Fraunhofer IML

WHITEPAPER

Future Challenges in Logistics and Supply Chain Management

A MULTI-LIGHT-NODE BLOCKCHAIN ARCHITECTURE

WHITEPAPER

A MULTI-LIGHT-NODE BLOCKCHAIN ARCHITECTURE

Any existing modern business already operates a great number of different IT systems. If businesses want to leverage the trust and other security benefits of blockchain in their business model, many, potentially all of these systems may have to be integrated. We suggest an architecture of multiple blockchains with a limited number of full nodes and many light clients that aims to keep costs low and the security level high. Our architecture shows how multiple distinct businesses can cooperate through blockchain technology, for example in logistics processes. This paper should serve as a starting point for managers, system architects and implementors who want to lay out their own architecture for their specific business requirements.

FUTURE CHALLENGES IN LOGISTICS AND SUPPLY CHAIN MANAGEMENT

Die Schriftenreihe »Future Challenges in Logistics and Supply Chain Management« greift aktuelle Herausforderungen auf, beleuchtet Trends und fokussiert neuartige Technologien und Geschäftsmodelle.

Die verschiedenen Ausgaben der Schriftenreihe zeichnen das Zukunftsbild einer innovativen Branche, das von Forschung und Praxis gestaltet und gelebt wird.

AUTOREN

Dominik Sparer, Fraunhofer IML
Max Günther, Fraunhofer IML
Christofer Heyer, Fraunhofer IML

HERAUSGEBER

Prof. Dr. Dr. h. c. Michael ten Hompel
Prof. Dr. Michael Henke
Prof. Dr.-Ing. Uwe Clausen

INTERNET

Das Whitepaper steht Ihnen auch im Internet unter www.blockchain-europe.nrw zur Verfügung.

DOI

10.24406/IML-N-614399

http://publica.fraunhofer.de/dokumente/N-589139.html

Ausgabe 16 • 30. November 2020

KONTAKT

Fraunhofer-Institut für Materialfluss und Logistik IML

Joseph-von-Fraunhofer-Str. 2–4
44227 Dortmund

schriftenreihe@iml.fraunhofer.de

WHITEPAPER

A MULTI-LIGHT-NODE BLOCKCHAIN ARCHITECTURE

A MULTI-LIGHT-NODE BLOCKCHAIN ARCHITECTURE

INTRODUCTION

Private blockchain is an emerging field of technology that allows multiple businesses to cooperate on cryptographically secured data via »democratic« processes. We discuss an architecture with security mind for businesses that want to get started with private blockchain technology. The architecture suggests using multiple, purpose-driven blockchains and proposes guidelines for an architecture of light clients and full nodes in a blockchain network that encompasses multiple businesses. Lastly, we recommend a number of architectural security measures to keep such a vast blockchain network secure.

BLOCKCHAINS

A blockchain is a distributed ledger that arrives at a consensus about transactions of assets and the general state of data via a peer-to-peer network. The established consensus can be verified as unmodified at all times through a combination of cryptography and replication in the peer-to-peer network (Xu, Weber, & Staples, 2019, pp. 3-5).

The Blockchain was first invented under the pseudonym »Satoshi Nakamoto« to serve as the backbone of the »Bitcoin« network. Its growing success brought it into the public focus and as such the technology has been greatly extended, innovated and improved upon by various organizations in the last decade (Crosby, Pattanayak, Verma, & Kalyanaraman, 2016, pp. 8-9).

Many variations and classifications of blockchain technology exist. For this paper the distinction between public and private blockchains is important. In public blockchains, everyone can participate. Whereas in private blockchains, access to the blockchain is restricted and managed (Zheng, Xie, Dai, Chen, & Wang, 2017, pp. 559-560). This makes private blockchains more suitable to corporate and inter-corporate needs. For our suggested architecture, we focus entirely on private blockchain.

BLOCKCHAINS IN LOGISTICS

While the exchange of currencies is an ancient but ever relevant concept brought to the digital age by cryptocurrencies, the transportation and exchange of goods, i.e. logistics, is even older. Just as there are many parties exchanging currencies, there are many parties in the world exchanging goods and currencies in a decentralized and seemingly chaotic manner.

Digital currencies such as Bitcoin seek to revolutionize exchange of currencies by supplanting existing digital exchange of centrally issued currencies. This transformation is not as straightforward for logistics. Any currency, and by extension any crypto currency, is essentially trust placed in a shared but otherwise ephemeral concept (McCallum, 2015). Logistics, however, involves people, goods, processes, warehouses and a cornucopia of transportation modes and vehicles. All these people, systems and – for lack of a better term – things present a challenge in that they would have to be integrated in a logistical world based on blockchain (Jakob, Schulte, Sparer, Koller, & Henke, 2018, pp. 7-8).

At first, we present basic knowledge about full nodes and light clients in a blockchain network. Afterwards, we introduce and discuss an architecture and loose organizational structure based on multiple blockchains deployed on a shared infrastructure that is connected to existing digital systems across multiple organizations.

FULL NODES AND LIGHT CLIENTS

As blockchains grow, their storage demands increase concomitantly. Because a blockchain relies on chaining of transactions and blocks via cryptographic hashing, at least parts of a blockchain network have to keep the entire ledger. The computers – often servers – that store the entire ledger are called full nodes. Other computers don't have to store the entire ledger in order to interact securely with the blockchain are called light clients, or »light nodes«.

In general, adding more full nodes to a network increases the requirements on storage and computation resources, which in turn increases the overall cost (Singhal, Dhameja, & Panda, 2018, pp. 140-141). Because of these potentially expansive storage requirements, it is desirable to have as few full nodes as possible from a cost perspective, while maintaining a sufficient number of full nodes to ensure the function, scaling and security of the blockchain. Every additional full node might perform redundant computation across the network, but this also increases resilience against attacks and improves integrity and availability (Xu, Weber, & Staples, 2019, pp. 6-7).

BLOCKCHAIN NODES

A Blockchain network generally relies on redundancy to provide access to the shared data across all participants. Full nodes download and validate the entire blockchain and provide part of the blockchain security. The entire ledger history can become exceedingly large, but most devices are not capable of storing such expansive amounts of data. For example, as of 2019 the Bitcoin and Ethereum ledgers are both respectively more than 200GB and 600GB large (Xu, Weber, & Staples, 2019, p. 56). It is foreseeable that in the future only specifically equipped server hardware will be able to provide the storage requirements. Some types of computer such as IoT devices and embedded systems don't have the capabilities to store such amounts of data. While other types of computer such as smartphones, or tablets could conceivably store the entire ledger in the beginnings of a blockchain, it is not desirable to allocate a large amount of the total storage of a phone or tablet to a blockchain, as it adds little perceived value to the individual phone or tablet in the eyes of the owner or holder. Other server systems might want to interact with a blockchain, such as ERP systems or other enterprise applications and might be able to support a full node or could be expanded for this purpose, but would also have to support the additional cost. In conclusion, it is desirable to be able to interact with the blockchain network without the burden of downloading the entire ledger.

Some full-nodes have extra responsibilities. This is heavily dependent on the framework which implements them. The following list enumerates four typical scenarios of full node responsibilities.

A full node may

- … only download the entire history.
- … download and validate the ledger.
- … download, validate and participate in the consensus mechanism.
  … take on all of the above responsibilities and also hold administrative rights, such as granting and revoking access.

Light clients are nodes in the blockchain network which only download required data without sacrificing validation. These light clients check the integrity of blocks from another full node by employing a lighter, but just as secure, hashing procedure. How this integrity check is implemented differs across blockchain implementations.

