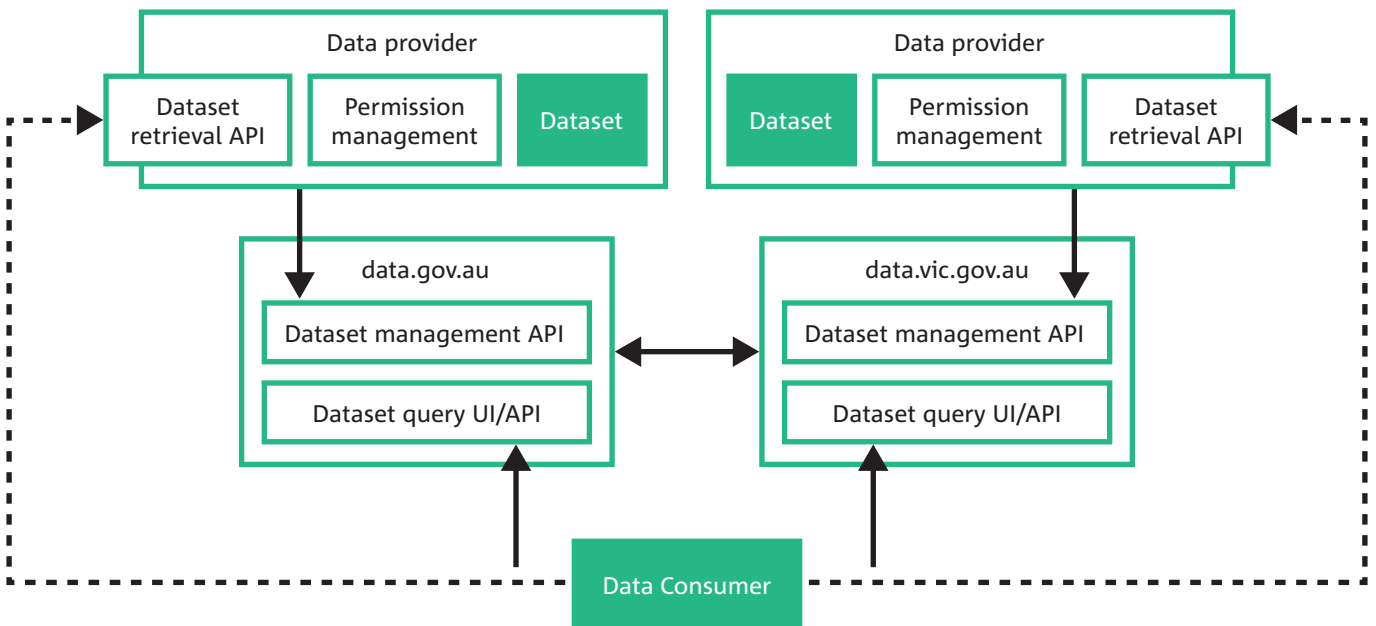


Figure 6 Design for a registry using conventional technologies, operated by a single agency

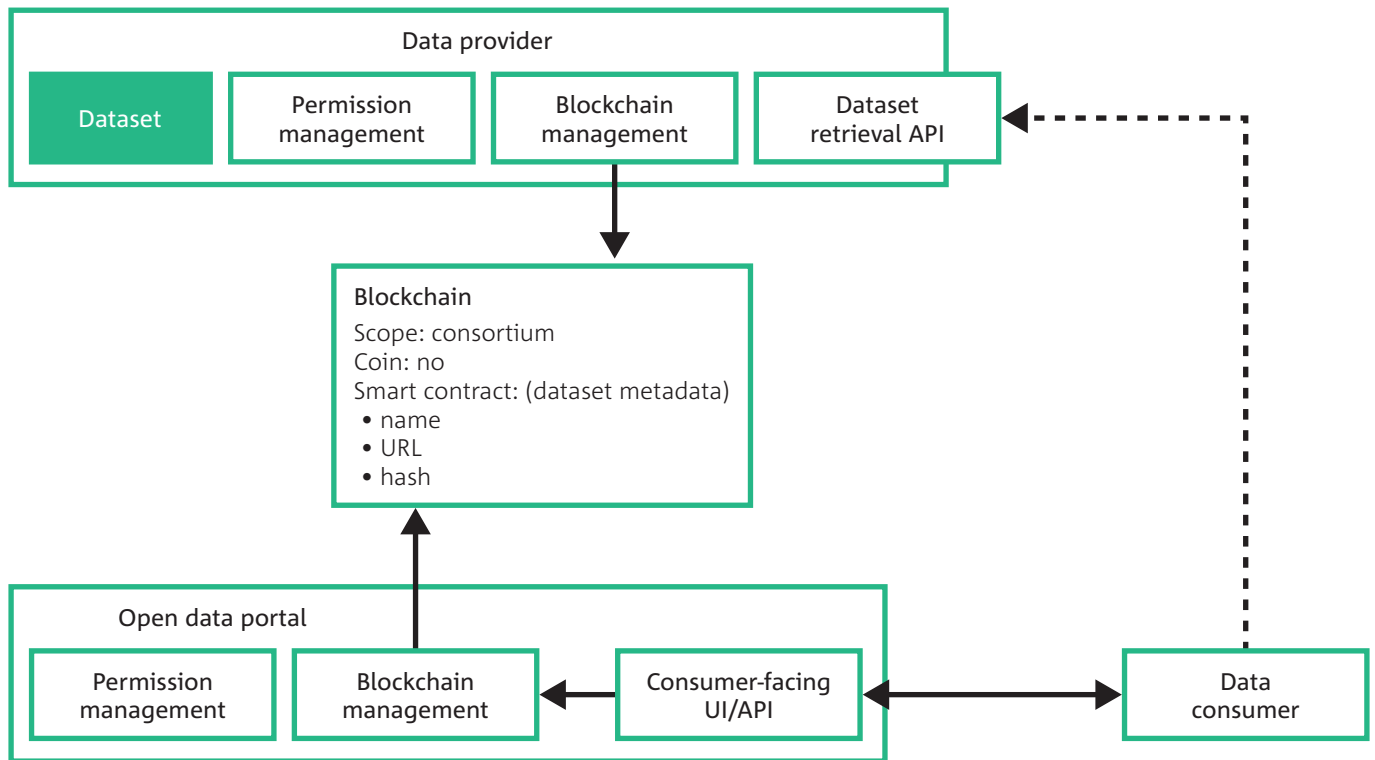


Figure 7 Design for a registry using a private blockchain

Design 3: Data registry on public blockchain

Finally, we consider a design which replaces the registry with a public blockchain. In this design there is no agency operating the registry. Instead the data providers independently record metadata on the public blockchain and perform their own permissions management and access control for their data sets independently. Note that there may still be an agency leading governance for the registry. In this design, data consumers are required to interact directly with the blockchain, rather than with a consumer-facing user interface or API. However, those consumer interfaces may be provided by a variety of commercial or personal systems, depending on the data provider's preferences. An illustrative high level design is shown in Figure 8.

4.2.4 NON-FUNCTIONAL PROPERTIES

Integrity

Design 1 relies on the registrar to create registry entries on behalf of data providers. New registry entries are validated solely by the registrar. In designs 2 and 3, registry entries can only be created by the data provider, using their private key, which must be kept secret for this purpose. All transactions are validated by all processing nodes in the blockchain network. In design 2, data consumers only access the registry via an interface which could modify information reported to consumers. In contrast, in design 3, data consumers hold a local copy of the blockchain, through which they access the registry, which removes the interface from design 2 as a possible point of manipulation.

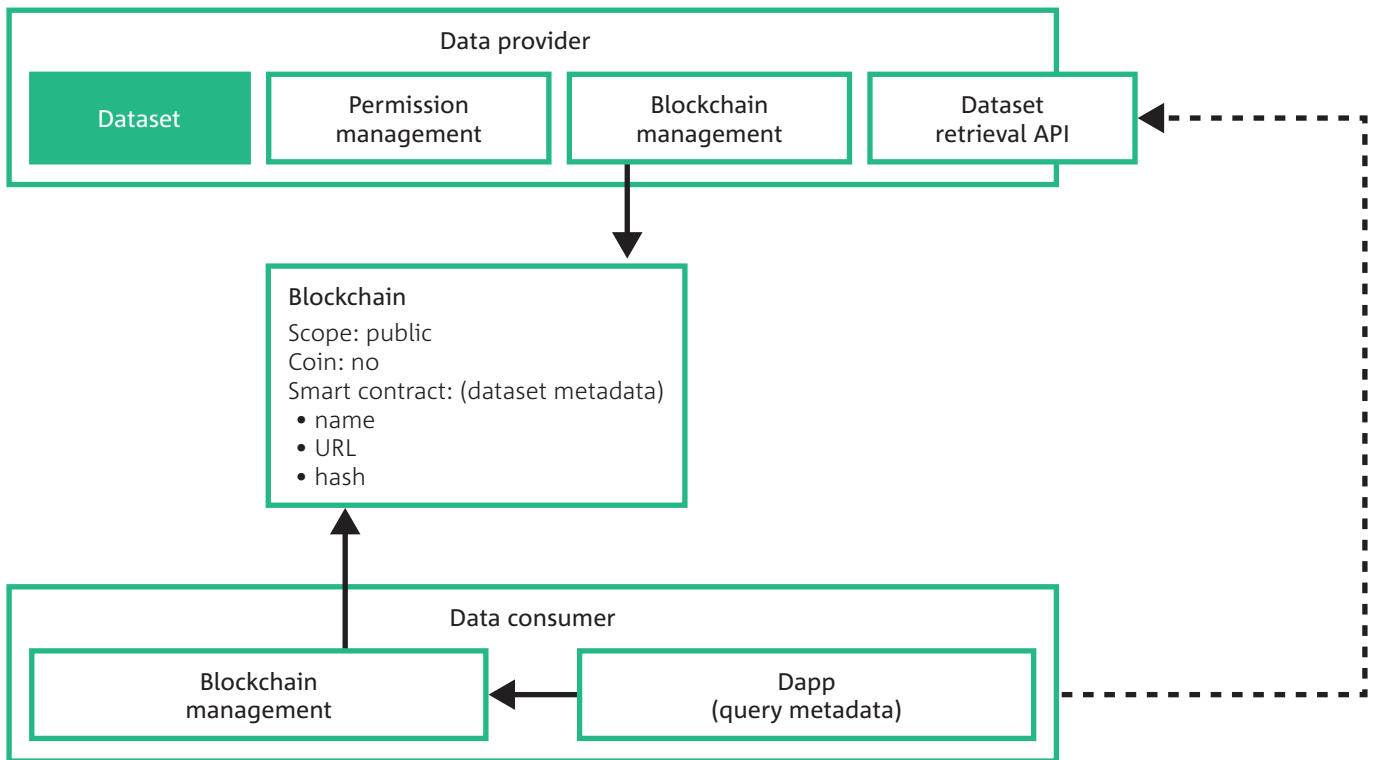
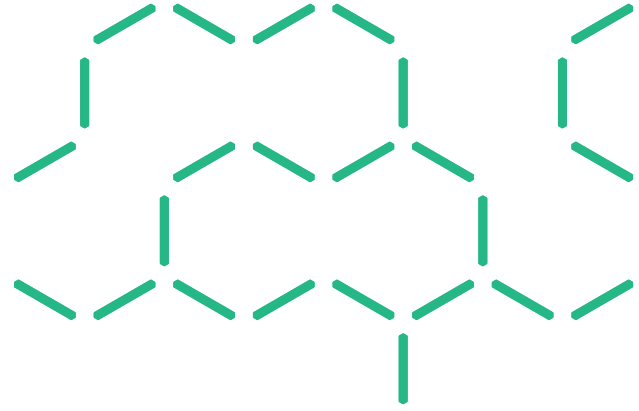


Figure 8 Design for a registry using a public blockchain

Availability

In design 1, the data registry system is a single point of failure for registry availability for all stakeholders. In design 2, the open data portal is a single point of failure for data consumers. Such a single point of failure could be mitigated with an IT architecture using redundant servers and network infrastructure. The use of a blockchain allows increased data redundancy which can improve read availability for data consumers. In this use case, write latency is not critical, and makes it easier to achieve higher service availability for writing registry entries.

Interoperability

In design 1, the datasets in different CKAN repositories refer to each other through importing the metadata from the referred repository to the primary repository and transferring it to the format used by the primary repository with possible customer-defined fields. Designs 2 and 3 use a blockchain as shared infrastructure, which means different registries can more easily interact with each other.

Read latency

Reading in design 1 and 2 is performed through a remote API over the internet. Compared with design 3, this is slower: in design 3 a blockchain local node is collocated with the consumer's query interface, and reading is done locally.

Ease of adding providers

In designs 1 and 2, new data providers are added through account creation and network configuration for the registry back-end services. In design 3, new providers can join by independently creating a new public/private key pair. Authentication of their public key could be certified by the registrar on the blockchain, or separately off-chain. Data providers in designs 2 and 3 must integrate with the blockchain, and should ideally run a blockchain node.

Cost of using public blockchain

To investigate the cost of using a public blockchain for this use case, we built a CKAN-inspired registry on a laboratory deployment of the Ethereum blockchain. The registry was populated with data taken from data.gov.au⁶. The example registry has three entities: organisation, package and resource. Each entity has 6 attributes that are stored on the registry. Architectural decisions can affect the cost of deploying and executing the registry. For example we could use a 'single' registry that holds all records as values in the data store as a singleton smart contract, or we could use a 'distributed' registry which manages each record as a separate smart contract. For a 'distributed' registry, a main registry smart contract creates entry contracts and stores pointers to them. The 'single' option is suitable for simple registries, while the 'distributed' option is suitable for registries with complex operations, such as finer-grained permission management at the level of individual records. Table 1 gives statistics about the different options, both for creating a registry on the blockchain and the cost of adding a record to the registry.