The advantage of such a setup is a decreased workload on the client systems through division of labor and a separation of concerns: Full nodes generally validate the ledger and broadcast transactions, while light client systems retrieve only required information and act as gateways for users to interact with the network in an uncomplicated manner that is computationally light weight. They receive block headers from another full node and, if required, a type of proof that certain states or blocks are undeniably part of the blockchain (Al-Bassam, Sonnino, & Buterin, 2018, p. 6).

BLOCKCHAIN TECHNOLOGIES

In practice these two concepts of light clients and full nodes are widely used and supported in multiple existing blockchain frameworks and implementations. Public blockchain technologies such as Bitcoin and Ethereum both offer light versions called »Simplified Payment Verification) (SPV) Clients. First mentioned in the Bitcoin whitepaper, Nakamoto argues that nodes only require to read the heads of blocks to check if a transaction is legitimate, or not (Nakamoto, 2008, p. 5).

In principle a light client has to download some amount of data from one or more full nodes and must thus place some trust in these source full nodes. Without further security measures a light client would be at the mercy of these source full nodes. The Ethereum developers propose solutions for this conundrum called a »fraud proof«. The light client may request fraud proof from a full node, based on the cryptographic properties of the Merkle tree, guaranteeing that a certain transaction or state is part of the chain. This is a universal approach which guarantees that a light node may operate correctly, even if all but one full node in the network acts maliciously (Al-Bassam, Sonnino, & Buterin, 2018).

Especially in public networks, problems may occur when a user decides to connect to an unknown full node operated by a third party. This security threat is mitigated in private blockchain networks where the identity is much more reliable, because it has been issued by the blockchain organization.

Not all blockchain technologies implement this verification of downloaded blocks in light clients with this fraud proof approach. The following section provides a brief overview of blockchain technologies and their approaches to verification and transaction flow in regard to the addition and download of new blocks in light clients with a final comparison at the end in Table 1: A final summary and overview of three discussed Blockchain technologies.

Hyperledger Fabric

Hyperledger Fabric is part of the Hyperledger project which is an effort to create open source blockchain applications by the Linux foundation (Dhillon, Metcalf, & Hooper, 2017, pp. 139-140). This particular blockchain is developed in cooperation with IBM and has a strong consortial character and is designed for private enterprises. As such, the technology has strong capabilities which enable a fine-grained permission management for private communication channels between parties (Nasir, Qasse, Abu Talib, & Nassif, 2018, pp. 1-2) and business oriented smart contract capabilities through so called »Chaincode« (Androulaki, et al., 2018, pp. 9-10)

A Hyperledger network is comprised of three different node types. The network is secured by »peers«, which can also have an »endorser« role that allows them to generate blocks. Transactions are ordered and delivered through the »ordering service« nodes. Lastly the »submitting-client« invokes transactions with the endorsers. This client relies on the peers for its read and write operations and may connect to any number of peers of its choice, which check and validate the client transactions (Androulaki, et al., 2018, p. 5). In practice this process executes as follows:

1. A client sends a transaction to their connected peers.

2. The endorsing peers validate and simulate the transaction according to their constitution.

3. If the process is successful, each of these peers generates and sends a certificate back to the client which signifies that they endorse the transaction.

4. As soon as enough peers signify their endorsement, depending on how many are required by the constitution, the client relays the transaction alongside the signatures to the ordering service.

5. The ordering service then broadcasts the finished transaction to all endor sing peers which verify the endorsements themselves and commit the changed state to the database.

The Hyperledger Fabric network relies on »Certificate Authorities«, which need to be configured to issue certificates to administrators and network nodes. Members of the network identify, authenticate and sign messages through a mechanism called »Membership Services« based on these identities (Hyperledger Fabric – Read

the Docs, 2020). The setup process for these mechanisms allows for finely granulated access permissions for different communication channels between parties, but as such the setup is not a trivial matter and requires careful planning. Hyperledger Fabric project currently offers SDKs for Java[1] and Node.js[2].

Tendermint and Cosmos SDK

Tendermint and Cosmos SDK are open source projects developed by »All in Bits Inc« and supported by the Swiss non-profit »Interchain Foundation (ICF)« (Tendermint - Documentation, 2020). Tendermint provides a consensus and networking layer which is realized with a »Byzantine Fault Tolerance« algorithm. A typical block addition is held in a voting round where all permissioned nodes may vote (Zheng, Xie, Dai, Chen, & Wang, 2017, pp. 560-561).

The Cosmos SDK builds on top of the Tendermint API[3] and implements blockchain related functionalities such as account, smart contract and token management via so called »modules«. Similar to Hyperledger, the network provides full nodes which may have the »validator« role. These special nodes can write and change the ledger according to the blockchain constitution. Access to the chain for »light clients« is granted through a full Node. These clients may connect to any number of full nodes (Cosmos SDK – Documentation, 2020). The transaction flow is as follows:

1. The Tendermint light client creates, signs and prepares to broadcast a tran saction to all the connected full nodes.

2. The full Nodes receive and check if the transaction adheres to the Block chain constitution. If successful it is added to the »mempool« which is used to collects transactions. A Client may choose to wait for the transaction to:

    a. Be included in a block

    b. Passed a stateful check and be added to the mempool.

    c. … or continue asynchronously after a stateless basic validation.

3. The full nodes gossip valid transactions to their peers, which then repeat step 2.

1 (Hyperledger Fabric Gateway SDK for Java - Github, 2020)
2 (Hyperledger Fabric SDK for Node.js - Github, 2020)
3 (Tendermint Cosmos SDK, 2020)

4. A random validator node proposes a block, which includes transactions from their mempool.

5. The other validator nodes vote if they agree with the content of the Block and then commit it to the chain.

6. The Light client receives notice that the block has been successfully added through gossip communication.

The Tendermint light client does not trust a single blockchain node, but rather depends on the whole set of validators. It employs several different mechanisms to secure the validity of incoming transactions, such as sequentially verifying transaction headers. A more detailed summary can be found in their documentation and publications (Kalyaev, 2020). The native light client of Cosmos SDK is written in GO.

MultiChain

MultiChain is developed as an open source project by »Coin Sciences« and receives financial backing through »Mosaic Ventures«. It originated as a fork from the Bitcoin protocol and extends the functionality to provide means to regulate access for a private or consortial deployment and removes the mining overhead. MultiChain is characterized as a simple and easy to use Blockchain framework.

A node may have a different set of permissions: The »admin« permission denotes the ability to vote on changes to the Blockchain institution and adding new admin and other high-level permissions. Another high-level permission is »mining«, which allows for nodes to participate in the consensus of adding new blocks. Communication between parties happens through so called »streams« to which the responsible node may grant or revoke access as needed. MultiChain does not offer any advanced on-chain code execution except basic filtering functionalities.

MultiChain also does not offer a native light client. However, as it is based on the underlying Bitcoin protocol it is possible to employ the aforementioned SPV mechanism to communicate with a MultiChain network in a light-weight manner. Many implementations for Bitcoin already exist and may serve as a reference point for the potential development of a light-weight client to be used with MultiChain.

Table 1: A final summary and overview of three discussed Blockchain technologies.

	TENDERMINT/COSMOS SDK	HYPERLEDGER FABRIC	MULTICHAIN
Public	Yes	No	Possible
Private/Consortial	Yes	Yes	Yes
Consensus	Byzantine Fault	Raft consensus	Round Robin
On Chain Computation	Yes: Modules	Yes: Chaincode	No: Only basic filters
Light Node/ Client	Yes: »Light Client«	Yes: »Submitting Client«	No: But could be realized via SPV.
Viability for a Multi-Light-Client Blockchain Architecture	Yes: Through usage of existing modules and developing new ones based on requirements.	Yes: Mostly out of the box with correct configuration.	Questionable: Theoretically possible but so far, no existing functionalities.

ARCHITECTURE OF A MULTI-LIGHT-NODE BLOCKCHAIN

While blockchain is often described as a distributed »ledger« technology (Bashir, 2018, p. 31), we propose a different description: private Blockchain is a union of participants with the purpose of securing a shared truth for perpetuity using a mathematically formalized constitution.

- Union of participants: Emphasizes the need for some kind of organization between the participants for private blockchains.
- Shared truth: Maintaining a shared truth between many is the core benefit of blockchain. Note that in general anything that has been appended to the blockchain cannot be removed.
- Mathematically formal constitution: Private blockchains works very much like a democratic government and similarly must balance the interest of the organization as a whole with the interests of individual participants and protect against malicious actors.

This definition focuses on the design perspective and organizational tasks at the inception of a new private blockchain in the context of a corporate environment or the cooperation of multiple organizations. We call this the »business purpose of a blockchain« and go into greater detail in the next subsection.

BUSINESS PURPOSE OF A BLOCKCHAIN

When establishing a private blockchain, a necessary precondition is to come to an agreement between the business partners that they want to share data in a decentralized and tamper proof manner. This is what we call the union of participants in the definition above. Establishing a blockchain in a single company is not necessary in most cases as no trust and transparency issues occur. To increase trustlessness between a few business partners it can make sense to include external providers that operate validator nodes, depending on business and security requirements.

A shared truth is the purpose of a blockchain designed to align the network participants and create a consensus without the necessity of a third party. It is generally better to define the data to be saved on the blockchain in the strictest terms, as the cost burden increases for every bit appended to the blockchain across all blockchain participants.

The constitution provides mathematically formal rules for validating new transactions and new blocks, but also the rules for authorization. This can be provided as a software library shared between all participants. We suggest that the constitution encompasses the following aspects

- Define the validation mechanism for transactions. In other words, define what can be appended to the blockchain.
- Define the validation mechanism for blocks as an extension to the validati on mechanism of transactions, for blocks are formed out of transactions.
- Define if smart contracts are included in the blockchain and, if included, how they operate.
- Choose to include an internal currency or not.
- Define the voting mechanism for new blocks.
- Define the minimum number of validator nodes and who operates them.
- Define the privilege system, including, but not limited to, operations like account creation and change of privileges.
- Define an update mechanism to the constitution.

There may also be some rules that can't be stated in mathematically formal terms, but can be enforced in some way by the blockchain organization through means outside of the blockchain. This can include a cost structure to share the cost burden of the blockchain more equally, for example, by instituting fees in proportion to the appended data per participant.

MULTIPLE CHAINS

The underlying blockchain network serves as the data basis and backbone that all business and smart contract decisions rely upon. This implies a high load in terms of writing and reading and requires a high level of availability. In line with blockchain and general design philosophy we suggest keeping blockchains as small and self-contained as possible to achieve a maximum of decentralization. When thinking in blockchain terms, usually a »one size fits all« solution is not feasible. In concrete terms this means:

(1) Separate blockchains for simplicity.

Blockchain applications, especially in a business environment work best if they are designed with a minimalist design paradigm in mind. Projects which desire to include blockchains as part of their architecture can quickly grow in complexity. Decisions about governance, responsibility for upholding the consensus and costs distribution become easier if the goal and purpose of the particular chain is defined as simply as possible. This eases decision making about storage, transaction and energy consumption implications.

(2) Separate blockchains for adaptability and extensibility.

In line with designing blockchains in a way that each chain has its own clear purpose, it simplifies the process of extending their functionality by combining them with different compatible chains. In the context of logistics, for example, we recommend keeping a separate warehouse chain to manage inventory storage, a separate transportation chain and one for book-keeping. Should a desire or need arise to further extend or update existing functionality of a blockchain application, then only an update for the systems involved in the blockchain is required and the other functional parts of the system can continue working undisturbed. Having dedicated blockchains allows dedicated software development teams to take responsibility for the blockchain and maintain a high level of security as business requirements change.

(3) Separate blockchains for an increased security.

Especially in terms of security it makes sense to separate blockchains not only by classification, but also by security risk. Blockchains with sensitive data hold more value than blockchains which receive and write more transient information, such as sensor data collected over long periods of time. In that sense it is easier to reason, that certain nodes may be more exposed and less secured than others.

Additionally, in terms of private networks this keeps the design decentralized, helping to contain security breaches.

(4) Separate blockchains for the deletion of data

A problem for blockchain is the fact that data is kept for perpetuity. Data which has once been recorded on the chain may not be deleted or changed, as it would break the chain which is needed to validate the transaction history (Nofer, Gomber, Hinz, & Schiereck, 2017, p. 184). This leads to constantly growing resource requirements and costs. This is especially true for data that quickly loses its value such as data obtained from sensors. Blockchain organizations can agree to cycle blockchains at regular intervals, switching to a new blockchain with the same constitution and deactivating the old blockchain after a latency period. It is possible to submit relevant data from the old blockchain to a record keeping system for continued safe-keeping.

(5) Separate blockchains, but with interoperability in mind.

Smart contracts enable interoperability between blockchains, by recording results from one blockchain onto another (Schulte, Sigwart, Frauenthaler, & Borkowski, 2019, pp. 5-6). One such scenario would be recording that the parcel has reached its destination on one chain, saving a picture of the delivery on a second and finally conducting the payment on a third blockchain. These chains should provide data fields for cross reference, similar to primary and secondary key relations in databases.

Following these guidelines creates a strong foundation upon which business processes can be modelled. Participants and operators of validator nodes may cherry-pick blockchains which are relevant to their use case, as some participants may be interested only in some data and don't want to pay for data that holds no business value to them.

SUGGESTED ARCHITECTURE

Various servers and devices are used throughout a business, all of which potentially need to communicate with a blockchain. We suggest a security-oriented architecture, which spreads full nodes and light clients across multiple businesses. This architecture is not intended as a complete solution, but rather as a starting point which can be expanded upon when starting out with private blockchain