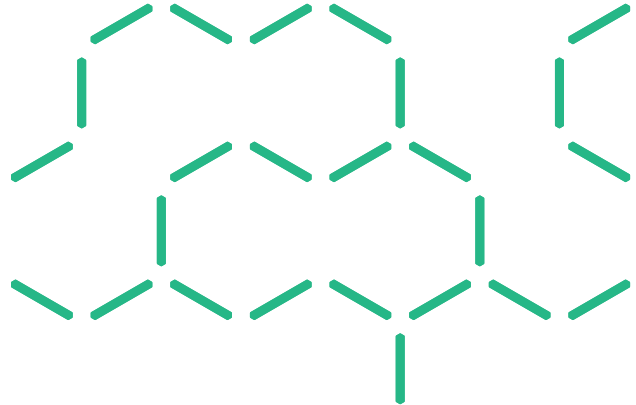
The cost of creating a registry contract is comprised of fixed costs and variable costs. Fixed costs include the base amount for the transaction itself and the cost for allocating an address on the blockchain. Variable costs are affected by the architectural design of the registry contract, for example, the cost of data payload. Similarly, the cost of adding records to a registry is also comprised by a fixed cost for the transaction itself, and some variable costs including for the data payload and cost to execute the functions defined in the registry contract. Compared with conventional databases, using public blockchain costs more to add records. However, the data becomes globally replicated and the blockchain ecosystem will retain this data indefinitely as long as the blockchain exists, at no additional cost.

Table 1 Gas cost and dollar cost of registry functions ⁷

	NUM.	REGISTRY DEPLOYMENT				RECORD CREATION (AVERAGE)			
		GAS COST		USD COST ⁷		GAS COST		USD COST	
		SINGLE	DISTRIBUTED	SINGLE	DISTRIBUTED	SINGLE	DISTRIBUTED	SINGLE	DISTRIBUTED
Organisation	533	1836926	2542604	US\$0.9	US\$1.3	183266	931179	US\$0.1	US\$0.5
Package	33810	1836926	2542540	US\$0.9	US\$1.3	340022	1090174	US\$0.2	US\$0.5
Resource	64147	1777127	2548455	US\$0.9	US\$1.3	302041	1065760	US\$0.2	US\$0.5

⁶ Scraped at: 2017-03-07T15:59:32+1000

⁷ https://poloniex.com/exchange#usdt_eth



4.3 Use case 3: Payments



Many workers in Australia regularly send money back to their families overseas. Remittances are low-value payments individually. Nonetheless, they constitute up to about 10% of GDP in some developing countries (27% in Tonga and 20% in Samoa). [17, 18] Thus, high remittance costs have important implications on socio-economic development of these countries. However, remittance costs in Pacific Island countries are among the highest in the world. For example, it costs \$33.20 to send \$200 from Australia to Vanuatu, and \$28.60 to Samoa. [17,18]

There can be many parties involved in the chain of transactions made for these payments and there is sometimes little transparency on the total cost of exchange rates and fees. Remittance payments can also be complicated and made more expensive by the difficulties of satisfying AML/CTF (Anti-Money Laundering/ Counter-Terrorism Financing) regulation, especially where the receiving party may not have a bank account. These transactions can have high latency, with transaction times ranging from less than 1 hour to 5 days.

4.3.1 CONTEXT AND STAKEHOLDERS

In this use case, stakeholders include remitters, beneficiaries and different types of financial institutions, including banks and Money Transfer Operators (MTOs). We consider the stakeholders and functions depicted in Figure 9.

In a preliminary phase the local financial institutes each independently perform Know Your Customer (KYC) checks about the identities of remitters and beneficiaries. When a remittance begins, the remitter pays for a remittance to a financial institute from the remitting territory, which transfers the money across the border. Another financial institution from the beneficiary territory receives the money and exchanges the money to local currency, which it disburses to the beneficiary. Prior to the completion of the exchange, and depending on the amount of money transferred, Anti-Money Laundering (AML) and Counter-Terrorism Financing (CTF) checks that are required by regulators in either territory (and in any financial institution in intermediate territories) will be performed, based partly on the previously-established identity of the remitter and beneficiary, and perhaps including assessment of the purpose of the transfer.

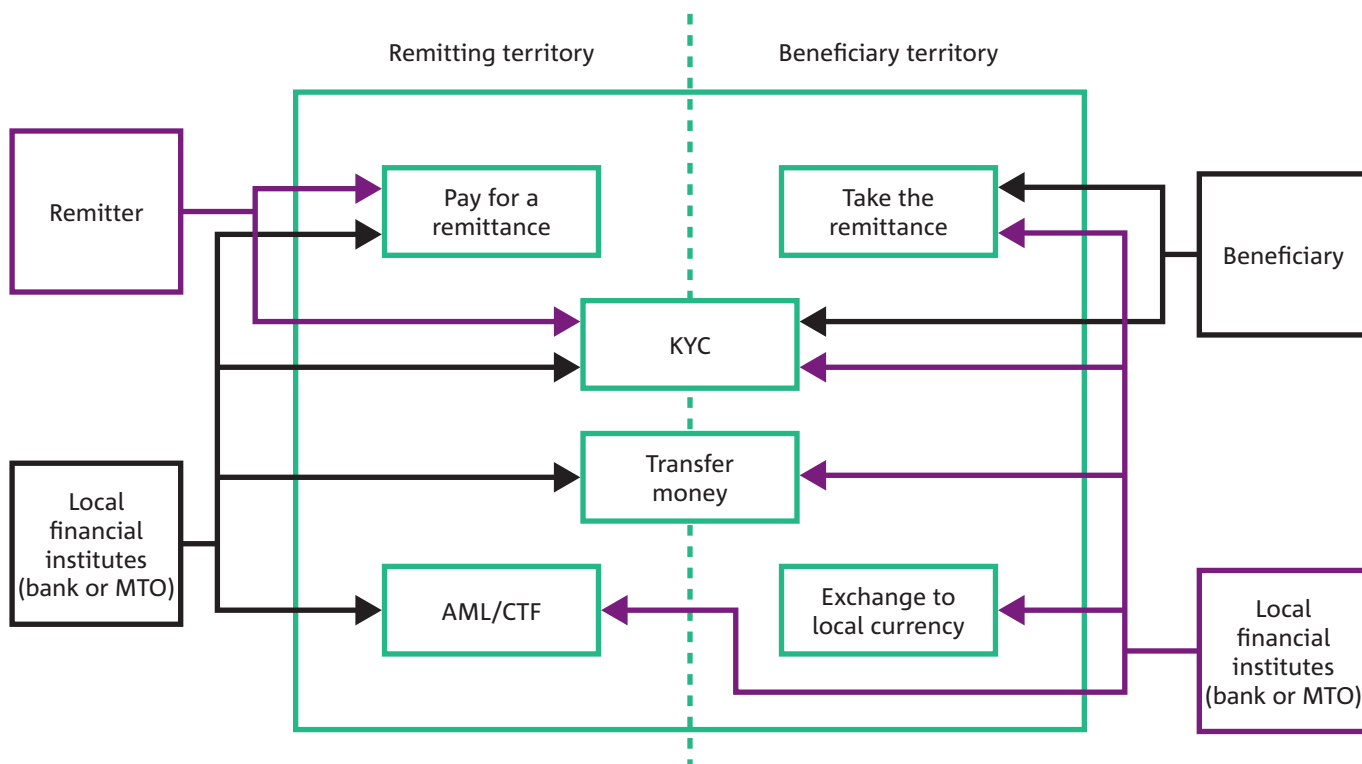


Figure 9 Stakeholders and functions for remittance payments

4.3.2 KEY NON-FUNCTIONAL REQUIREMENTS

Remittance has many Non-Functional Requirements (NFRs) [7], but here we focus on a few key requirements. Although integrity is one of the most critical NFRs for remittance, we do not focus on it because it is less contrastive between the design options.

- Transaction latency: completing a remittance payment should ideally be instantaneous, or at least take place comfortably within the context of human interaction with a physical kiosk or web form.
- Cost: the total cost of remittance should be a low percentage of the transaction value.
- Cost transparency: the total cost including fees and exchange rate should be transparent to participants.
- Controlled confidentiality: for regulatory compliance, all required AML/CTF checks must be performed, but appropriate levels of commercial confidentiality must also be maintained. Foreign correspondent accounts have been used to launder money and to potentially finance terrorism.
- Barriers to entry: increased competition can drive lower costs and greater service innovation, but this requires low barriers to entry (cost, time, and regulatory burden) for new remittance service providers.

4.3.3 DESIGN OPTIONS

We provide three illustrative design options for remittances, one using conventional remittance technologies (via a bank or a money transfer operator), and two using blockchain (payment through blockchain, and sharing identity information through blockchain). Note that these are simplified designs for illustrative purposes. There are many blockchain-based systems currently in development within the financial services industry to facilitate international payments. A variety of system architectures have been proposed for development, usually combining blockchains with conventional technologies. We are not able to explore all possible designs here, but note that emerging technical solutions for transactions using zero-knowledge proofs may significantly change the design landscape with regards to confidentiality and transparency of transactions.

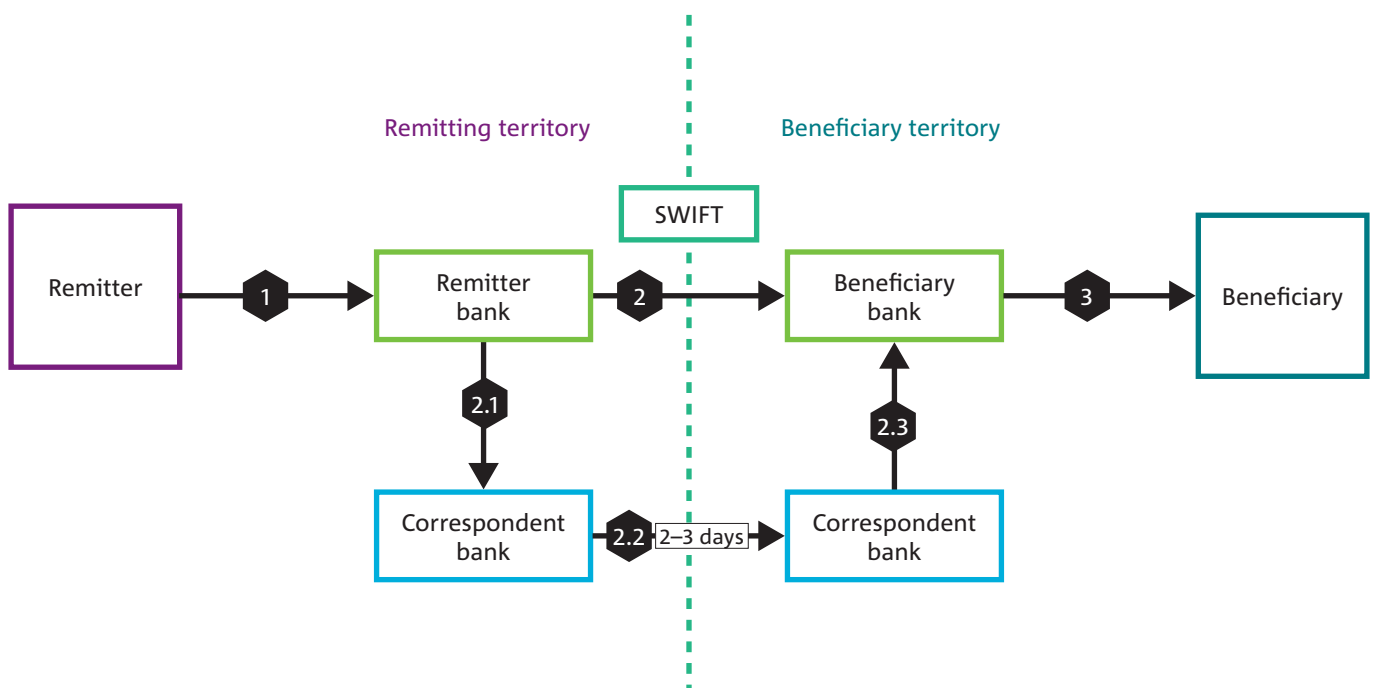
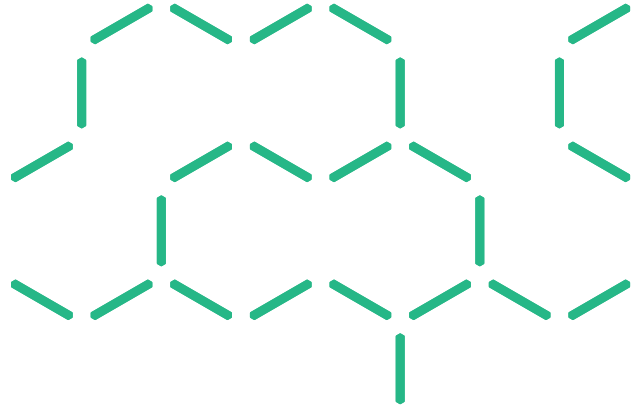


Figure 10 Remittance through bank



Design 1: Current state with remittance through banks or MTOs

The process depicted in Figure 10 starts when the remitter deposits money into their bank. The remitter's bank then initiates a SWIFT wire transfer to send the money across to the beneficiary bank, possibly through several intermediary correspondent banks. It can take 2-3 days for the money to be sent. The receiving bank then informs the beneficiary's bank that the money in the foreign currency has arrived, it then gives the beneficiary's bank the local currency equivalent. Finally, the beneficiary's bank disburses the local currency to the beneficiary.