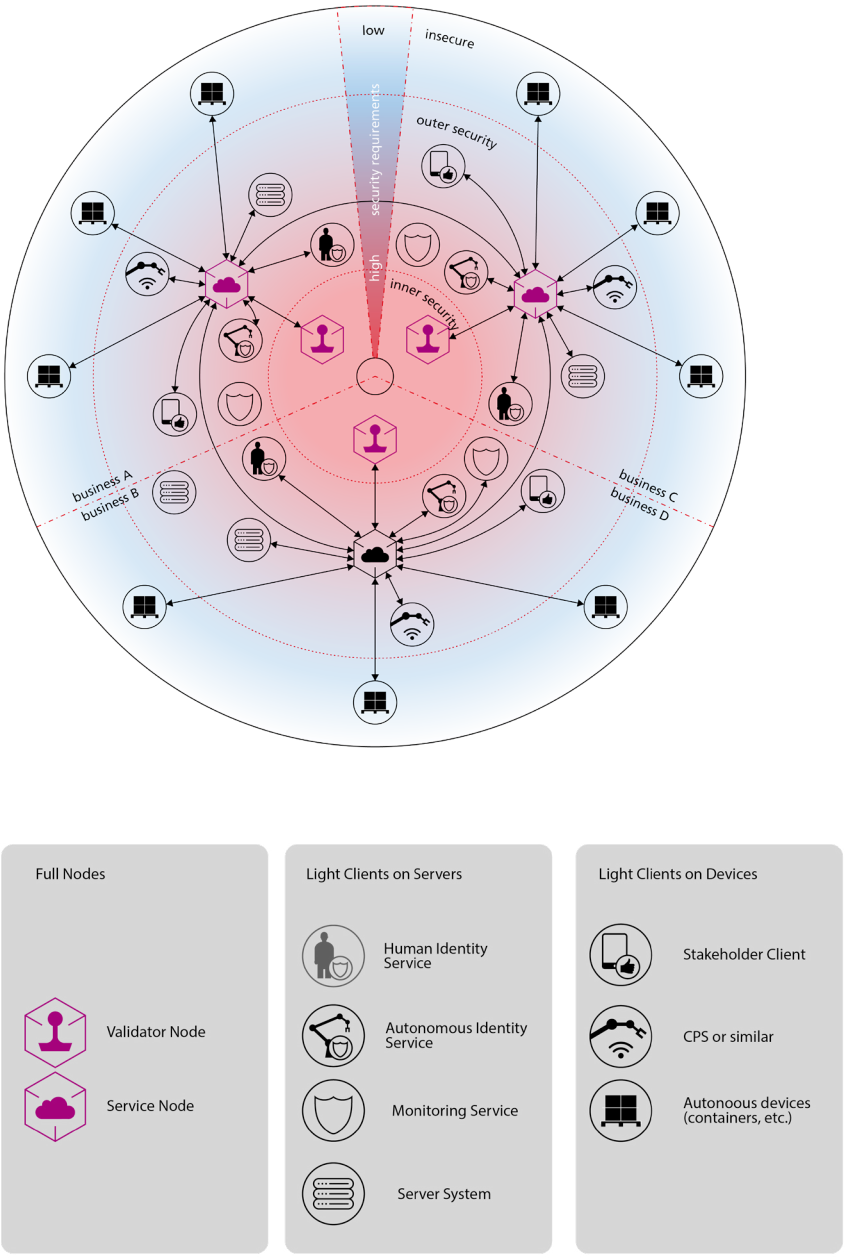


Figure 1: Suggested blockchain architecture across multiple businesses. Arrows indicate communication on the blockchain.

technology. Figure 1 shows the suggested architecture as a security »onion« diagram, which we will further elaborate on in the coming subsections. It should be noted that the diagram only shows communication on the blockchain network. If keeping secure records of that communication is not a business concern, then communication with blockchain services may happen outside of the blockchain, without compromising security.

We divide the blockchain network into business and security shells. Each business maintains their own, segregated, part of the blockchain network, enabling pre-existing and future business systems to interact with the blockchain. Every node that participates in the blockchain network, full node or light client, is placed in one of the three security shells:

- The inner shell has the strictest security and pertains to the security and operation of the blockchain itsel f.
- The outer shell extends as far as the physical boundaries of a business with controlled physical access.
- The outermost shell represents the world outside of the business premises and physical access restrictions. As such it is inherently insecure.

Note that in many cases a security shell model is not sufficient or detailed enough to adequately represent security requirements. Figure 1 thus represents security as a gradient from most secure in the center to insecure on the perimeter.

VALIDATOR NODES

Validator nodes provide the core security concept in the blockchain network. They are full nodes that validate new blocks and cast their votes in accordance with the blockchain constitution. All validator nodes together form the blockchain consensus on new blocks.

Because validator nodes provide the consensus in the blockchain, they form the most security sensitive part of the blockchain and as such should be distributed amongst the participating members of the blockchain organization. We recommend that each business operates at least one validator node to maintain their stake in this core mechanism of the blockchain. A single business may opt to operate more than one validator node, in accordance with the blockchain constitution and organization. The number of validator nodes per business is a matter that should be agreed upon by the blockchain organization, such that no business can overwhelm the blockchain consensus. We follow up on the security implications of validator nodes in the security section of the paper.

We advise that validator nodes of a business only communicate with the blockchain network through a full node of the same business which acts as a sentry. We present a compromise by merging the sentry functionality into the service node. Security can be increased by separating a dedicated sentry node from the service node. On the other hand, costs can be lowered by merging the validator functionality into the service node. However, we don't recommend this approach, because this way the private key authorized to participate in the blockchain consensus is exposed to a greater security risk.

It is also possible that a trusted and certified third party provides a part of the blockchain consensus security by operating one or more validator nodes, without participating in the business application that the blockchain is used for. This may be necessary as blockchain organizations may not have enough members to securely operate the necessary number of validator nodes to protect the blockchain consensus sufficiently. We discuss the minimum number of validator nodes in a private blockchain network in the security section in greater detail.

SERVICE NODES

Services nodes are full nodes that propagate new blocks in the blockchain across all businesses. They check incoming transactions from light clients for correctness in accordance with the blockchain constitution. As the main hub of communication in the blockchain network for each business they provide query functionality to all light clients within a business, similar to a database.

Implementors are encouraged to implement at least cursory validation checks at the service node as a first barrier to malformed or malicious transactions. As such the validator node is not the only type of node in the blockchain network that validates transactions.

It's possible to operate more than one service node per business to meet business scale, reliability and redundancy requirements. The exact purpose of service node depends greatly on what the business goals are. The representation of data for efficient access on the service node depends on the business application itself. It is therefore likely that each business may want to implement and maintain its own software version for service nodes, but it may be possible to share some source code for service nodes among the members of the blockchain organization.

MONITORING NODES

Monitoring nodes watch for aberrant behavior of full nodes and light clients across the entire blockchain network including, possibly, the blockchain activity of other businesses. As this service provides an important security feature, it is placed near the inner security shell. The blockchain organization may choose to centralize or outsource this monitoring activity to save costs or to benefit from external expertise in this area. However, centralizing the monitoring activity also centralizes an element of trust throughout the blockchain organization. This may be a future business model in a corporate blockchain world. We further elaborate and justify the monitoring activity in the security section.

AUTONOMOUS IDENTITY SERVICE

A business involved in blockchain may have many thousand if not millions of light clients. Management of blockchain identities for a single business may prove challenging in such a scenario. We suggest a service with special privileges for the automated creation of blockchain accounts for light clients that occur in great quantities, such as containers, parcels, etc. We call this the autonomous identity service, because this service handles the creation and crucially rotation of blockchain accounts for autonomous light clients such as containers and parcels in an automated fashion. The autonomous identity service utilizes a light client to communicate with the blockchain. It is located near the inner security shell as it holds a specifically privileged blockchain account for the creation and blocking of new blockchain accounts of a particular security class. There may be multiple instances of this service depending on redundancy and reliability requirements, but also to separate issuance of blockchain accounts across different security classes to enhance security in separation of concerns.

HUMAN IDENTITY SERVICE

A business may want to enable access to the blockchain for their personnel. Because blockchain keeps its own account with private/public key cryptography it is not feasible to use a corporate identity management system to sign blockchain transactions and access data on the blockchain. Instead a blockchain identity service can be used to issue blockchain accounts across multiple blockchains, once the user has confirmed their corporate identity via a traditional login. The user can deposit their public key with the human identity service after creating their own private/public key on their end-device. This legitimizes the user for an account on the blockchain using the corporate identity in an automated fashion. Otherwise adding new accounts would necessitate a vote of blockchain stakeholders or a number of trusted administrators in order to be secure. The human identity service is placed near the inner security shell as it creates new blockchain accounts and as such is especially security sensitive.

STAKEHOLDER CLIENTS

Stakeholder clients are light clients that are used by trusted business personnel to manually vote on behalf of the business in security sensitive blockchain votes in accordance with the blockchain constitution. We call these votes blockchain stakeholder votes – more on that in the security section. A business should take extra precautions to protect these devices. While these votes are particularly security sensitive, we don't place them in the inner most security ring, because these

user-held devices are inherently less secure due to the human factor involved. We suggest distributing multiple such stakeholder clients across each business for redundancy and distribution of responsibility, in case a stakeholder is unavailable or a stakeholder (device) is compromised.

SERVER SYSTEMS

Servers systems, such as ERP (enterprise resource planning) systems and other centralized IT services, are commonplace within businesses and likely require integration with the blockchain. This can be done by adding a module or addon with a light client to the system. The system then uses the light client to read data from the blockchain and submits transactions. All systems that are integrated in this fashion are likely to have their own process blockchain in a corporate IT architecture.

We placed the server systems, representing all centralized IT services, near the center of the outer security layer, as it is a security critical piece of infrastructure. As such it should already be sufficiently secured – both physically and in cyberspace – and doesn't need additional security measures for the light client. But it should be noted that the private key associated with the blockchain account of the server system is likely equipped with higher privileges, because the server system is a central data hub of a business and may want to interact with the blockchain in many ways. We recommend implementing additional security measures depending on the exact purpose, design and privileges of each server system in the blockchain.

CPS

Cyber-physical systems are a class of systems separate to the server systems class, because they also have a physical component and as such may be less secure as more people have physical access to them. As such we placed CPS on the outer perimeter of the outer security shell. Because of the greater exposure, CPS should have a much more limited set of privileges compared to server systems. Similar to the server systems class various CPS systems likely require their own process-blockchain in a corporate IT architecture.

AUTONOMOUS DEVICES

With industry phenomenon such as the Internet-of-Things (IOT) and Industry 4.0 more devices may be located outside of the premises of a business. Autonomous devices should communicate with dedicated process blockchains according to their business application to separate concerns. These devices, here summarized as autonomous devices, offer a much greater attack surface to malicious actors, because

there are no or little physical access barriers at best. For example: An IOT device on a pallet in transit is easily reachable and thus far more easily accessed and manipulated compared to a shop floor of a business or even a secured server room. For these reasons we placed autonomous devices on an additional »security shell« that represents the inherently insecure outside world. For these devices strict security measures should be taken and we present some ideas and concepts in the chapter on security.

EXTENDING THE ARCHITECTURE

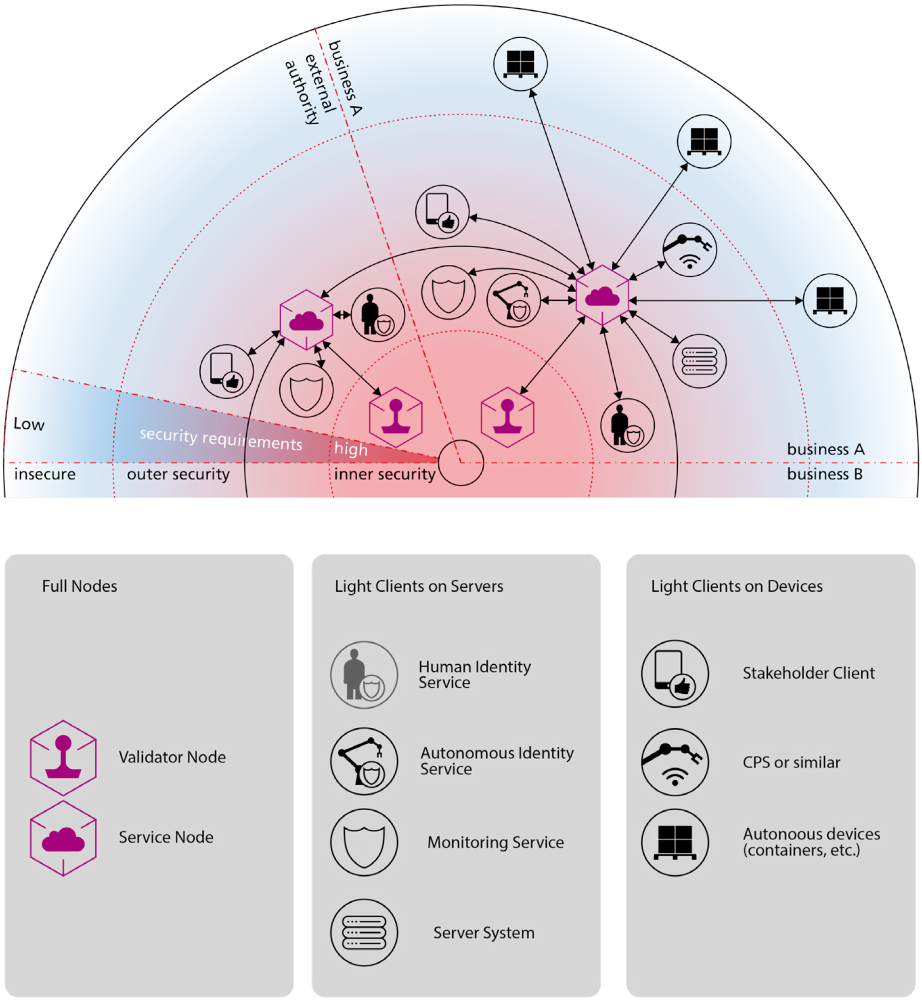
The nodes presented in Figure 1 only cover the essential and typical use cases that we envision. Any businesses implementing private blockchain technology may want and need to extend the suggested architecture. In this case we recommend that the blockchain is accessed either through a full node or a light client. While it may seem tempting to communicate with the blockchain through a dedicated (micro-) service, users should refrain from this practice for security and simplicity. Blockchain is both a database and a network technology and thus must interact with many business concerns. Any additional layers of indirection only complicate the architecture unnecessarily.

We continue with brief examples how the architecture could be modified for the inclusion of external authorities that have some stake in the blockchain and how external partners can gradually become full members of the blockchain organization.

EXTERNAL AUTHORITIES

External authorities such as government agencies and non-governmental organizations (NGOs) can use private blockchains for monitoring, compliance and transparency, as the immutability of data through shared and democratic record-keeping is a fundamental security advantage of the private blockchain technology to them. They participate by holding a stake in a private blockchain organization through the operation of a sufficient share of validator nodes as shown in Figure 2. Additional, more or less automated, monitoring systems can be implemented by these agencies and NGOs to detect malfeasance in accordance with their goals by auditing all transactions on the blockchain.

Figure 2: External authority monitors the blockchain to maintain some stake.



EXTERNAL BUSINESSES

Some business partners may eschew the IT overhead of the blockchain in the beginning, but still want to participate in the blockchain network. They can do so by having their systems interact with a service node of a fully-fledged member of the blockchain organization that permits them to do so, as shown in Figure 3. As business and scale requirements increase for the external partner, they can opt to operate their own service node in a second step. Once an external partner increases their involvement in the blockchain they may want to increase their stake in the blockchain by becoming a member of the blockchain organization themselves and operate their own validator node in a third step. More systems, such as monitoring service, can be added subsequently. Note that this order of progressive involvement is in no way idiomatic, but rather a suggestion that we deem easily feasible. Other progressions are possible, depending on the priorities of the involved parties, but we hope that our ideas provide a starting point for such external businesses in getting involved into private blockchain technology at their own pace.