Another widely-used way to do remittances is through a Money Transfer Operator (MTOs), as depicted in Figure 11. In this case, a Remitter uses either cash or other payment instruments to pay the MTO. Once a group of payments is received, the remitting MTO pools all money into a single transaction. The MTO also prepares a file with instructions on breaking down the remittance to individual orders, and sends the file to the beneficiary MTO. Then, the money is transferred by the MTO to its foreign bank as a normal international transfer as shown above. The bank charges the MTO once for all the remittances. When the beneficiary MTO receives the money, it distributes it according to the instructions received earlier.

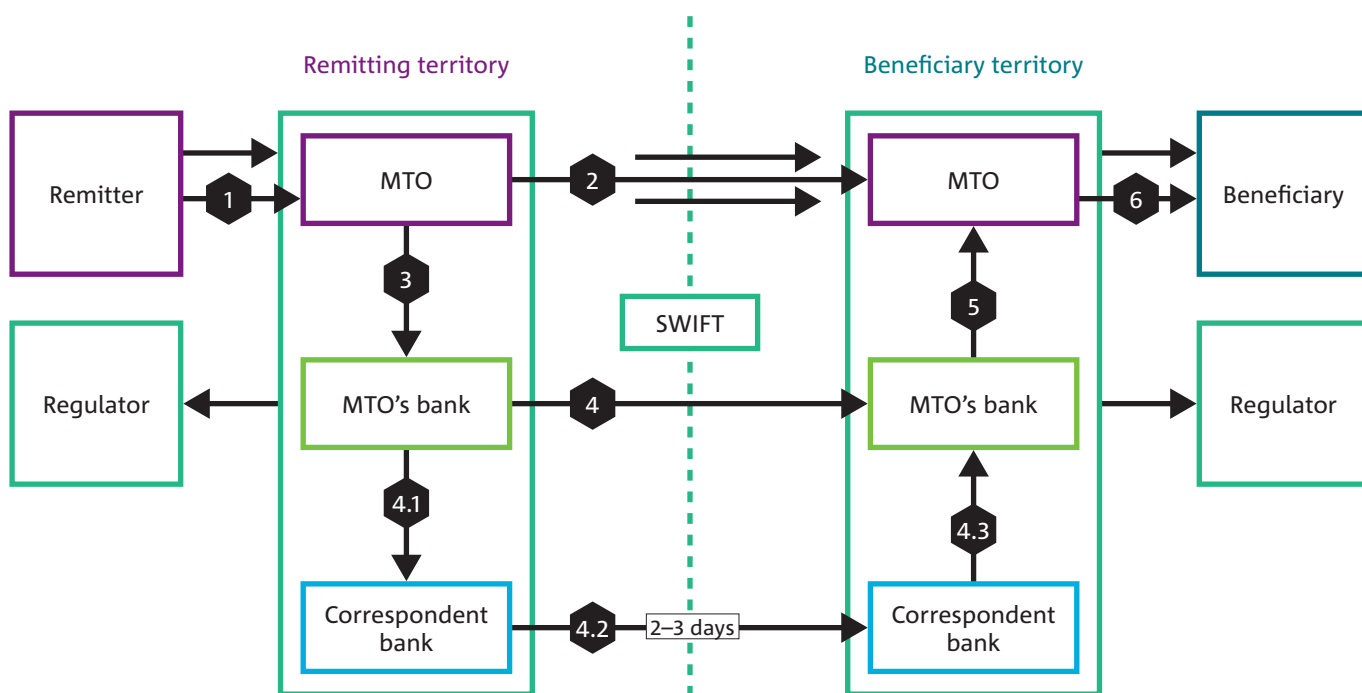


Figure 11 Remittance through MTO

Design 2: Payment through Blockchain

Banks, financial institutes, and MTOs could join a private blockchain to enable real-time settlement, as depicted in Figure 12. Apart from speeding up money transfers, blockchain could also help banks to operate continuously, 24 hours a day. The on-chain portion of the design can include SWIFT instructions or other payment instructions and the payment status. The native currency of the blockchain might be used as an intermediary currency to facilitate foreign exchange. Identity and risk information, fees, and foreign exchange rates are exchanged through conventional means, off-chain.

When Bitcoin is used this is sometimes called ‘rebitance’. Some companies use Bitcoin directly as an intermediary currency to do foreign exchange. The underlying Bitcoin layer is invisible to end users. In this case, every remittance has a corresponding transaction recorded on the Bitcoin blockchain. Other companies maintain a separate blockchain to facilitate settlement among branches, and anchor their blockchain with the Bitcoin blockchain as a way to leverage Bitcoin’s immutable, independently auditable ledger. Although Bitcoin addresses are pseudonymous, and are not always tied to known identities, individual addresses can be externally linked to KYC’d identities, and this is typically done by exchanges (and for remittance companies) within source and destination countries.

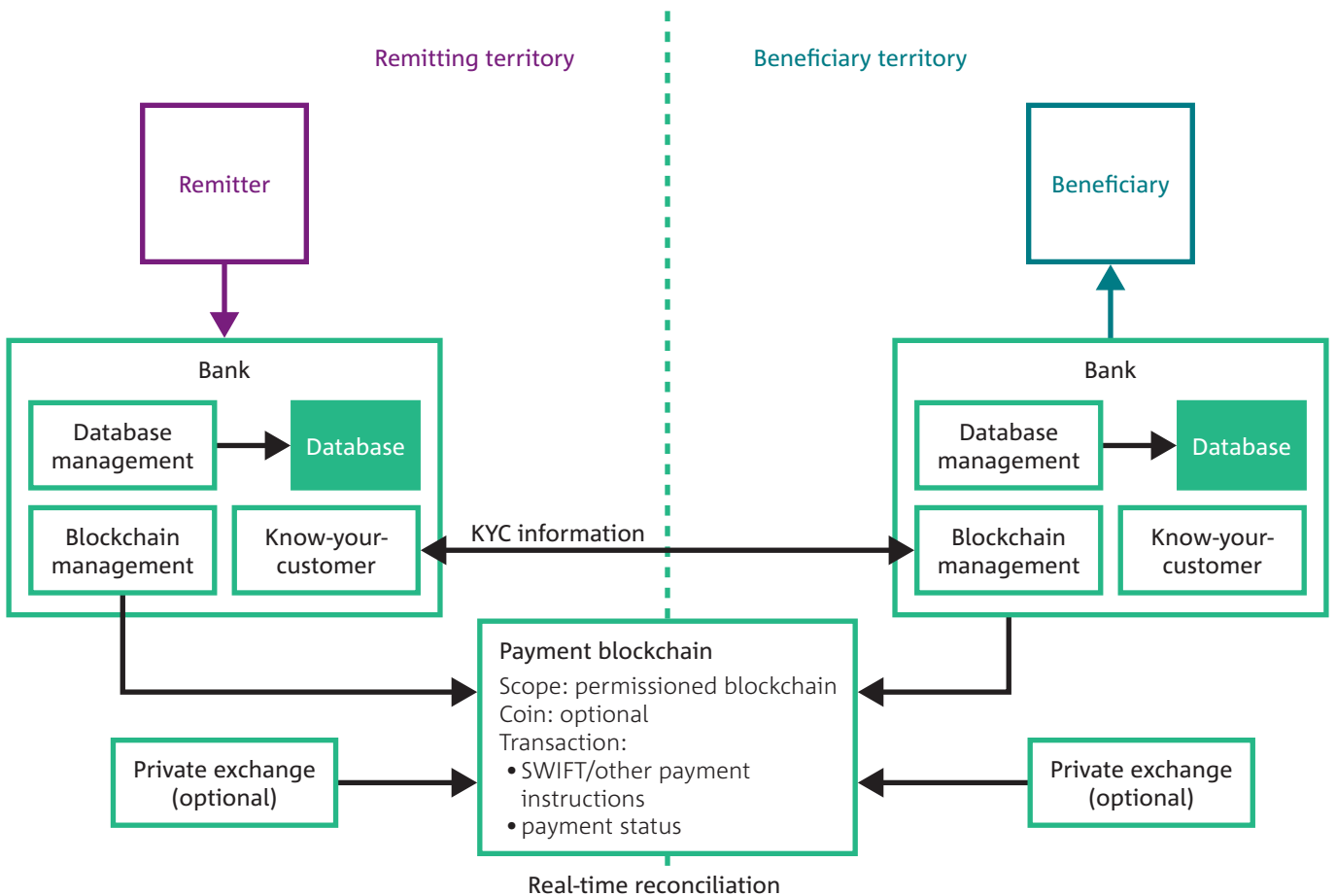
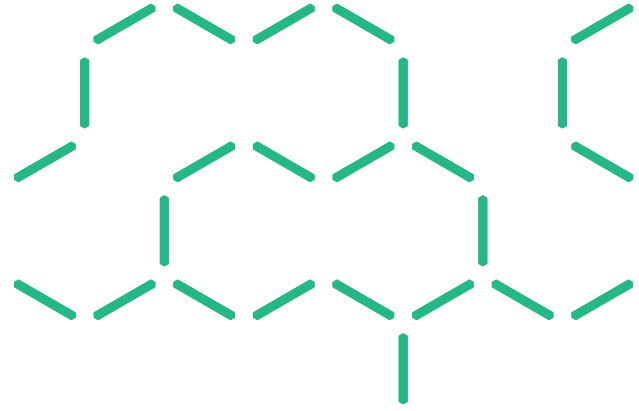


Figure 12 Payment through blockchain



Design 3: KYC through blockchain

KYC processes when on-boarding customers can sometimes take 30 to 50 days to complete due to the requirements financial institutions must meet. Current KYC mechanisms can entail substantial duplication of effort across banks and other financial institutions [8].

Blockchain can potentially help banks fulfil basic KYC requirements for new customer on-boarding while providing increased transparency, security and cost-efficiencies. An illustrative model is depicted in Figure 13. A blockchain can provide a single place to manage identity globally, which in turn can simplify and streamline the on-boarding KYC processes from the bank's perspective and enable more efficient AML/CTF and sanction screening at transaction level. The on-chain portion of the design could include encrypted personal details and supporting documents, and the status of the payments associated with the person. Alternatively, the data on the blockchain could be merely a reference point

with a cryptographic hash and a digital signature that gives users access to the corresponding customer's information, which in turn is stored in a separate repository outside the blockchain, to ensure a secure and confidential way of conducting and storing a customer's KYC information.

The information on a KYC blockchain could be managed by banks. Once a bank has KYC'd a new customer, the bank could then enter the relevant summaries, references, or hashes of KYC documents into the blockchain. This could then be used by other banks and financial institutes without the need for the customer to repeat the KYC process. Alternatively, the information on a KYC blockchain could be also jointly managed by clients. In such case, a KYC service on blockchain would allow participants to create and manage their own identities and relevant documentation. They could potentially grant permission to other participants to access their identity for KYC purposes. For additional validation, recognised notaries could provide attestation about the identity or documents.

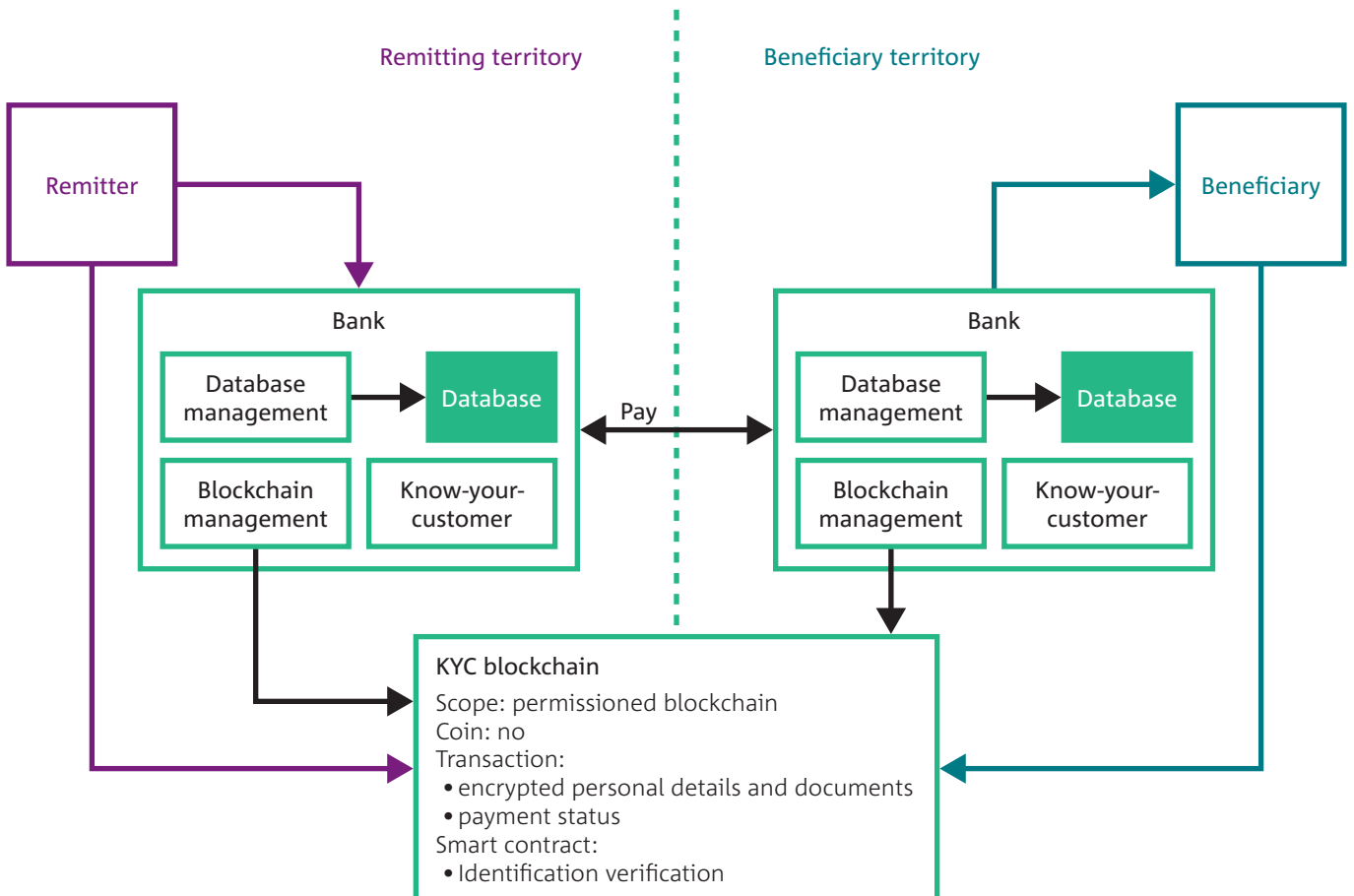


Figure 13 KYC through blockchain

4.3.4 NON-FUNCTIONAL PROPERTIES

Transaction latency

Design 1 can be time consuming depending on the number of correspondent banks involved. End-of-day batch processing can introduce delays of up to 24 hours, which can be exacerbated by time zone differences. Design 2 uses blockchain to enable real-time processing with latencies that vary from seconds to hours, depending on the blockchain. For example, on Bitcoin, the latency is typically one hour if 6-confirmation is assumed. At least 10 minutes is required for 1-confirmation.