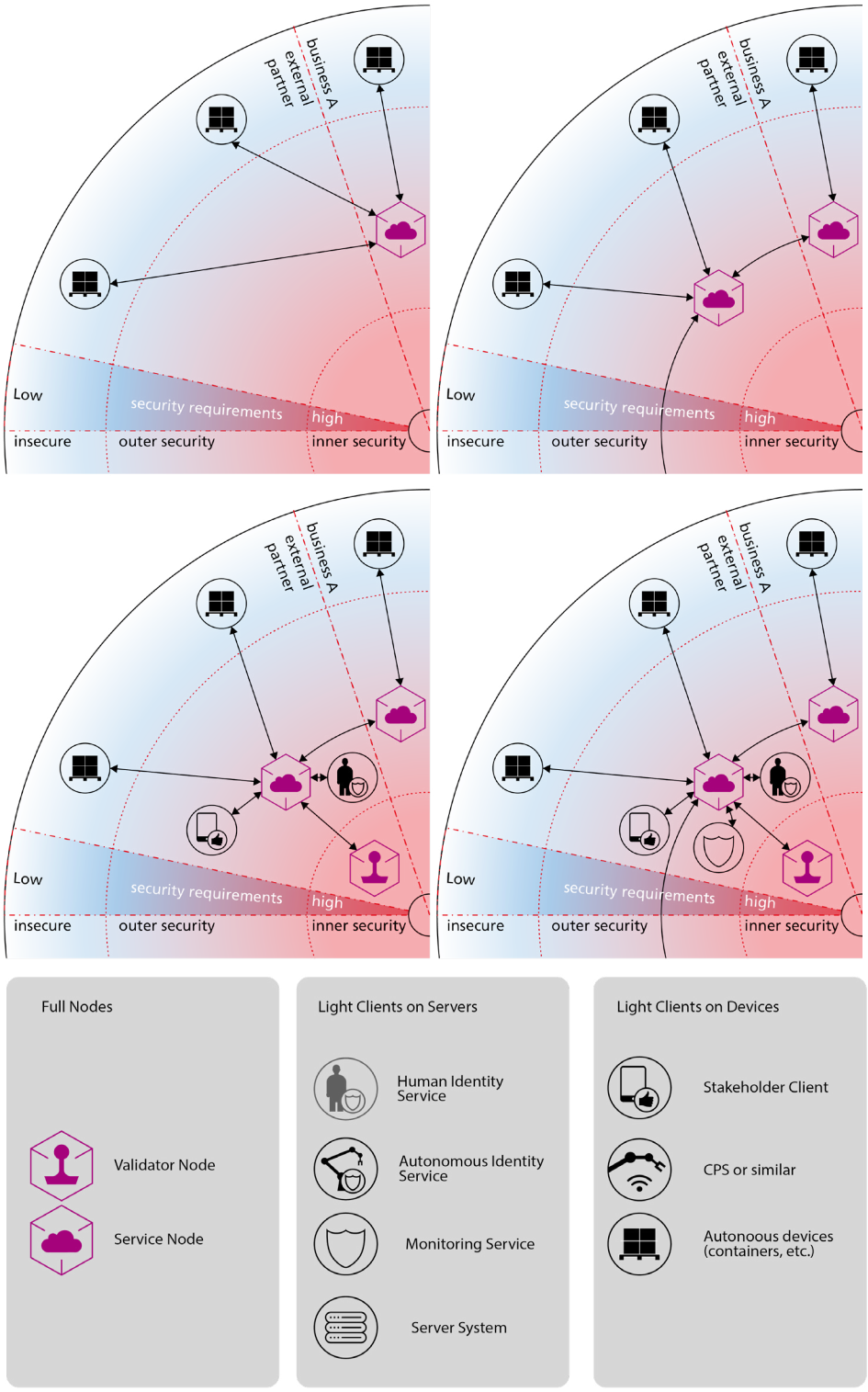


Figure 3: External partner becoming more and more involved in the blockchain.

SECURITY

While blockchain technology is often treated like a database, it is – at its core – a security technology and measures must be taken to keep it secure. The actual content of the blockchain and its internal formalisms are mathematically proven to be secure, since they rely on their underlying cryptographic mechanisms. A Blockchain, per definition, depends on its chain of transactions which are immutably linked together by a list of hashes. Any new transaction is verified by mechanisms such as symmetric key cryptography (Singhal, Dhameja, & Panda, 2018, pp. 114-124). This makes a Blockchain as a data structure immutable, forgery resistant, consistent and resilient (Singhal, Dhameja, & Panda, 2018, pp. 124-125).

However, this relies on the assumption that the managing parties take care to protect their stake. Because business want to rely on the correctness and authenticity of the data on the blockchain for smart contracts and other business processes, access and permissions must be considered carefully to prevent malicious writes or unauthorized copies of data.

We've divided security into five subsections. In network security we discuss how to divide the blockchain network into segments and isolate important nodes. We proceed to provide advice on how to design and manage blockchain accounts. After that we delve into blockchain stakeholder votes, followed by advice on monitoring light clients and software versions across the blockchain.

NETWORK SECURITY

Blockchain consists of a network of nodes, as such it is important to keep the blockchain network secure. Communication should be encrypted at the network layer to prevent man in the middle attacks and package sniffing. As of today, the Bitcoin Network traffic is unencrypted (Conti, Kumar, Lal, & Ruj, 2018, p. 3440), however most popular and private networks adopted the standard practice of securing their network communication. Ethereum as an example uses their own design called »RLPx« (Ethereum – Github, 2020) and Hyperledger Fabric implements a classic TLS protocol (Hyperledger Fabric – Read the Docs, 2020). As a general security mechanism nodes and clients in a private blockchain may only participate if they have been issued an account on the blockchain.

The suggested architecture in Figure 1 allows the business to rely on their intranet for most purposes of their blockchain infrastructure. Only two types of communication need to venture outside of the intranet: Communication of full nodes between businesses and communication with autonomous devices.

It is crucial to also separate the network along the security shell presented in Figure 1. It is most important of all to isolate the validator nodes from the rest of the blockchain network and the rest of the business intranet, as these nodes perform the most critical task of voting on new blocks. Devices outside of physical access control by the business should be isolated in a similar fashion. Additional separation can be introduced depending on the business requirements and other business specific security requirements. For example, light nodes that reside on servers in specifically secured server rooms can be viewed as their own security shell, separating them from light nodes on more easily accessible systems such as CPS[4].

The blockchain communication between businesses can be secured with B2B VPN tunnels. Alternatively, this communication can happen through the internet. In this case we recommend to whitelist the IP addresses of all other full nodes and block all other traffic. It is possible to increase security against denial-of-service attacks on the service nodes by using dedicated full nodes for the communication between business, but this should not be necessary if access is limited to other known and trusted full nodes. Of course, other means of protection against denial-of-service attacks can also be employed (Gupta, Joshi, & Misra, 2012, pp. 271-274).

Securing the blockchain communication with autonomous devices depends on the network setup used. For example, mobile networks or satellite borne internet may be used with special access protections provided by telecommunication corporation.

Pre-existing systems such as CPS and ERP systems already have their own network security, which needs to be modified to let the blockchain communication between light nodes and service nodes pass through.

4 We have opted to leave this aspect out of Figure 1 for simplicity.

BLOCKCHAIN ACCOUNTS

Private blockchain rely on carefully administered blockchain accounts for much of their security infrastructure. How this achieved depends on the blockchain constitution, but we will make some recommendations by classifying some account types. In Figure 4 we summarize these account types without going into application specific account types.

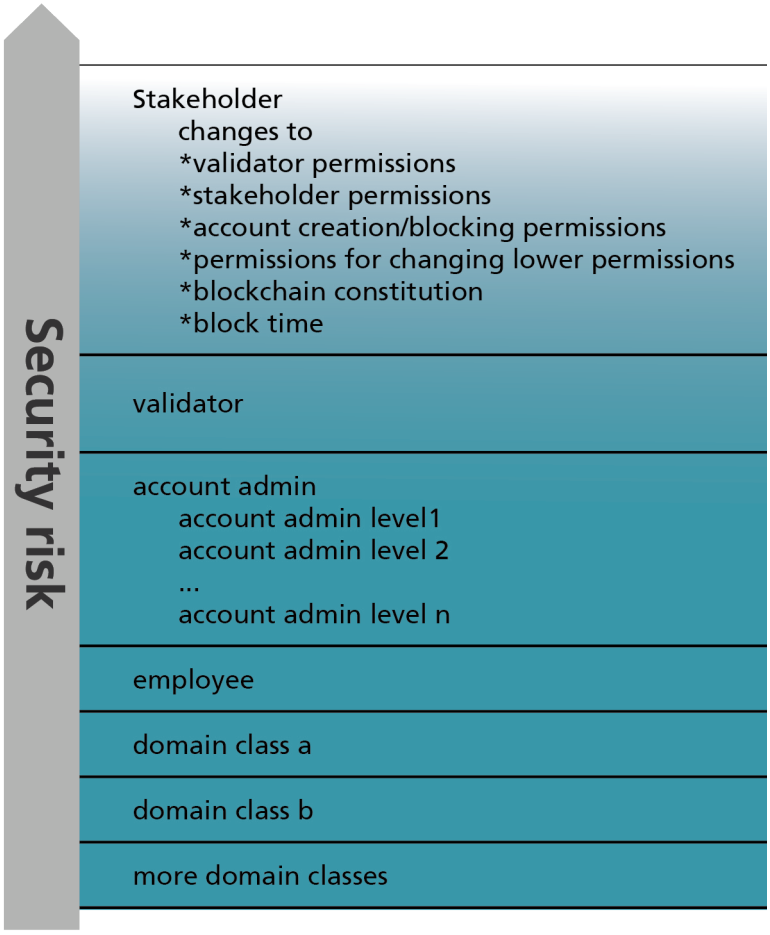
Most technologies[5] start the blockchain in a sort-of setup mode that simplifies creation and initial permission modification of accounts in the beginning. This is generally achieved by writing that info into the first block of the Blockchain which is often referred to as the »genesis block«. After this initial phase, it is important to lock permissions and account creation down.

Adding non-validator accounts must be secure, but doesn't require the full voting of all blockchain stakeholders. Human identity services can be used to automatically issue blockchain accounts based on corporate identities. Such a service would create an account and elevate its permissions based on business internal parameters, such as roles that are provided securely by a corporate identity system. Alternatively, this task can be handled by a set of human administrators. A stakeholder vote should be necessary to provide the blockchain accounts used by administrators or a human identity service with the permission for account creation and permission changes up to a certain level.

In blockchain networks where autonomous devices interact with the blockchain, adding and blocking accounts should be a frequent occurrence, because the accounts on these especially exposed devices should be rotated frequently in accordance with their security risk. It is thus necessary to have an autonomous identity service that is able to issue accounts with very few permissions to these autonomous devices. This presents an acceptable compromise between security and costs, because the automatically issued accounts should have very few permissions. Additional measures, such as regular reviews of automatically issued accounts by authorized people can be taken. Obviously, the account of the autonomous identity service must be given the necessary privileges through a stakeholder vote.

5 A few examples on how popular frameworks implement this setup process:
- Configuration Block / Genesis Block (Hyperledger Fabric - Read the Docs, 2020)
- Initialization / Genesis (Tendermint Core - Documentation, 2020)
- Getting started (Multichain - Documentation, 2020)

Figure 4: Example of blockchain account classes



Some of the most important accounts are the accounts for the validator nodes. It is important that the blockchain organization as a whole authorizes the issuance of validator permissions. This prevents expansion of voting rights in the blockchain consensus. If any party gains a majority of voting rights for the creation of new block, it can determine the truth of the blockchain, resulting in a fully compromised blockchain. A blockchain must have enough validator nodes. A minimum number of validator nodes is difficult to determine in general.

The blockchain organization should take care to distribute the validator nodes as evenly as possible to increase resilience against attackers, but must also represent the political and economic weight of the participating businesses. This balancing

6 A few consensus algorithm examples in respect to the previous three technologies as of 2020:
- Multichain implements a round-robin variant of the Bitcoin consensus (Greenspan, 2015, pp. 7-8).
- Tendermint implements a variant of a Byzantine Fault Tolerance algorithm (Buchman, Kwon, & Milosevic, 2018, pp. 1-13).
- Hyperledger Fabric 2.0 recommends using the "Raft" consensus algorithm (Ongaro & Ousterhout, 2014, pp. 1-16).

of voting power across the blockchain is vital to keep businesses in the blockchain organization without compromising the security of the consensus mechanism. The exact details also depend on the actual blockchain technology used, as voting/consensus mechanisms and required majorities differ depending on implementation[6].

The required number of validator nodes may present an obstacle for an emerging blockchain. The blockchain can be bootstrapped with lowered security requirements or outsource the operation of additional validator nodes (which is an emerging business model). For maximum security each business should operate their own set of validator nodes to protect their stake in the blockchain network.

There will be more account types for any given blockchain depending on the use case and structure of the blockchain organization. In general, all blockchain accounts should be classified in four ways:

- Risk exposure
- Value held by the account, for example tokens or currency
- Read permissions on the blockchain and the value of that data
- Write permissions on the blockchain and the potential damage that could be inflicted with malicious write operations.

These four etrics can be used to make informed design decisions when implementing a blockchain and, once in operation, can help to quickly identify the risk when a potential security issue has been identified.

BLOCKCHAIN STAKEHOLDER VOTES

To manage blockchain accounts and sensitive votes securely, blockchain organizations should distribute a carefully controlled number of stakeholder accounts that are authorized to partake in votes on the issuance of validator permissions. Stakeholder accounts are held by real people that manually cast their votes on stakeholder devices (see Figure 1) or stakeholder apps. We recommend that at least the following types of changes are to be handled by stakeholders through votes:

- Validator permissions
- Stakeholder permissions
- Account creation permissions
- Account blocking permissions

- Permissions for changing permissions up to a certain level
- Blockchain constitution, i.e. updates to the blockchain protocol and rules for validating transactions and new blocks.
- Block time (the time between the creation of new blocks)

The number of stakeholder accounts should be controlled carefully to prevent any of the business from having an undue number of votes, which could compromise the blockchain. Businesses should also provide stakeholder deputy accounts, such that the business stake in the blockchain can be kept while a stakeholder is absent, for example due to illness.

Registration of validator nodes is especially security sensitive, as an attacker can overwhelm the blockchain consensus if he or she is capable of registering more validator nodes. Registration of new validator nodes therefore should require a blockchain stakeholder vote. Adding new validator nodes to the blockchain network must be done via voting of all controlling parties to avoid losing control over the network. If hypothetically one party was allowed to add new validator on its own, then they could quickly fill up the ranks and perform something akin to a majority attack or force a fork in the chain (Conti, Kumar, Lal, & Ruj, 2018, pp. 656-657) depending on the employed consensus mechanism.

MONITORING

Blockchain networks are potentially large and have thousands of light nodes participating – especially if autonomous and other IOT devices take a part in the blockchain network. All of these systems and devices are exposed to security risk to some extent. It is therefore vital that the activity on the blockchain network is monitored. The need for such monitoring is increased, if any light node devices are in circulation in the outside world, where malicious actors can physical access to the devices.