Cost

In both design 1 and 2, remitting banks charge transaction fees and liquidity providers charge through the spread on foreign exchange rates. There are also correspondent bank fees in design 1. Design 3 potentially reduces the KYC costs significantly compared to the current KYC process.

Transparency

In design 1, each bank in the payment chain is aware of its own actions, but some KYC information is transmitted through the chain of correspondent banks. How FX spread is calculated and what will be charged in fees is not always predictable. Blockchain used in design 2 provides a common shared view of the payment status that enable real-time fraud analysis and prevention. On Bitcoin, regulators and others can access historical data in the blockchain, but would need additional information to know how to interpret the pseudonymous addresses and the identities of senders and recipients. Design 3 uses blockchain to share KYC information, which can potentially be linked to a real-time view of the payment.

Controlled confidentiality

In design 1, KYC regulatory compliance requires costly technology capabilities and complex business processes. There is substantial duplicated effort between banks and financial institutions. Design 2 replaces intermediary banks with a blockchain to provide a shared record of payments and KYC checks, and thus may simplify regulatory compliance along the payment chain. Some automated and real-time compliance checks may be available on-chain using smart contracts, depending on the blockchains used. Design 3 also enables automated and real-time compliance checking using shared KYC information on blockchain.

Barriers to entry

Design 1 requires participants to have banking or financial services licenses. Design 1 further requires business relationships with correspondent banks. Design 2 requires new technology development and integrations, but some existing transactions standards can be reused. Interaction between separate proprietary blockchains would require inter-ledger protocols. Bitcoin provides lower barriers to entry for new participants, but regulatory or banking constraints for digital currency exchanges apply to end-points within countries.



PART III DISCUSSION

Framing findings and recommendations from the study

The following sections discuss the risks and opportunities of blockchain technologies, and issues related to design, assurance, and regulation of blockchain technologies. We identify some of the limitations in our study. Key findings and recommendations are highlighted.



5 RISKS AND OPPORTUNITIES FOR BLOCKCHAIN-BASED SYSTEMS



Here we address some common ‘myths’ about blockchain, discuss some of the opportunities to work within the limitations of blockchain technology, and identify some distinctive opportunities provided by blockchain technology.

5.1 Blockchain myths

Table 2 identifies some common myths about blockchains, and elaborates on a more realistic description of the situation for each of these issues.

5.2 Working within blockchain’s limitations

Blockchain technology does have limitations, but these limitations are irrelevant to some use cases.

For example, privacy and confidentiality are hard to establish on a public blockchain, because any member of the public can obtain a full copy of the whole transaction history and use it without restriction. Even if parties try to use pseudonyms, the contents of a transaction are publicly visible, and reuse or connection of addresses through transfer of digital currency can provide opportunities for linkage attacks to re-identify participants. Nonetheless, this limitation does not matter for all use cases.

For example, public blockchains may be suitable as a register for some kinds of public advertising or fully-public (‘open’) government registries, even in highly regulated industries. Consider that banks advertise on television, but television is not required to be a trustworthy mechanism for highly-regulated banking transaction systems. Integrity may still be required. But rather than privacy or confidentiality, publicity is instead important. Public blockchains can provide integrity and publicity. Other similar examples might include secure software package management, and IoT device configuration updates. The open data registry discussed in our study is another example.



FINDING: PUBLIC BLOCKCHAINS MAY BE APPROPRIATE FOR SOME PURPOSES

Even in highly-regulated industries, public blockchain systems may be appropriate for use for some limited purposes, such as public announcements, product catalogues, software update checking, or fully-public government registries.

Another limitation of blockchains is that they are not suitable for storing Big Data, i.e., large volumes of data or high velocity data. This is an inherent limitation of blockchains, because of the massive redundancy from the large number of processing nodes holding a full copy of the distributed ledger. Nonetheless, this limitation does not matter for some use cases. As discussed previously, instead of storing data on a blockchain, often only a hash or other meta-data is stored on blockchain for large data. This can support integrity checking and enhanced information for data sets accessed through other off-chain communications channels. Again, the federated open data registry discussed in our study is an example.

Table 2 Blockchain myths

MYTH	REALITY
Solves every problem!	A blockchain is a kind of database and computational platform, with advantages and disadvantages compared to conventional technologies. Sometimes a blockchain may be an appropriate choice in the design of a software system, but for many purposes, conventional technologies will be more appropriate. In particular, if a system is used only within a single organisation or organisational unit, it is almost never advisable to build it on blockchain technology.
Trustless	Using a blockchain does not remove trust, because users are still exposed to risk in their use of blockchain technology. In a blockchain, what is trusted (i.e., relied upon) is the blockchain software, the incentive or contractual mechanisms driving the behaviour of processing nodes that operate the blockchain system, and the trusted third parties that act as ‘oracles’ which record information about the external world on the blockchain. Although a blockchain does not remove trust, it can remove the need to trust a single specific third party to maintain a ledger, and so is sometimes called a ‘distributed trust’ mechanism. In a blockchain-based system, the trust boundaries are wider. For example, if users access a blockchain through an intermediary, such as a digital currency exchange, they trust that intermediary: if the intermediary’s system fails, their users may lose control of assets on the blockchain.
Secure	‘Security’ is a broad class of NFPs. Classically, the three major security properties are Confidentiality (only authorised reading), Integrity (only authorised valid writing), and Availability. [1] (Other properties include Privacy, Anonymity, and Non-Repudiability.) Different use cases may need more or less of these various security properties. Because a blockchain replicates the full contents of its distributed ledger to all processing nodes, specific techniques such as encryption, or holding data off-chain, must be used with blockchains to achieve Confidentiality.
Smart contracts are legal contracts	This is not a settled question, but there are reasons to think that smart contracts may not be regarded at law as legal contracts. A smart contract is perhaps best thought of as either the source code text of a program or as a distributed physical machine executing a digital representation of that program. In either case, a smart contract is not an agreement, per se. Nonetheless, a smart contract may be some kind of evidence for an agreement, or may be a means for the execution of provisions of a contract.
Immutable	The linking of blocks in a chain of cryptographic hashes does support a kind of immutability for historical transactions. In traditional database systems, the ACID properties (Atomicity, Consistency, Isolation, Durability) are critical. However, for blockchains that use Nakamoto consensus (longest chain wins), the classic Durability property does not hold because a transaction initially thought by a participant to be committed (i.e. on the longest chain) may later turn out to have been on a shorter chain, and so no longer be committed. Such blockchains only offer a long-run probabilistic durability property, and so are not immutable in a simple way. [12] However, a) switching to a longer chain is evident to participants, and b) when a transaction has been committed to a blockchain for a sufficiently long time, it will in practice be immutable. Blockchains that use other consensus mechanisms (such as Practical Byzantine Fault Tolerance) can offer stronger, more conventional immutability properties.
Need to waste electricity	The public Bitcoin and Ethereum blockchains use a consensus mechanism called ‘Proof of Work’ which requires all mining nodes to compete to solve a difficult cryptographic puzzle. This guarantees that miners have invested resources in the blockchain (and so aligns their incentives with that of the blockchain community), and is also a form of random leader election, to allow a single node to resolve non-determinism in the formation of the next block. However, the world-wide pool of computers performing this cryptographic puzzle creates significant electricity usage, most of which is ‘wasted’ by not leading directly to a successful puzzle solution. This is a known and current limitation, but alternative consensus mechanisms are being developed for public blockchains, such as ‘Proof of Stake’, which do not use a computationally expensive puzzle, and will be markedly more energy-efficient. Private blockchains also often use alternative consensus mechanisms which do not rely on Proof of Work. Nonetheless, the massive redundancy in the large number of processing nodes in a blockchain system will always mean that more electricity is used than in a centralised non-replicated database. This is an inevitable trade-off for the distributed trust and increased availability offered by a blockchain.
Are inherently unscalable	Blockchain systems such as Bitcoin and Ethereum cannot currently match the maximum throughput of conventional transaction processing systems such as the Visa payments network. This is a known and current limitation, but is being addressed by the development of new mechanisms such as sharding, state channels, and reduced inter-block time. The extent to which techniques such as sharding can increase scalability depends on how effectively inter-dependencies between transactions can be dynamically identified and managed, but this is not yet well understood. While blockchains are currently not highly scalable, this is not necessarily an inherent limitation, and may be overcome in the medium-term future.
If beneficial, will be adopted	It is often assumed that if blockchain technology has significant benefits, then it will inevitably be adopted. However, there are many challenges to the adoption of blockchain. First, the many risks and limitations of blockchain must be weighed against their possible benefits. Second, the path to adoption of a technology is not always clear, especially where many of the benefits are significant only with large-scale adoption because of network effects, and where it is not clear whether the parties who benefit also bear the costs of deployment and operation. Third, the potential disruption and disintermediation enabled by blockchain may be a threat to powerful incumbent organisations who may act to limit the acceptance of blockchain technologies.

5.3 Distinctive opportunities

We cannot provide a full picture of all of the distinctive opportunities for blockchain technologies, because this is inevitably unpredictable for innovation with any new technology. The question is still being explored globally by governments, enterprises, and the startup ecosystem.

However, some initial lessons are apparent. The key advance from blockchain technology is distributed trust – removing the need to rely on a specific single trusted third party (or small number of specific trusted third parties) to facilitate transactions. This provides a distinctive opportunity when either a specific trusted third party is unknown or not sufficiently trustworthy, or when they are extracting fees that are high in relation to the transaction value.

Although not explored in detail as an illustrative example in our study, the combination of distributed integrity, digital currency, and smart contracts in a blockchain may enable new kinds of ‘programmable money’. Potentially, parties could attach policies on how parcels of money are spent or transferred, which would be self-enforced on blockchain as smart contracts. It would be configurable where the policy constraint expires for that parcel of money; a policy may expire on payment to a third-party, or else be carried through the payment ecosystem. For government expenditure, it might be possible to programmatically control policies on the spending of parliamentary entitlements, grants, or social service payments; or more generally to implement forms of dynamic fiscal policy.



FINDING: BLOCKCHAINS AND SMART CONTRACTS MAY MAKE IT POSSIBLE TO CREATE ‘PROGRAMMABLE MONEY’

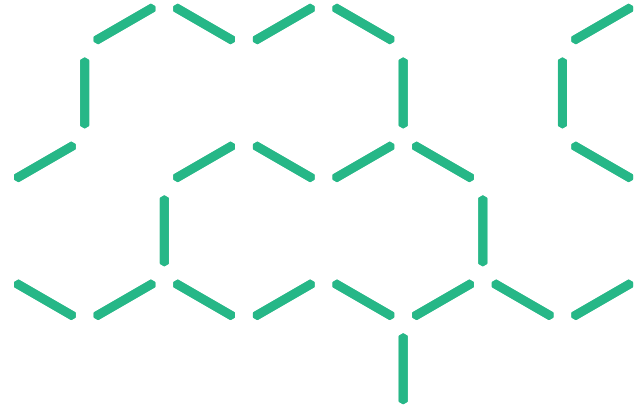
Blockchains and smart contracts may make possible a new form of ‘programmable money’, where policies can be attached to how parcels of currency are spent.

For the remittance use case discussed in our study, remittance solutions demonstrate that blockchain could reduce the direct cost of financial transfers through an intermediary digital currency. However, KYC regulatory challenges must also be met, and this does not yet have a widely-accepted solution. Identity is critical here, but identity information does not necessarily need to be stored on-chain – off-chain protocols might be used instead. Whatever the solution, it needs to not just achieve the technical challenges of collecting and reporting KYC information, but must also be accepted by a wide variety of regulators in multiple countries. Nonetheless, the current very high cost and delays with remittances mean that there are significant benefits if these challenges can be met.



FINDING: BLOCKCHAIN MAY HELP REDUCE COST AND TIME OF REMITTANCES, BUT CHALLENGES REMAIN FOR SOLUTIONS TO KYC

Blockchain is a promising technology to reduce the cost and time of remittances, but significant challenges remain in finding solutions to KYC, and achieving acceptance of those solutions by regulators internationally.



Distributed trust may be critical in supporting the wide variety of participants in supply chain, as discussed earlier in our study. The blockchain can act as a kind of logically-centralised database of supply chain information, but can be geographically- and organisationally-distributed to match the structure of real-world supply chains. Data integrity in the historical log of events is key for creating provenance about individual shipments, and may improve supply chain quality overall. Logistics efficiency may also be improved by providing greater transparency on the status of shipments and processes which are currently often opaque. However, greater transparency is in tension with commercial confidentiality, and it is not yet clear how that tension will be resolved.

When supply chain information is available on a blockchain, there are many potential derived benefits. Visibility and data integrity for logistics and commercial documentation in the supply chain can provide evidence to manage risk, enabling trade finance and insurance applications.



FINDING: SUPPLY CHAINS ARE A HIGHLY PROMISING DOMAIN FOR THE APPLICATION OF BLOCKCHAIN TECHNOLOGY

Supply chains are a highly promising domain for the application of blockchain technology. Blockchains hold potential not just to integrate information exchange and improve operational efficiencies across a diverse industry, but also to improve supply chain quality, facilitate provenance for branded goods, and reduce the cost of regulatory approvals. However, research is required on issues around commercial confidentiality.



FINDING: SUPPLY CHAIN ON BLOCKCHAIN MAY ENABLE SIGNIFICANT OPPORTUNITIES FOR TRADE FINANCE AND INSURANCE

Trade finance and insurance are highly promising areas that may benefit if high-quality logistics and commercial supply chain documentation becomes available on blockchain.

6 DESIGN AND ASSURANCE OF BLOCKCHAIN-BASED SYSTEMS

A simple point, but often forgotten in the discussion of blockchain technology, is that a blockchain is almost never a whole system in itself. This simple point is foundational in the design blockchain-based systems.



FINDING: A BLOCKCHAIN IS USUALLY ONLY ONE COMPONENT OF A BROADER IT SYSTEM

Blockchains are usually combined with other components in a broader system. Functionality such as user interfaces, cryptographic key management, IoT integration, and communications with other external systems are all inherently off-chain. Many databases are also better stored off-chain, for scalability reasons (big-data), confidentiality reasons (private data), or for dealing with legacy databases.

The core goals of software engineering apply to blockchain-based systems:

- What are the requirements of blockchain-based systems?
- How should blockchain-based systems be designed to meet requirements?
- What evidence is sufficient to justify that a blockchain-based system will meet its requirements?

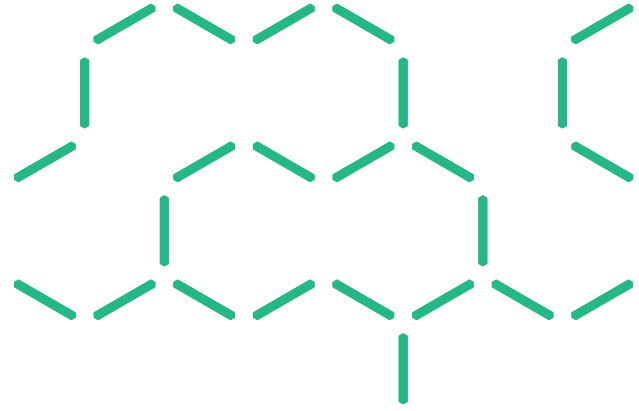
These questions are vital for the creation of dependable blockchain-based systems in regulated industries.

6.1 Design

Software design is a creative process, proposing and evaluating solutions to complex problems with many conflicting constraints. The final design of a software system is the result of many design choices about the selection, configuration, and integration of software, hardware, and communications components. For blockchain-based systems, design choices include the use of a blockchain instead of a traditional database, or the use of a private blockchain instead of a public blockchain. Often in a single blockchain-based system, some data may be stored on a blockchain while other parts of the data are stored and communicated using conventional computer systems, and this is another design choice. When using a blockchain, there are more detailed design choices about the type of blockchain, consensus protocol, block size and block frequency.

As with any software system, there are trade-offs between NFPs in the design of blockchain-based systems. Design decisions that improve the performance of one NFP for a system may harm the performance of other NFPs. Some simple examples of this include:

- Encrypting data before storing it on a blockchain may increase confidentiality, but will reduce performance, and may harm transparency or independent auditability.
- Storing only a hash of data on-chain and keeping the contents off-chain will improve confidentiality and may improve performance, but partly undermines the distinctive benefit of blockchains in providing distributed trust. This may create a single point of failure, reducing system availability and reliability.



- Using a private blockchain instead of a public blockchain may allow greater control over the admittance of processing nodes and transactions into the system, but will also increase barriers to entry for participation and thus partly reduce some of the benefit of using a blockchain.
- For blockchains that use Nakamoto consensus such as Bitcoin or Ethereum, waiting for a large number of confirmation blocks may increase confidence in integrity and durability of transactions, but will harm latency and thus may impact service availability.

For the deployment and operation of systems, there are a spectrum of options ranging from centralised monopolies, through to centrally-facilitated competition between parties, through to services provided jointly by consortia, through to fully open service provision in public peer-to-peer systems. It is possible that some components or functions are decentralised while others are centralised.

Software architecture is often supported by the use of models, which communicate design decisions to other software engineers, and which can be used to support

some kinds of quantitative evaluation of NFPs, using simulation or analytical approaches. For example, a recent paper [21], has used model-based architecture simulation to predict latency for a blockchain-based application. Benchmark measurements are taken of individual sources of latency, and these are combined with a system model to simulate the overall latency of the system. This can be used for ‘what-if’ analyses to evaluate design alternatives early in the design lifecycle, or to predict behaviour which may not be able to be tested completely prior to large-scale production usage.

However, models can also be used generatively, to automatically create working systems. In our supply chain use case we discussed how business process models can be automatically transformed into smart contracts on blockchain to perform inter-organisational process coordination. For our open data use case, we have used data schema models to automatically create smart contracts on blockchain to store and manage registries. A screen shot of a user interface for the definition of these data schema in a prototype system for generating registries on blockchain is shown in Figure 14.

The screenshot shows the Regeator interface with a dark blue header containing the 'DATA 51' logo, navigation tabs for 'Regeator', 'Design', and 'Manage', a user ID '0x1b437e922bf1e4eba430e5345cd6ebf0c9e0939a', and a 'Log Out' button. Below the header is a 'Create New Registry' button. The main content area displays three schema definitions in orange boxes:

- DatasetRegistry**
 - Attributes**
 - Record
 - <<PK>> id: uint32
 - owner: address
 - title: string
 - description: string
 - <<FK>> parent_datasets: uint32[]
 - records: mapping(uint32 => Record)
 - Functions**
 - + record_get_attr_title(record_id)
 - + record_get_owner(record_id)
 - + record_does_exist(record_id)
 - + record_get_attr_description(record_id)
 - + record_get_fk_parent_datasets(record_id)
 - + record_get_state(record_id)
 - + get_all_record_ids()
 - + record_delete(record_id)
 - + record_set_fk_parent_datasets(record_id, parent_datasets)
 - + record_set_attr_title(record_id, title)
 - + record_set_attr_description(record_id, description)
 - + record_create(record_id, title, description)
 - Events**
 - + records_history(record_id, owner, creation_timestamp)
 - + record_revisions_history(record_id, action_sig, revision_timestamp)
- FileRegistry**
 - Attributes**
 - + records: address[]
 - Functions**
 - + get_all_record_ids()
 - + admin_get_fk_external_refs()
 - + record_does_exist(record_id)
 - + record_set_fk_dataset_id(record_id, dataset_id)
 - + record_create(name, url)
 - + record_set_attr_name(record_id, name)
 - + record_set_attr_url(record_id, url)
 - + record_delete(record_id)
 - + admin_set_fk_referenced_registry_Dataset(Dataset_registry)
 - Events**
 - + records_history(record_id, owner, creation_timestamp)
- FileRecord**
 - Attributes**
 - <<PK>> id: address
 - owner: address
 - name: string
 - url: string
 - <<FK>> dataset_id: uint32
 - Functions**
 - + record_get_state()
 - + record_get_fk_dataset_id()
 - + managing_registry()
 - + record_get_attr_name()
 - + record_does_exist()
 - + record_get_owner()
 - + record_get_attr_url()
 - + record_set_attr_url(url)
 - + record_set_fk_dataset_id(dataset_id)
 - + record_set_attr_name(name)
 - + record_delete()
 - Events**
 - + record_revisions_history(action_sig, revision_timestamp)

Figure 14 Screenshot of a user interface for defining schema for the generation of blockchain-based registries

6.2 Assurance: Evidence and acceptance

In engineering, it is not enough to create a working system; we must also create evidence to back assurances that the system works. The conventional software engineering approach to establishing confidence in dependable software systems is to demonstrate compliance with well-defined development processes, and to perform systematic testing at multiple levels of design abstraction. However, because of software's complexity, these may only provide limited substantive evidence for assurance claims. [10]

In industry and academia, there is promising lines of work in the formal specification of smart contract requirements using logics for legal informatics [9], and in formal verification of smart contracts, using mathematical theories to reason about their possible behaviour. This work at the level of smart contracts can be complemented by work on formal verification of blockchain protocols. These kinds of approaches may provide much stronger evidence about the functional properties of blockchain-based systems.

For NFPs, model-based approaches can be used the analysis and simulation of architectural designs, to quantitatively predict system performance. These techniques require empirical validation and calibration for specific blockchain platforms, but may be able to sufficient predictive accuracy to give reasonable confidence about qualities such as latency, throughput, execution cost, and service availability.

Finally, these pieces of evidence must be integrated into overall assurance arguments with systematic links to regulatory requirements in domains that need trustworthy blockchain-based systems. Research is required to extend all of these approaches from their use for conventional technologies to blockchains.



RECOMMENDATION: INCREASE R&D ON TRUSTWORTHY BLOCKCHAINS

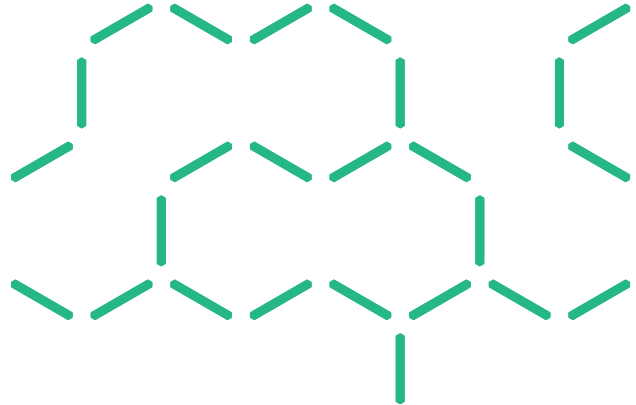
Industry and academia should increase research and development of approaches to better justify assurances about the functional and non-functional properties of blockchains and blockchain-based systems. This should include work towards formal specification and verification of smart contracts and blockchain protocols, and design-time architectural analyses of key NFPs.

It is not yet well-understood how to deal with failures in the use of blockchain-based systems. Most proof of concept blockchain-based system deployments to date have only demonstrated 'sunny day scenarios' (also known as 'happy paths'), where no error or exception occurs. However, in real world blockchain systems, problems may arise such as disputed transactions, incorrect addresses, exposure or loss of private keys, data-entry errors, or unexpected changes to assets tokenised on blockchain. The immutability of blockchain ledgers may make them less adaptable than conventional technologies controlled by trusted third party organisations. However, proof of concept demonstrations of blockchain-based do not yet commonly explore the cause and resolution of 'rainy day scenarios'. Some problems may be anticipated problems, with resolution mechanisms built-in to smart contracts or surrounding infrastructure. Other problems may be unanticipated, but mechanisms for resolution outside the blockchain will still be required.



RECOMMENDATION: TEST BLOCKCHAINS IN THE RAIN

Field trials of blockchain-based systems should not just demonstrate feasibility of 'sunny day scenarios' in the normal successful use of those systems, but should also demonstrate responses to 'rainy day scenarios' arising from both anticipated and unanticipated problems in the use of those systems.



We have discussed in this report how blockchains introduce risks to various NFPs that are critical in many use cases. Software designs must resolve trade-offs for these NFPs to deal with business and regulatory requirements. When considering evidence of whether blockchain-based systems are trustworthy, we should pay particular attention to these risks.



RECOMMENDATION: SCRUTINISE TECHNOLOGY-SPECIFIC RISKS FOR NEW SYSTEMS

Regulators and enterprise should be aware of the typical technical risks and limitations of blockchain technologies, and pay particular attention to ensure that proponents for new blockchain-based systems provide sufficient evidence that the new systems meet requirements related to those risks and limitations.

Even if we develop a blockchain-based system that meets its requirements, and even if we provide evidence that the system meets its requirements, a further hurdle in large enterprises and in regulated industries is for this evidence to be accepted as sufficient to address compliance obligations and regulatory risks of concern. Industry needs to be able to properly target the creation of evidence to satisfy regulatory requirements. What will satisfy a regulator for a proposed system? Can industry be confident that the evidence they have prepared would be adequate? The Australian Securities and Investments Commission (ASIC) has recently released an information sheet and pointers to guidance for the evaluation of DLT [2]. This is an early example of the kind of guidance that will help industry create marketable products and services. The guidance covers at a high level some of the technical issues discussed in this report, but also identifies other issues such as how new products and services based on blockchain or DLT will work under the law. Similarly, the analytical framework proposed by the Bank for International Settlements [7] identifies a range of questions to be considered for DLT – not just for technical issues for key NFPs, but also concerning implications for the broader financial system.



RECOMMENDATION: PROVIDE INDICATIVE GUIDANCE ON SUFFICIENT EVIDENCE FOR REGULATORY ACCEPTANCE OF BLOCKCHAIN-BASED SYSTEMS

Regulators should provide indicative guidance on how they will evaluate what constitutes sufficient evidence that a new system meets regulatory requirements.

How should enterprise policies and regulatory requirements be expressed? In order not to reduce opportunities for innovation, these constraints should be technologically-neutral. Instead of mandating or prohibiting specific technological solutions, risks should be treated through requirements on NFPs for systems. For example, rather than imposing a blanket prohibition on public blockchains, constraints could be imposed on policies for admission of participants or transactions. An exception to this general principle may be to mandate the use of de jure standards to support interoperation and reduce barriers to entry. However even then these standards should ideally serve only to constrain interfaces rather than mandate specific implementations.



RECOMMENDATION: TECHNOLOGICALLY-NEUTRAL REGULATION AND POLICY

Regulators and enterprise should be technologically neutral in framing the criteria for acceptance of a system in their domain. That is, there should be no regulation nor prohibition for blockchain technology specifically, even for public blockchains. Similarly, there should be no requirement for the use of a specific non-blockchain technology.

6.3 Non-functional properties

Blockchain systems emerged to support a financial transaction (digital currency) system, and so it is not surprising that major NFPs are those that are critical in that domain: integrity and non-repudiation (including immutability of data, and transparency). As a highly-distributed and redundant data store, blockchain systems can also support high levels of availability for reading data. As discussed earlier, there are some well-known limitations on NFPs for blockchain systems. Some are inherent to the technology, but others are only current limitations and may well be overcome in the near future. We discuss a variety of NFPs below.

6.3.1 COST

An important NFP is the monetary cost of implementing and operating systems. Blockchains are massively distributed with many redundant processing nodes, and provide data integrity about their full transaction history. This inevitably impacts the cost of using blockchain, and means that blockchain has a different cost model than conventional (cloud or in-house) infrastructure.

For example, a recent study compared the cost of executing business process on blockchain with cloud-based process execution. [13] Blockchain was two orders of magnitude more expensive than cloud. However, blockchain storage retention time is paid once-off in a transaction fee for small pieces of transaction and event data, whereas storage in cloud requires ongoing monthly fees. For financial transfer, blockchain-based digital currency can have much lower fees than conventional money transfers.

The rate at which blocks can be created is limited, often by using a proof of work mechanism, whereby a processing node can only add a new block by demonstrating that a difficult task has been completed.



FINDING: BLOCKCHAINS HAVE A DIFFERENT COST MODEL

Blockchains have a different cost model than conventional technologies. For digital currency transfer, blockchains might be cheaper than conventional fiat currency transfer. (This may be supported by avoiding AML/CTF costs associated with KYC information.) Blockchains have a low one-time fee for permanent storage of small pieces of transaction and event data. However, blockchains have much higher cost for execution of programs (smart contracts) than on conventional (cloud or in-house) infrastructure.

6.3.2 PERFORMANCE

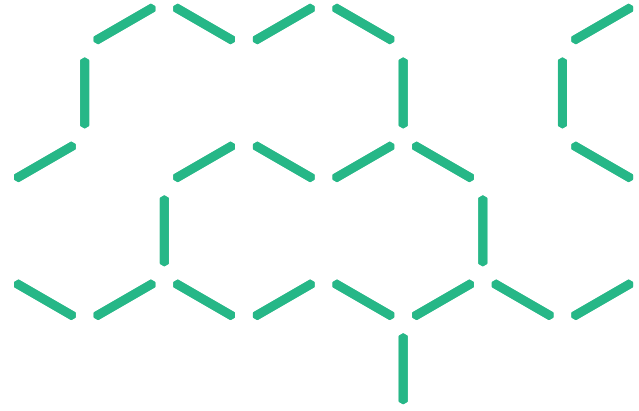
Time-related ‘performance’ means a variety of things for software systems. *Throughput* is the total number of transactions a system can process within a time window, whereas *latency* is the time required to respond to a single transaction. A system with low latency does not necessarily have high throughput. That can for example also depend on the *scalability* of the system, which concerns the relationship between resource utilisation and performance as demand on the system grows. We have briefly discussed scalability in the section on blockchain myths, above. We discuss read and write latency below.

Read latency

When data has previously been written to the blockchain, *read latency* is the response time for accessing historical data from a blockchain client. Read latency can be much faster on blockchain than with conventional technologies, because clients can keep a full local copy of the database, and so there are no network delays.

To illustrate this, we performed a simple experiment⁸ to compare the read latency of blockchain with web-based access of a remote API. The average read latency from a local blockchain node was 2.0ms, and from a comparable remote web API was 43.6ms.

⁸ The read latency of blockchain is tested on a local node that connects to the public Ethereum blockchain. The DAO smart contract at address 0xBB9bc244D798123fDe783fCc1C72d3Bb8C189413 was used for benchmarking, specifically the function *balanceOf(address)* which returns the balance of an account from a contract HashMap-like storage. We used web3 0.17.0 library as an Ethereum node connector. The read latency for accessing a remote API was tested on the same machine, accessing a RESTful API from data.gov.au. Both blockchain and RESTful API have comparable small size of response that is comparable.



Write latency

The request to write data into a blockchain is done by sending a transaction to the network. The *write latency* is a probabilistic, and there are several sources of uncertainty. All blockchains will have small network delays. Also, for blockchains with Nakamoto consensus, one cannot be highly confident that the most recently-included block will still be included later on. To increase our confidence that data has successfully been committed to the blockchain, we can wait for a number of ‘confirmation blocks’. Waiting for more confirmation blocks will increase write latency. We call ‘inclusion time’ the time at which we see a transaction included in a block and ‘commit time’ the time at which we have seen a pre-defined number of confirmation blocks.

To illustrate this, we have performed a simple experiment to test write latency on the Ethereum public blockchain. Figure 15 depicts the distribution of the time it can take for an Ethereum transaction to be included in a block for the first time, as observed in a trace collected in a local client node of Ethereum. The data shown in the diagram is based on a 10-day observation period of Ethereum. This node listens and collects transaction and block announcements. The blue line represents the cumulative distribution function (CDF) of the period between the time we observed transactions being announced and the time we observed them as committed (after 12 blocks confirmations, which is often recommended for Ethereum). While this is known to be 3 minutes on average, we can see that there is a long tail of long write latencies.

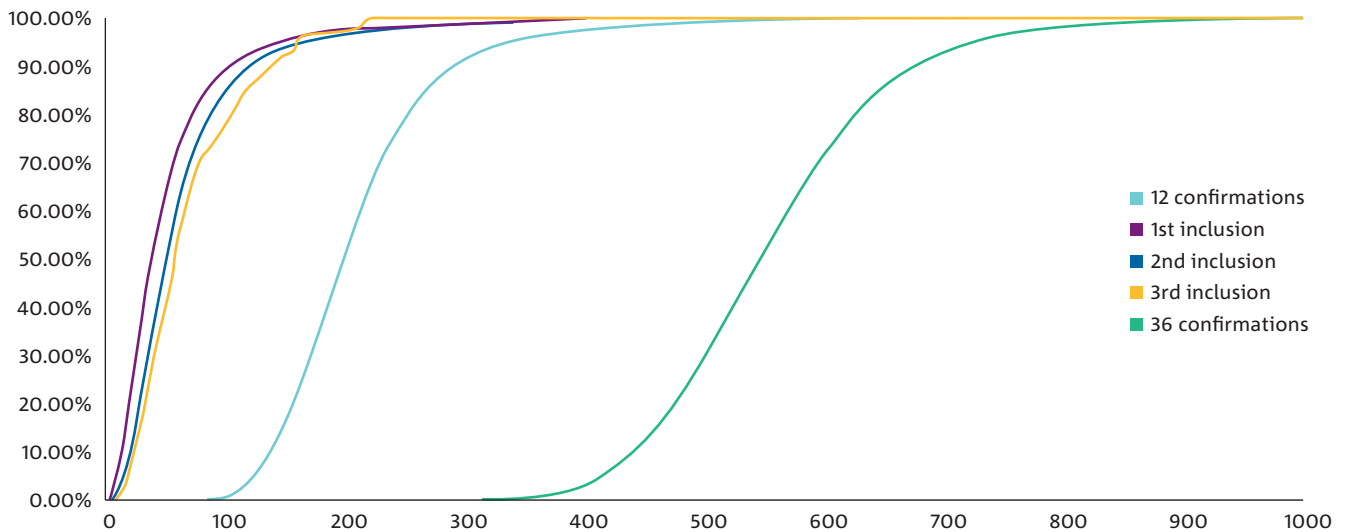


Figure 15 Time to commit transactions in Ethereum blockchain

6.3.3 SECURITY

The classic security properties are Confidentiality, Integrity, and Availability. [1]

Confidentiality

Confidentiality means that unauthorised disclosure of information does not take place [3]. This is usually harder to establish in blockchain-based systems, because the default is that information is visible for everyone in the network. Information can be encrypted: asymmetrically with a particular party's public key, so that only this party can decrypt it; or symmetrically with a shared secret key, so that the group of parties with access to the secret key can decrypt it. The latter case requires a secure means of exchanging the secret key off-chain.

However, once information needs to be processed by smart contract methods, this information needs to be decrypted. This is because smart contract code runs on all nodes of the network, and thus any of them needs to be able to process the input data. This is required to achieve consensus on the outcomes of smart contract execution. Embedding keys within a smart contract would reveal the key to all participants.

As discussed in the supply chain use case, commercially sensitive data can be at risk if it is shared on a blockchain, even if pseudonyms are used, and even if encryption is used. Private blockchains operated over private networks can provide some level of access control, but this will not provide commercial confidentiality between competitors jointly operating a private blockchain. Some emerging blockchains (e.g., Corda, Fabric) can provide finer-grained read access control, but only achieve global integrity by re-introducing reliance on third-party organisations as notaries.

There are interesting technologies on the horizon, which could alleviate some of these pain points. For instance, zero knowledge proof methods like zk-Snark can be used to hide the contents of a transaction, while still allowing independent validation of the integrity of that transaction. Current implementations (like zCash) are limited to hiding simple transfers of cyber currency, but in future more sophisticated transactions may be able to be kept private. As for computation on encrypted data, that is the goal of techniques like homomorphic encryption and confidential computing. However, such approaches have not been utilised for smart contracts as yet, in part due to their significantly increased computational requirements over regular computation.



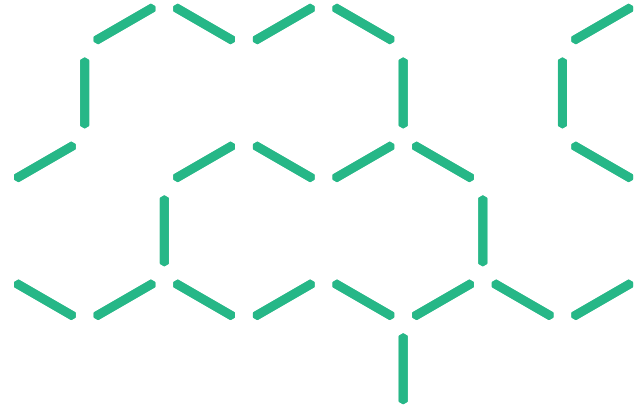
FINDING: PRIVATE BLOCKCHAINS ARE OFTEN NOT PRIVATE ENOUGH

Many private blockchains share information among all participating nodes. This is OK for a fully vertically-integrated solution, but if competitors are present on the same blockchain, they may be able to discover information that is normally held commercial-in-confidence. Some alternative private blockchains instead restrict sharing of information to only interested parties, but these blockchains typically re-introduce individual trusted third parties into the operation of the blockchain.

Integrity

Integrity is the absence of improper (invalid or unauthorised) system alterations [3] and is a key attribute for blockchains. Once a transaction is included in a blockchain and committed for enough time, it becomes part of the effectively-immutable ledger and cannot be altered. This also applies to smart contracts: their bytecode is deployed in a transaction, and thus is subject to the same integrity guarantees. The key integrity property of Bitcoin is that addresses cannot spend money they don't have. Ethereum's integrity property is more complicated, because it requires the correct operation of a Turing-complete smart contract programming language. However, for client applications, Ethereum provides significant power by allowing user-defined integrity conditions to be implemented as checked preconditions and defined behaviours in smart contracts.

Blockchain emerged to support a crypto currency, and so it is unsurprising that integrity is a key dependability attribute, because integrity is the key dependability attribute for commercial computer security. The seminal work is the Clark-Wilson security policy model [6], and blockchains are broadly consistent with its requirements. Smart contracts can implement Clark-Wilson's Transformation Procedures to generate and update internal data or other smart contracts that realise Clark-Wilson's Constrained Data Items. Blockchains natively create the log required by Clark-Wilson for reconstructing operations. Finally, blockchains use a kind of separation of duty through the replicated validation performed by all mining nodes.



Ethereum smart contracts are written in a Turing-complete programming language. This makes it more difficult to verify that the smart contracts correctly implement required integrity properties. Some blockchains, such as Kadena and Corda [4], avoid the use of Turing-complete smart contract languages for this reason, and instead use less-expressive domain-specific languages that can be automatically checked. Smart contract languages with strong typing mechanisms (such as the Pact language on the Kadena blockchain) can also help programmers enforce some integrity constraints.

High integrity and non-repudiation is not always ideal. For example, sometimes historical data must be deleted or changed. If a vexatious or improper registry entry has been created, a court may order the registrar to change the registry to remove that entry, 'as if it had never been created'. This is not technically possible on many blockchain platforms. Similarly, blockchains might be 'poisoned' by illegal content. Some blockchains have been proposed to deal with this challenge, but there is not yet widespread acceptance about good solutions. Alternatively, if blockchains only store hashes of data stored off-chain, then traditional data management can deal with court-ordered data deletion if required, even though the hashes might never be able to be deleted.



FINDING: SOMETIMES TOO MUCH INTEGRITY CAUSES PROBLEMS

It is not possible to change the transaction history in most blockchains. This is normally a good thing, but can cause problems if blockchain contains illegal content, or if a court orders content to be removed from the blockchain.

Availability and reliability

Availability is the readiness for correct service, whereas reliability is the continuity of correct service. [3] More specifically in the context of blockchain-based systems, availability concerns the ability to invoke functions of the system, whereas reliability refers to receiving consistently correct outcomes from those invocations.

The operation of public blockchains can involve hundreds or thousands of independent processing nodes. Each node holds a full replicated instance of the blockchain transaction history and can operate for users as a transaction interface to the blockchain network. Because of this massive redundancy, naively we might expect that a blockchain system has extremely high availability. We can assume that local components of a blockchain-based system are connected to a local full node on the blockchain network. Submitting a transaction to a blockchain network is done through the local full node, which broadcasts that to all nodes it is connected to. The availability of a locally-reachable full node is thus heavily reliant on thus the organisation operating a blockchain-based system. The more complex question is: how certain can one be that the transaction is included in a block and confirmed, in a timely manner?

For blockchains, there are circumstances in which the distinction between reliability and availability can be blurred as there is no globally-specified time by which a transaction should complete. If a blockchain system never includes a transaction, that will be both an availability and reliability failure of the blockchain system from the perspective of a client application. However, if a transaction is initially included in some block, that does not guarantee that block will be recognised as being part of the blockchain in future. One could take the following view: first an application designer can specify a number of confirmation blocks by which they will regard a transaction to have been committed. If a fork happens after that number of confirmation blocks, the system will have had a reliability failure, because a transaction thought to have been committed will have turned out not to be. Alternately, if a fork happens prior to the specified number of confirmation blocks, the system may experience enough delay to have an availability failure.

Transactions deploying smart contracts or invoking their methods add a further level of complexity. First, they are subject to more parameters, like the current gas limit in Ethereum, that may impact their successful inclusion. Second, they utilise more complex functionality of the network, and thus rely on the network sharing the same accepted norms about this functionality with the system.

6.3.4 MAINTAINABILITY

Maintainability refers to a system's amenability to undergo modifications and repairs [3]. In blockchain-based systems that use smart contracts, this is harder to implement for the smart contracts than in regular distributed systems. This is because smart contracts comprise code that regulates the interactions between mutually untrusting parties; trust is derived from the fact that the code cannot be changed easily. Consider an example where an organisation has established trust in the code of a particular contract, and verified that it implements the agreed rules for handling crypto-coins. If others can change the code without that organisation's knowledge or consent, any trust in the code would be void. Although the code of an Ethereum smart contract cannot be changed, the current state of variables within that smart contract can be updated by invoking its methods. In particular, these variables may refer to other smart contracts. This mechanism provides a kind of indirection that allows the dynamic updating of smart contract code. However, support for this kind of updating must be specifically provisioned ahead of time.

Finally, changes may be made to a blockchain-based system not by changing the data stored on a blockchain, but instead by changing the interpretation of data on the blockchain. As an extreme example, a client application might choose to not honour all data previously written to the blockchain under some previously-acknowledged addresses. Instead, the client could in principle re-create all required data on the blockchain under some new address.

A distinctive benefit of blockchain-based systems is that there is no single party with control of the system. However, this inherently creates challenges for governance: the management of the evolution of blockchain-based systems. Changes may be made to correct defects, add features, or migrate to new IT contexts. However, in a multi-party system with no single owner, managing these changes is more like diplomacy than traditional risk management or conventional technical change management or product management. Lessons may be drawn from open-source software, which face similar development challenges. However, the governance of a blockchain is not just a software development problem – it is also a deployment and operations problem. For both public and private blockchain systems, key stakeholders include the users of the blockchain, software developers with moral or contractual authority over the code base, miners or processing nodes in the blockchain ecosystem, and government regulators in related industries. There are still lessons being learned about who are the key stakeholders for blockchains: the 2016 hard fork of Ethereum in response to the DAO controversy made it apparent in hindsight that digital currency exchange markets are a key stakeholder for public blockchain systems. (The market provided by the Poloniex exchange for trading the unforked 'Ethereum classic' digital currency has supported the ongoing operation of that blockchain, which might have otherwise failed to continue to be viable.)



FINDING: THERE ARE OPEN QUESTIONS ABOUT BLOCKCHAIN GOVERNANCE

It is a challenge and currently unknown how to best perform governance for blockchains and blockchain-based systems. How should relevant stakeholders influence and manage changes to the software and the operational infrastructure for blockchains and blockchain-based system, when there might be no central owner, and where the blockchain platform might be serving many purposes for different stakeholder groups?

7 LIMITATIONS OF THIS STUDY



Blockchain technology is still in its infancy and under active development, so our study is inevitably an early one in a domain that is still not yet well understood. Our study has a number of possible limitations which we acknowledge here.

We have used only a small number of illustrative use cases (three). However, we use an exploratory, qualitative approach and do not make claims that rely on statistical evidence about the populations of use cases.

The selected use cases may not adequately cover nor be representative of good uses for blockchain. We have tried to mitigate this risk by seeking broad input about candidate use cases from the literature and from workshop participants, and by explicitly identifying selection criteria to support our study objectives.

We have performed only limited validation of the context and requirements for our selected use cases, using literature, industry reports and discussion with some domain experts. We believe we have captured enough relevant information for the use cases to explore architectural evaluation of technical risks and opportunities for blockchain-based systems in these domains.

The design candidates we have presented are highly simplified, and would not be sufficiently detailed to use in the implementation of a system nor for use in a thorough assessment of risks associated with regulatory compliance. However, they are detailed enough to for our purposes: to illustrate design alternatives and provide a basis for indicative qualitative comparisons between them.

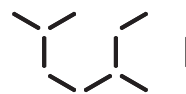
The design candidates we have presented for the use cases may not be optimal in their use of blockchain technologies. It is possible that alternative architectures exist that better address the key NFRs for each use case. However, we believe our approach is reasonable in proposing simple designs that make straightforward use of blockchain technologies, to reveal risks and opportunities that may be commonly encountered in this early stage of blockchain technology development.

The design analyses we have performed on design candidates may not be valid, because they are yet to be widely used and studied for blockchain-based systems. However, the high-level qualitative approaches we employ have been previously used in a variety of other technology domains, so we believe it is reasonable to use them to support the indicative qualitative findings in our study.

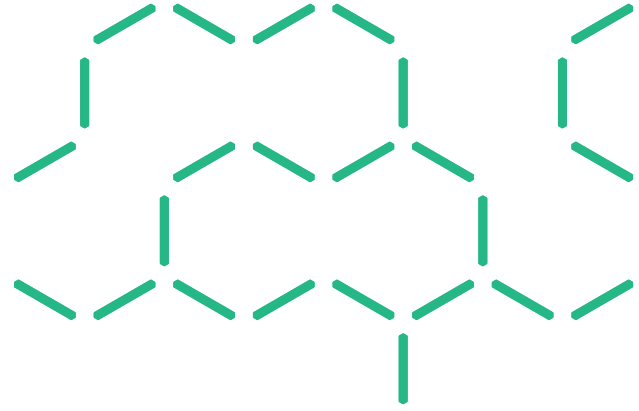
Blockchain technologies are under active development globally, and there may be recent advances that impact our findings. To mitigate this, we have endeavoured to follow advances in blockchain technologies by monitoring international technology conferences, published academic papers, and grey literature (such as white papers, and blogs).

APPENDIX A

FINDINGS AND RECOMMENDATIONS



- 1 Regulators and enterprise should be technologically neutral in framing the criteria for acceptance of a system in their domain. That is, there should be no regulation nor prohibition for blockchain technology specifically, even for public blockchains. Similarly, there should be no requirement for the use of a specific non-blockchain technology.
- 2 Even in highly-regulated industries, public blockchain systems may be appropriate for use for some limited purposes, such as public announcements, product catalogues, software update checking, or fully-public government registries.
- 3 Regulators and enterprise should be aware of the typical technical risks and limitations of blockchain technologies, and pay particular attention to ensure that proponents for new blockchain-based systems provide sufficient evidence that the new systems meet requirements related to those risks and limitations.
- 4 Regulators should provide indicative guidance on how they will evaluate what constitutes sufficient evidence that a new system meets regulatory requirements.
- 5 It is a challenge and currently unknown how to best perform governance for blockchains and blockchain-based systems. How should relevant stakeholders influence and manage changes to the software and the operational infrastructure for blockchains and blockchain-based system, when there might be no central owner, and where the blockchain platform might be serving many purposes for different stakeholder groups?
- 6 Field trials of blockchain-based systems should not just demonstrate feasibility of 'sunny day scenarios' in the normal successful use of those systems, but should also demonstrate responses to 'rainy day scenarios' arising from both anticipated and unanticipated problems in the use of those systems.
- 7 Blockchains are usually combined with other components in a broader system. Functionality such as user interfaces, cryptographic key management, IoT integration, and communications with other external systems are all inherently off-chain. Many databases are also better stored off-chain, for scalability reasons (big-data), confidentiality reasons (private data), or for dealing with legacy databases.
- 8 Industry and academia should increase research and development of approaches to better justify assurances about the functional and non-functional properties of blockchains and blockchain-based systems. This should include work towards formal specification and verification of smart contracts and blockchain protocols, and design-time architectural analyses of key NFPs.
- 9 Supply chains are a highly promising domain for the application of blockchain technology. Blockchains hold potential not just to integrate information exchange and improve operational efficiencies across a diverse industry, but also to improve supply chain quality, facilitate provenance for branded goods, and reduce the cost of regulatory approvals. However, research is required on issues around commercial confidentiality.



10

Blockchains and smart contracts may make possible a new form of ‘programmable money’, where policies can be attached to how parcels of currency are spent.

11

Blockchain is a promising technology to reduce the cost and time of remittances, but significant challenges remain in finding solutions to KYC, and achieving acceptance of those solutions by regulators internationally.

12

Trade finance and insurance are highly promising areas that may benefit if high-quality logistics and commercial supply chain documentation becomes available on blockchain.

13

Blockchains have a different cost model than conventional technologies. For digital currency transfer, blockchains might be cheaper than conventional fiat currency transfer. (This may be supported by avoiding AML/CTF costs associated with KYC information.) Blockchains have a low one-time fee for permanent storage of small pieces of transaction and event data. However, blockchains have much higher cost for execution of programs (smart contracts) than on conventional (cloud or in-house) infrastructure.

14

Many private blockchains share information among all participating nodes. This is OK for a fully vertically-integrated solution, but if competitors are present on the same blockchain, they may be able to discover information that is normally held commercial-in-confidence. Some alternative private blockchains instead restrict sharing of information to only interested parties, but these blockchains typically re-introduce individual trusted third parties into the operation of the blockchain.

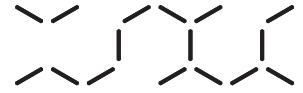
15

It is not possible to change the transaction history in most blockchains. This is normally a good thing, but can cause problems if blockchain contains illegal content, or if a court orders content to be removed from the blockchain.

SHORTENED FORMS

AML	Anti-Money Laundering
API	Application Programming Interface
CTF	Counter-Terrorism Financing
DAO	Distributed Autonomous Organisation
DLT	Distributed Ledger Technology
IoT	Internet of Things
KYC	Know Your Customer
MTO	Money Transfer Operator
NFP	Non-Functional Property
NFR	Non-Functional Requirement

GLOSSARY OF TERMS



Anti-Money Laundering and Counter Terrorism Financing

Anti-Money Laundering and Counter Terrorism Financing (AML/CTF) legislation and regulation aims to prevent money laundering and the financing of terrorism by imposing a number of obligations on the financial sector, gambling sector, remittance (money transfer) services, bullion dealers and other professionals or businesses that provide particular regulated services.

Application Programming Interface

An Application Programming Interface (API) is a technical interface to a web service or programming language library that exposes functions or methods in the interface to be able to be invoked by clients using a programming language.

Bitcoin

Bitcoin is a peer-to-peer payment system invented by an unidentified programmer, or group of programmers, under the name of Satoshi Nakamoto.

Block

A block in a blockchain is the container of transactions. Each block contains a timestamp and a link to the previous block.

BFT Consensus

Byzantine-Fault-Tolerant (BFT) Consensus: a mechanism for achieving fault-tolerant consensus. It has stronger consistency guarantees than Nakamoto consensus, but requires a known and smaller maximum number of participants.

Consortium blockchain

A private blockchain deployed and operated by a consortium of organisations.

Digital currency

Digital currency is an Internet-based form of currency or medium of exchange distinct from physical (such as banknotes and coins) that exhibits properties similar to physical currencies, but allows for instantaneous transactions and borderless transfer-of-ownership.

Distributed Autonomous Organisation

A Decentralised Autonomous Organisation (DAO) is a special kind of Smart Contract. A DAO is code that operates as a decentralised autonomous business model for organising both commercial and non-profit enterprises (generally in a financial way). The legal status of a DAO is uncertain. A specific DAO called 'The DAO' ran as an investment vehicle on Ethereum in 2016, but failed because of poorly-understood smart contract code.

Ethereum

Ethereum is a public blockchain-based distributed computing platform, featuring smart contract functionality. It provides a decentralised virtual machine, the Ethereum Virtual Machine (EVM), which can execute peer-to-peer contracts using a token called ether.

Functional requirement

In Software engineering and systems engineering, a functional requirement defines the observable input/output behaviour of a system or its component, for example, calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

Gas

As smart contracts execute, they use computational resources in many participating nodes. To limit resource utilisation and compensate for the use of these resources, some blockchains such as Ethereum charge 'gas' for the execution of smart contracts. Gas is usually paid for with the blockchain's digital currency. More demanding smart contracts use more gas, and blockchains may impose a limit on the amount of gas that can be used per transaction or per block.

Hash

A hash function is any function that can be used to map data of arbitrary size to data of fixed size. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes. Hashes are non-reversible: if some data is hashed, it is impossible to recover the data from the hash value alone. However, if the data can be guessed, then its hash values can be used to confirm that the data was originally hashed.

Immutable

Not able to be changed. Blockchain data cannot in practice be easily changed because it is continually replicated across many different locations and organisations. Blockchains are tamper-evident. Attempts to change it in one location will be interpreted as fraudulent and an attack on integrity by other participants, and will be rejected.

Internet of Things

The internet of things (IoT) refers to devices, sensors, motors, and other electronics connected to the internet. The rising rate of computational power in parallel with their falling cost foreshadows a potentially significant and increasing trend of adoption, with many billions of devices expected to be deployed in the coming years.

Interoperability

The ability of a system to work effectively with other systems. This will typically involve sharing or accessing data or services, through defined interfaces.

Know your customer

Know your customer (KYC) is the process of a business identifying and verifying the identity of its clients. The term is also used to refer to the bank regulation which governs these activities.

Miner

In a Proof of Work blockchain, the processing nodes which collectively operate the blockchain are known as miners.

Nakamoto Consensus

The Nakamoto Consensus mechanism is used in Bitcoin and other blockchain systems. When there are multiple alternative versions of the blockchain ledger, the Nakamoto Consensus mechanism favours the longest chain.

Non-Functional Property

Non-Functional Properties are criteria that can be used to judge the performance of a system, and include performance, scalability, and security.

Non-Functional Requirement

In systems engineering and requirements engineering, a non-functional requirement is a requirement for a Non-Functional Property. These requirements specify criteria about the performance of a system. They are contrasted with functional requirements.

Non-repudiation

The inability to deny a previous claim. On a blockchain, the immutability of historical transactions which are cryptographically signed means that there is always strong evidence that those transactions were performed by someone with control over those cryptographic keys.

Nostro/vostro accounts

Conventionally, a bank does not have a ledger jointly-shared with another bank. Instead they create internal accounts intended to mirror the accounts held at the other bank. The other bank holds dual sets of accounts, and these 'nostro/vostro' accounts can be periodically reconciled to check and maintain consistency between the two banks. A jointly-shared distributed ledger, perhaps implemented using a blockchain, is an alternative to this conventional approach.

Oracle

An oracle is a trusted person or program that creates a record on the blockchain about some event or condition in the external world.

Proof of Concept

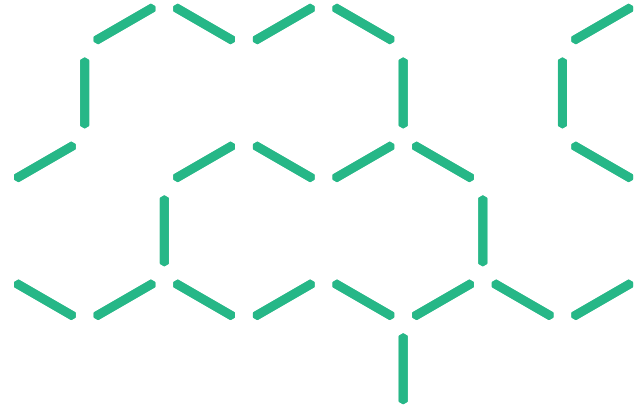
Proof of concept (POC) is a realisation of a certain method or idea in order to demonstrate its feasibility or a demonstration in principle with the aim of verifying that some concept or theory has practical potential.

Proof of Stake

A proof-of-stake (POS) is a type of consensus protocols used by blockchain systems, where the probability of mining a block is dependent on varies how much digital currency is controlled by the miners.

Proof of Work

Proof-of-Work (POW) is a type of consensus protocols used by blockchain systems, where the probability of mining a block is dependent on how much work is done by the miners.



Private blockchain

A blockchain operated by a private entity or consortium, with no or limited access by other parties, and typically with a small number (tens or hundreds) of processing nodes operating the blockchain. In this context, compared to public blockchains, technical optimisations may be used to improve the latency and throughput of the blockchain, and BFT consensus mechanisms may be used to provide stronger guarantees about the completion of transactions.

Private Key

See Public Key.

Public blockchain

A blockchain operated as a public peer-to-peer system. Parties are usually identified by pseudonymous public/private keys, and a form of Nakamoto consensus is typically used to allow a large number (thousands) of processing nodes to operate the blockchain.

Public Key

In cryptography a public key is a published number which is used as a parameter in an encryption function, to encrypt and check signed messages. Public keys are paired with secret private keys, which are used to decrypt and sign messages.

Security

Security (information security, cybersecurity) is a collection of Non-Functional Properties, which classically include Confidentiality, Integrity, and Availability, but can also include properties such as Privacy, Non-Repudiability.

Sharding

Sharding is a technique of breaking apart a database or blockchain into separate independent pieces. If the pieces are truly independent, they can be processed concurrently, which can significantly increase the throughput of the overall system.

Smart contract (blockchain)

In blockchain technology, a smart contract is a program that is recorded on the blockchain ledger and executes as part of transaction validation on the blockchain. In addition to executing the logic encoded in the program, smart contracts can carry digital currency or control access to other digital assets or tokens recorded on the blockchain. Some blockchains allow smart contracts to be arbitrary Turing-complete programs, while other blockchains only allow more limited programs.

Smart contract (legal informatics)

Smart contracts are computer programs that facilitate, verify, or enforce the negotiation or performance of a legal contract.

State channels

State channels are a design pattern for the use of smart contracts to adjudicate on the completion of an off-chain protocol. Participants first jointly commit to this smart contract. Then they exchange a series of messages off-chain about which may be too confidential or rapid or large to perform on the blockchain. The final state of the off-chain exchange is submitted back to the smart contract, which resolves final exchange of assets on the blockchain.

Trusted

In dependable systems, being trusted means being relied upon to achieve some purpose.

Trustworthy

In dependable systems, being trustworthy is the quality of having good evidence for being dependable.

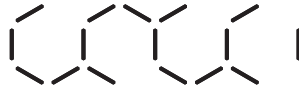
Turing complete

In computability theory, an instruction set or a programming language is said to be Turing complete or computationally universal if it can simulate a Turing machine. The Church-Turing thesis is that all such languages have equivalent computational power. Programs in a Turing-complete language can be arbitrarily complex, and there is no mechanism to automatically determine the correctness of all programs in a Turing-complete language.

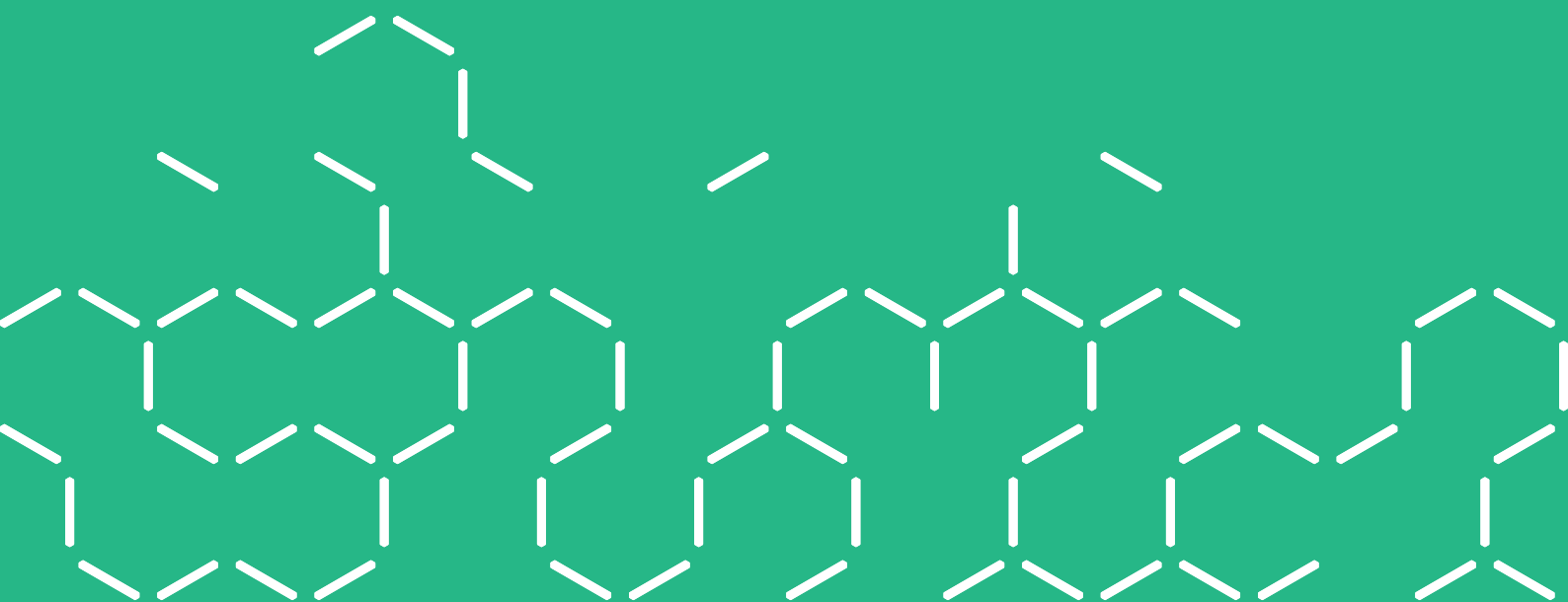
Use case

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (or actor) and a system, to achieve a goal.

REFERENCES



1. R. Anderson. "Security engineering". John Wiley & Sons, 2nd edition, 2008.
2. Australian Securities and Investments Commission (ASIC), "Evaluating distributed ledger technology", Information sheet INFO 219, March 2017.
3. A. Avizienis. J.-C.. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, Jan 2004.
4. L. Bass, P. Clements, and R. Kazman. "Software Architecture in Practice". Addison-Wesley Professional, Boston, MA, USA, 3rd edition, 2012.
5. R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn. "Corda: An Introduction". White paper, August 2016.
6. D. D. Clark and D. R. Wilson, "A comparison of commercial and military computer security policies," in IEEE Symposium on Security and Privacy, 1987, pp. 184–184.
7. Committee on Payments and Market Infrastructure, "Distributed ledger technology in payment, clearing and settlement: an analytical framework", Bank for International Settlements, Feb. 2017.
8. FinTech Network, "Four Blockchain Use Cases for Banks", 2017.
9. F. Idelberger, G. Governatori, R. Riveret, and G. Sartor. "Evaluation of Logic-Based Smart Contracts for Blockchain Systems". In Proc. RuleML, 2016, pp. 167-183.
10. D. Jackson, M. Thomas, and L. I. Millett (Eds), "Software for Dependable Systems: Sufficient Evidence?", The National Academies Press, 2007.
11. S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". <https://bitcoin.org/bitcoin.pdf> 2008.
12. C. Natoli, and V. Gramoli, "The Blockchain Anomaly". In Proc IEEE 15th International Symposium on Network Computing and Applications (NCA), 2016, pp. 310-317.
13. P. Rimba, A. B. Tran, I. Weber, M. Staples A. Ponomarev, X. Xu (2017) "Comparing Blockchain and Cloud Services for Business Process Execution". In Proc. International Conference on Software Architecture (ICSA 2017).
14. N. Szabo "Formalizing and securing relationships on public networks". First Monday, vol 2, no. 9, 1 Sep 1997.
15. I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling. "Untrusted business process monitoring and execution using blockchain". In Proc. International Conference on Business Process Management, Rio de Janeiro, Brazil, September 2016.
16. G. Wood. "Ethereum: A secure decentralized generalised transaction ledger – homestead draft". Technical report, 2016.
17. The World Bank Group, "Migration and Remittances Factbook", Third Edition, 2016.
18. The World Bank Group, "Remittance Prices Worldwide: Making Markets More Transparent" <https://remittanceprices.worldbank.org/en/countrycorridors>, 2016
19. X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, and S. Chen. "The blockchain as a software connector". In Proc. 13th Working IEEE/IFIP Conference on Software Architecture, 2016.
20. X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba. "A taxonomy of blockchain-based systems for architecture design". In Proc. IEEE International Conference on Software Architecture, April 2017.
21. R. Yasaweerasinghelage, M. Staples, and I. Weber. "Predicting latency of blockchain-based systems using architectural modelling and simulation". In Proc. IEEE International Conference on Software Architecture, April 2017.



CONTACT US

t 1300 363 400
+61 3 9545 2176
e csiroenquiries@csiro.au
w www.data61.csiro.au

WE DO THE EXTRAORDINARY EVERY DAY

We innovate for tomorrow and help improve today – for our customers, all Australians and the world.

WE IMAGINE
WE COLLABORATE
WE INNOVATE

FOR FURTHER INFORMATION

Rob Hanson MA MSc BBus CISA CISM CRISC CRMA
Senior Research Consultant

t +61 2 6216 7028
e rob.hanson@data61.csiro.au
w www.data61.csiro.au