A monitoring service would monitor activity of blockchain accounts and flag suspicious behavior. Many processes are known ahead of time and allow for validation of actual behavior with expected behavioral patterns.

It is also advisable to keep data about the involved software and hardware versions associated with the device or system. This way, if a security vulnerability becomes known, the affected devices can be easily and quickly identified.

Light clients that have been flagged for suspicious behavior, software or hardware vulnerabilities, or blockchain account renewal/rotation should be either monitored by able personnel or blocked from the blockchain, until the necessary measures have been taken to remedy the potential threat.

Autonomous devices may be compromised and send false identification or may pretend to be offline. All autonomous devices must be regularly identified and verified with an external system. This task can also be performed by a monitoring system. For example, all parcels on an incoming transport should be verified as expected with data from the blockchain. Stowaway or stray smart devices should be identified and handled with care as they represent a potential threat and ingress point for attackers.

In order to limit the impact of potential security flaws, it is necessary to rotate blockchain accounts regularly. Should a light client or the device or system that a light client resides on be compromised, the damage can be bounded in time, because the next blockchain account rotation should detect or at least lock an attacker out. These account rotations are best coupled with a complete software wipe and re-install to root out any malicious code on hacked devices. To keep costs low, it will be necessary to develop systems that automate this process.

SUMMARY AND OUTLOOK

A great number of servers, IoT devices, CPS and other business systems may want to partake in a blockchain. For any blockchain architecture which aims to provide business value, it is necessary to make copious use of light clients, to reduce deployment and operational costs. This avoids large storage and computational requirements which would otherwise be imposed upon businesses if full nodes were to be used everywhere.

In this paper we presented a starting point for a blockchain architecture encompassing multiple businesses and multiple blockchains. We began with important blockchain infrastructure across businesses with service (full) node validator nodes, stakeholder clients, monitoring and account management. We also included the integration of ERP and CPS systems as an example of preexisting business systems in our blockchain architecture. This shows how full nodes and light clients work together in a business blockchain.

As part of our discussion of important security considerations, we separated businesses with their own infrastructure and security requirements as part of the whole blockchain architecture. We also categorized different types of systems in regard to their general security into multiple security shells to provide a guideline to architects, businesses and implementors.

This paper should serve as an overview and starting point for future blockchain adopters. Further empiric research can build on this paper in order to gain more experience with the emerging field of business specific private blockchains.

Some generic software artifacts can be developed that should serve any implementor of the suggested architecture, such as monitoring software, human and autonomous identity services and stakeholder clients. In the research project »Blockchain Europe« we will be developing some of the software artifacts and aim to further validate our architecture.

We hope that researchers and businesses that apply our architectural ideas in the real-world publish their research to further the discussion and refinement of private blockchain architecture spanning multiple partners with full nodes and light clients.

BIBLIOGRAPHY

(2020, October 14). Retrieved from Hyperledger Fabric – Read the Docs: https://hyperledger-fabric.readthedocs.io/en/latest/glossary.html

(2020, October 14). Retrieved from Hyperledger Fabric Gateway SDK for Java – Github: https://hyperledger.github.io/fabric-gateway-java/

(2020, October 14). Retrieved from Hyperledger Fabric SDK for Node.js – Github: https://hyperledger.github.io/fabric-sdk-node/

(2020, October 14). Retrieved from Tendermint Cosmos SDK: https://tendermint.com/sdk/

(2020, October 14). Retrieved from Cosmos SDK – Documentation: https://docs.cosmos.network/master/basics/tx-lifecycle.html

(2020, October 14). Retrieved from Ethereum – Github: https://github.com/ethereum/devp2p/blob/master/rlpx.md

(2020, October 14). Retrieved from Hyperledger Fabric – Read the Docs: https://hyperledger-fabric.readthedocs.io/en/release-2.2/membership/membership.html

(2020, October 14). Retrieved from Hyperledger Fabric – Read the Docs: https://hyperledger-fabric.readthedocs.io/en/release-2.0/enable_tls.html

(2020, October 14). Retrieved from Multichain – Documentation: https://www.multichain.com/getting-started/

(2020, October 14). Retrieved from Tendermint Core – Documentation: https://docs.tendermint.com/master/tendermint-core/using-tendermint.html

(2020, October 14). Retrieved from Tendermint – Documentation: https://docs.tendermint.com/

Al-Bassam, M., Sonnino, A., & Buterin, V. (2018). Fraud proofs: Maximising light client security and scaling blockchains with dishonest majorities. arXiv.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., & Muralidharan, S. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. Proceedings of the thirteenth EuroSys conference (pp. 1-15). Proceedings of the thirteenth EuroSys conference.

Bashir, I. (2018). »Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained.«. Packt Publishing Ltd.

Buchman, E., Kwon, J., & Milosevic, Z. (2018). »The latest gossip on BFT consensus.«. arXiv.

Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of bitcoin. In Communications Surveys & Tutorials, 20(4), (pp. 3416-3452). IEEE.

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). »Blockchain technology: Beyond bitcoin.« . Applied Innovation, pp. 1-20.

Dhillon, V., Metcalf, D., & Hooper, M. (2017). »The hyperledger project.« Blockchain enabled applications. Berkeley: Apress.

Greenspan, G. (2015). »Multichain private blockchain-white paper.«. Retrieved from http://www. multichain.com/download/MultiChain-White-Paper.pdf

Gupta, B. B., Joshi, R. C., & Misra, M. (2012). »Distributed denial of service prevention techniques.«. arXiv.

Jakob, S., Schulte, A., Sparer, D., Koller, R., & Henke, M. (2018). Blockchain and Smart Contracts. https://www.iml.fraunhofer.de/content/dam/iml/de/documents/OE260/10_Whitepaper_BlockchainSmart-Contracts_Ausgabe_10_WEB.pdf: Fraunhofer IML.

Kalyaev, A. (2020, 06 25). Medium. Retrieved 09 30, 2020, from https://medium.com/tendermint/everything-you-need-to-know-about-the-tendermint-light-client-f80d03856f98

McCallum, B. (2015). The bitcoin revolution. Cato J., p. 347.

Nakamoto, S. (2008, 10 31). Satoshi Nakamoto Institute. Retrieved 08 31, 2020, from https://nakamotoinstitute.org/bitcoin/

Nasir, Q., Qasse, I. A., Abu Talib, M., & Nassif, A. B. (2018). »Performance analysis of hyperledger fabric platforms.«. Hindawi.

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. Business & Information Systems Engineering 59(3), pp. 183-187.

Ongaro, D., & Ousterhout, J. (2014). »In search of an understandable consensus algorithm.«. Annual Technical Conference. 305-319: USENIX.

Schulte, S., Sigwart, M., Frauenthaler, P., & Borkowski, M. (2019). »Towards blockchain interoperability.«. In In International Conference on Business Process Management (pp. 3-10). Springer, Cham.

Singhal, B., Dhameja, G., & Panda, P. S. (2018). »Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions.«. Apress.

Xu, X., Weber, I., & Staples, M. (2019). Architecture for blockchain applications. Springer.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017, 6). An overview of blockchain technology: Architecture, consensus, and future trends. IEEE international congress on big data (BigData congress), pp. 557-564.

blockchain
europe.nrw

#DWNRW Digitale Wirtschaft

GEFÖRDERT VOM

Ministerium für Wirtschaft, Innovation,
Digitalisierung und Energie
des Landes Nordrhein-Westfalen

INNOVATIONSLABOR
Hybride Dienstleistungen
in der Logistik

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung