A Deep Analysis of Blockchain Technology from Socio-Economic and Technological Perspectives: Intrinsic Barriers and Potential Impact

Ahto Buldas[1], Dirk Draheim[2]✉, Takehiko Nagumo[3,4], and Anton Vedeshin[5]

[1] Centre for Digital Forensics and Cyber Security
Tallinn University of Technology, Estonia
ahto.buldas@taltech.ee
[2] Information Systems Group, Tallinn University of Technology, Estonia
draheim@acm.org
[3] Graduate School of Management, Kyoto University, Japan
takehiko.nagumo.4s@kyoto-u.ac.jp
[4] Mitsubishi UFJ Research and Consulting, Tokio, Japan
takehiko.nagumo@murc.jp
[5] 3D Control Systems Ltd., San Francisco, USA
anton@3dprinteros.com

Abstract. *Purpose:* Our aim is to understand technological and socio-economic barriers to blockchain solutions that are intrinsic in the blockchain technology stack itself (permissionless as well as permissioned). On the basis of that, we want to understand the future potentioal impact of blockchain technology.
Design/methodology/approach: We provide an argumentation against the theoretical background of Williamson's instutional analysis framework, and triangulate the insights with results from four design science research efforts.
Findings: We (i) characterize cryptocurrency as one-tiered collateralized money. We (ii) review potential blockchain solutions against defined essential modes of communications. We review (iii) well-known scalability issues and potential denial-of-service attacks through a new probabilistic model. We (iv) characterize a typical neglection of physical network infrastructure in blockchain technology discussions. We (v) describe four successful blockchain solutions and explain their design. There is (vi) no evidence that the proclaimed "blockchain revolution" can disrupt our institutional stack; instead, it can only happen in the boundaries of the current institutional stack. Nevertheless, it is possible to (vii) design useful blockchain solutions.
Practical implications: The findings of this research enable policy makers, decision makers and information systems architects alike to make informed decisions about blockchain technology and its application.
Originality: Given its theoretic foundation in new institutional economics, triangulated with comprehensive results from design science efforts, this study is the first of its kind in the area of blockchain technology research.

Keywords: Blockchain · distributed ledger · monetary system · transaction cost economics · new institutional economics

1 Introduction

Triggered by the quick, unexampled spread of the cryptocurrency Bitcoin [1] since its introduction in 2009 and its immense resonance in media, we have seen, in the last decade, a phlethora of envisioned blockchain solutions. Often, such envisioned blockchain solutions claim to be *disruptive*. Such blockchain disruptiveness is often fundamental, i.e., it takes the form of (or is embedded into an overall) proclaimed *blockchain revolution*, i.e., goes beyond the total transformation (and take-over) of an existing business model or even business

(sub-)domain, as is the target of the usual start-up (with Booking, Uber, Airbnb as prominent examples). The proclaimed blockchain revolution targets the shake-up of whole vertical and horizontal markets, today's monetary system [2], the institutional design of the state including the judiciary system [3], i.e., whole societies and political systems. But how realistic is such a blockchain revolution? Is it the blockchain revolution that enacts a shake-up of the institutions around us? Or does the blockchain revolution need to rely on a hypothetical re-design of the institutional design instead? How critical is the blockchain revolution when it comes to the design of concrete blockchain technologies, let them be permissionless or permissioned? It is such questions which have motivated us to compile the recent paper and we hope that we can bring some light to them, at least to create some awareness for them.

In service of these questions, we contribute and extend the following, mutually dependent discussions, compare also with [4]:

- We characterize cryptocurrency as one-tiered uncollateralized M1 money.
- We define essential modes of business communications (message authentication, signature, registered letter, contract, order etc.) and how they are digitized classically. We review potential blockchain solutions for these modes of communications, including socio-economic considerations.
- At the technical level, we discuss (i) scalability issues and potential denial-of-service attacks and (ii) a characteristic neglection of physical network infrastructure.
- We look into four successful blockchain solutions (that we have designed over the last years) and explain their design:
  - Guardtime: A tamper-proof timestamping service.
  - 3DPrinterOS: An automated manufacturing software ecosystem.
  - Agrello: an electronic identity and digital signature solution.
  - Thinnect: a sensors-as-a-service marketplace.

We proceed as follows. In Sect. 2, we summarize essential preliminaries and some theoretical background[6]. We discuss related work throughout the paper and review some selected, important literature in Sect. 3. In Sect. 4, we characterize Bitcoin – and cryptocurrency in general – as an uncollateralized, one-tier M1 money. In Sect. 5, we explain modes of business communications and their possible blockchain realization. In Sect. 6 we discuss intrinsic technological barriers of permissionless blockchain technology. In Sect. 7, we discuss a series of successfull blockchain-based solutions. Section 8 aims at compiling a high-level decision maker discussion. It provides a scheme of requirements for the potential use of distributed ledger technology[6]. We finish the paper with a conclusion in Sect. 9.

2 Preliminaries and Theoretical Background

We start with a discussion of the Bitcoin vision in Sect. 2.1 as it laid the foundation for blockchain technology, proceed with an explanation of blockchains and distributed ledgers in Sect. 2.2 and 2.3. Then, we provide a brief explanation of Williamson's new institutional economics that we need in our analysis in Sect. 2.4.

2.1 The Bitcoin Vision

The Bitcoin vision was published in 2008 as a white paper by the anonymous author (or author collective) Satoshi Nakamoto [1]. The aim was to create a peer-to-peer payment system that works free from third party intervention in which payments were instant and irreversible. "Ordinary" bank money was not considered suitable for free electronic commerce as banks may (in principle) delay and revert payments and – for security reasons – require too much sensitive personal information from their clients. The Bitcoin system can be described as a fully automated financial transaction provider – the rules of which are public and cannot be

[6] Sections 2.2, 2.3 and 8 are based on a technical report (cryptographic algorithm lifecycle study) by AS Cybernetica [5] that has been ordered by the Information Systems Authority RIA (Riigi Infosüsteemi Amet) of the Republic of Estonia in 2018.

broken due to the principal design of the system. Payment orders are automatically processed in accordance with these rules and are added to an append-only public ledger that can be verified by everybody.

The syntax and semantics of the ledger (as a data structure) are supposed to guarantee its verifiable uniqueness – a successful verification of the ledger must be unique, i.e., so that alternative versions of the ledger cannot exist. In Bitcoin, this is supposed to be guaranteed via the *proof-of-work* consensus protocol: the idea is, that creating a correctly verified ledger would take so much energy that the existence of alternative/parallel versions of the ledger is beyond reasonable doubt. As reverting a payment means rewriting the ledger, it would not possible for any individual or any organisation to delay or revert payments, i.e., the Bitcoin vision needs to consider potential *double spend* attacks and DoS (denial-of-service) attacks as purely theoretical threats. Note that verifiable uniqueness of the ledger means that it is not important at all how and by whom the ledger was created. Only the data of the ledger itself is important for verification.

Automation assumes data processing, which certainly needs a machine. Now, it is possible to identify Bitcoin with the machine that is needed to run its data processing, and, henceforth, we can therefore talk about the *Bitcoin machine*. In the Bitcoin vision, such a Bitcoin machine has to be flawless and unstoppable. Destroying the machine would stop the payments. This necessarily leads us to the following considerations:

(i) *What is the Bitcoin machine and which organisation runs it?* By assumption, such a machine cannot be under control of any organisation as such an organisation would be able to stop the machine. Therefore, the Bitcoin machine is a network of servers that are run on voluntary basis. All network nodes (servers) maintain the exact same copy of the Bitcoin ledger.

(ii) *But what guarantees that there are sufficiently many volunteers that run a server?* For that, Bitcoin has an incentive mechanism. Each volunteer who wins a round in the infinitely repeating proof-of-work consensus race is paid with a certain amount of Bitcoin currency (a fixed *(coinbase) reward*, plus variable *transaction fees*). Volunteers are economically motivated as far as the Bitcoin "currency" presents some *economical value* to them.

(iii) *But what is the economical value of the Bitcoin "currency"?* Bitcoin has an economic value *if* it is accepted as payment in exchange of goods or services (in marketplaces) or currency (in financial markets).

Given (i) above, we have that Bitcoin is a technical machine, given (i-ii), we have that Bitcoin can be characterized as a *"social machine"*. It is the latter that explains the emergence of a whole new research area called *cryptoeconomics*.[7].

2.2 Blockchains and Distributed Ledgers

Some would identify blockchain with Bitcoin, i.e., with the Bitcoin blockchain. Others would at least insist that a blockchain is run with a proof-of-work consensus such as introduced in the Bitcoin vision. Some prefer not to speak about blockchains at all and prefer to talk about *distributed ledgers* instead. Again others would insist on a subtle difference between blockchains and distributed ledgers, i.e., saying that a blockchain is a data structure, whereas a distributed ledger is the electronic document realized by a blockchain. As long as they are no standardized definitions, we are rather agnostic towards terminology. For example, for the sake of this paper, we talk about blockchains often also merely as data structures, i.e., independent of the concrete kind of consensus mechanism used by the concrete blockchain technology.

A blockchain is a data structure composed of successive data blocks. Each new block is either created after a fixed time period or as a result of some other event, such as successful mining. The content of the blocks as a data structure represents a specifically encoded ledger, which records events that may have informal, commercial or legal significance. The blockchain format is known to its operators and users. The integrity of a blockchain is ensured by iterative hashing. A hash is calculated for each block by applying a hash function to the block's data and the hash of the previous block. The integrity of the hashes is ensured through the use of:

– *digital signatures*, i.e., digital signatures of persons authorised in some way (with public keys),

[7] https://ce.mit.edu/

– *hashed time-stamping* that uses a publication mechanism, or
– *non-interactive time-stamping* in the form of special formatting rules for blocks that make the creation of valid blocks a computationally complicated task (proof-of-work).

The blockchain is usually stored and managed in the form of a distributed ledger, with multiple parties keeping a copy of the ledger, which then implies the use of a handshake protocol between the components. However, also centralized blockchain system exist. The original and surely the most popular application of blockchains is *cryptocurrency*, where the ledger displays user account balances and inter-user payments in a "currency" defined by the ledger itself and not necessarily in one of the traditional currencies. Nevertheless, cryptocurrency may be traded on the stock exchange and exchanged for traditional money, which makes it hard to distinguish between traditional currency and cryptocurrency and as official vs. non-official currency. The most widely recognised cryptocurrency system is Bitcoin, which establishes and uses Bitcoins and Satoshis as currency.

Blockchain systems can be

– *Centralized*, where the ledger is managed as a centralised service by one legal entity. For example the Guardtime[8] system.
– *Permissioned ledgers*, in which the provision of services is distributed between a number of fixed actors acting on a contractual or other legal basis. For example, Ripple[9], which enables interbank transactions, or Sovrin[10], which is managed by financial institutions and is seeking to build a global decentralised identity system.
– *Permissionless ledgers*, where service providers are not fixed and, in principle, anyone can start operating the service. For example, Bitcoin[11,12] and the early versions of Ethereum[13].

Most permissionless blockchain systems include an independent cryptocurrency. The reason for that is that in the absence of an inter-operator contract, there are usually no other incentives to guarantee voluntary management of the blockchain.

A *ledger* is a constantly updated electronic document that records events of informal, commercial or legal significance. The ledger format depends on its field of use. A ledger has an organisational and a technical structure. Organisationally, the ledger includes:

– *Users*: Persons or entities who/which add entries to the ledger as appropriate. Users are usually identified in the ledger by their public key, which is mostly associated with traditional digital signatures, such as RSA, DSA, ECDSA, etc. The public key or its hash is generally treated as an account number.
– *Operators*: Persons or entities who ensure the validity, maintenance, and operation of the ledger. Operators only accept and add items to the ledger meeting the format. Operators may have to personally add data to the ledger if it is required by the format.

Technically, operators are viewed as automatically acting entities. In legal terms, operators can be either legal or natural persons, and their participation in the management of a ledger may not be subject to regulation by any law. For example, Bitcoin system operators operate on a voluntary basis and in most cases anonymously.

It is important to note that although the ledger maintenance activities of the operators may not be subject to regulation by any law, the contents of the ledger may have legal significance – if the users have so defined. For example, if a ledger manages inter-user payments and account balances, the records may include items such as payer, payee, paid amount, etc. The ledger format may require that the transaction item must

[8] https://guardtime.com/
[9] https://ripple.com/
[10] https://sovrin.org/
[11] https://github.com/bitcoin/
[12] https://www.bitcoin.com/
[13] https://ethereum.org/

be accompanied by a digital signature of the payer (for example with a traditional signing algorithm such as RSA, CDSA, etc.), the paid amount may not exceed the amount available on the payer's account and so forth. Now, if account balances are defined as entries and a payment with valid formatting is added to the ledger, the operator must also modify the entries representing the respective balances.

Formally, the format of a ledger can be represented as a set of verifiable conditions. The verification of such conditions can be a complex task, especially, if users can dynamically extend them, as is possible, e.g., with smart contracts that we discuss in due course in Sect. 2.3.

If there are multiple ledger operators, then the ledger is called a distributed ledger. In such a case, entries sent by users are received, directly or indirectly, by all operators managing the ledger independently. Distribution may be required for two main reasons:

- *Trust.* A solution with a single operator is not deemed sufficiently reliable considering the potential for corruption.
- *Reliability (Availability, Resilience).* A solution with a single operator is not deemed sufficiently reliable considering the possibility that an operator may become unavailable due to network issues or other technical issues.

Distributed ledgers need some form of consensus mechanism. Albeit trust and reliability are the two driving forces to step from a centralized to a decentralized solution, trust and reliability remain an issue in centralized solution. Actually, establishing trust is the essential (intrinsic) issue of permissionless distributed ledgers, but it might also be an issue in permissioned blockchains. Network problems may cause a situation where all operators do not receive all entries, and thus received entries may vary across operators. Again, to ensure consistency across all copies of the ledger, a consensus protocol between operators is required. Strategies for reaching consensus can be divided into several classes:

- *Agreement protocols*, in which the next block is agreed upon between operators only if all preceding blocks are agreed upon and consistent across the board.
- *Nakamoto consensus (proof-of-work)*, which uses a uniform chain comparison criterion; if there are two chains, then one of them is "stronger" and the operators always prefer the strongest chain (more precisely: a decision is made for the strongest chain in terms of accumulated difficulties (longest-difficulty valid chain [6]); if an alternative chain of equal length (and same strength) is formed and distributed later, it does not replace the main chain.) [1, 6].
- *Proof-of-stake*, in which operators who have more cryptocurrency gain priority in block creation.

Operators participating in an agreement protocol may have different versions of a block. The agreement protocol must be designed in such a way that a sufficiently large number of properly functioning operators involved in that protocol reach a consensus on the same version of that block. Agreement protocols are usually time and message-intensive. If the number of operators is very high, the use of an agreement protocol by all operators might become impractical or even impossible. Therefore, some blockchain technologies (e.g., Algorand[14]) use lottery-based scaling coalition. The main drawback of blockchain technologies based on agreement protocols is that the agreement protocol must necessarily succeed before the creation of new blocks. If, for some reason, it gets stuck, then it disrupts the entire ledger management.

The main purpose of the Nakamoto consensus is to create stabilize on a joint blockchain as fast as possible, in a situation where, the chains of properly functioning operators have longest possible identical initial segments of the chain. An operator who managed to create the next block, sends it to other operators, who try to attach it to the 'previous' known blocks to chain them up. The correct chain will be chosen based on the chain comparison criteria. It is important to understand that although a new block B_{t+1} refers unambiguously (based on hash values) to some earlier block B_t, there may not be an existing consensus about the previous block B_t, and it is therefore possible that later on, a chain $B_0, \ldots, B_{t-1}, B'_t, B'_{t+1}$, where the blocks B'_t and B'_{t+1} might not match blocks B_t and B_{t+1}, is deemed the valid chain. It may happen that block B_t contains some entry r, but later, the 'winning' blocks B'_t and B'_{t+1} do not contain the entry r.

[14] https://www.algorand.com/

Therefore, a situation may arise where some recently added entry r is *removed* from the ledger.Keeping that in mind, operators typically keep a separate account for incoming entries. If any entry r is removed from the ledger, the operator will re-enter it into the ledger as soon as possible. Malicious miners can exploit the potential removal of a blockchain entry for a so-called *double spend attack*.

As the Nakamoto consensus is based solely on chain comparison and does not assume that the entire preceding chain is synchronised, protocols based on Nakamoto consensus are much more reliable during poor network conditions than agreement protocols. At the same time, the possibility to remove freshly added items and to wait for the ledger to "stabilise" should be considered.

Finding a block with a specified strength is called mining. In the Bitcoin system, mining is about shuffling a 64-bit parameter in each block so that the hash value of that parameter would smaller as a set target value (the lower the target value, the higher the difficulty). Proof-of-work aims to protect the integrity of the ledger by making it difficult to modify "old" entries, as changing block would also require changing all subsequent (later) blocks.

Proof-of-stake is an alternative to proof-of-work that helps increase operators' economic incentives. Proof of stake requires that operators have a sufficient amount of personal cryptocurrency in the system and therefore have enough economic incentive to maintain the system. The rules for creating a block and determining its difficulty are created in such a way that operators who have more cryptocurrency gain priority in block creation. For example, the right to create a block may be decided by lottery, where the likelihood of winning is proportional to the amount of cryptocurrency possessed.

2.3 Smart Contracts

A smart contract [7, 8] is an entry in the ledger that represents a contract between users. A smart contract consists of a predicate term and programmed instructions for modifying the content of the ledger. Smart contracts may contain complex calculations. The blockchain system of Ethereum[13], for example, allows rules to be written in the full-fledged third-generation programming language Solidity. As incentive mechanism, a smart contract format may allow for charging a fee based on the number of instructions that the contract signers have to pay to the operator of the smart contract system.

As an instructive example, two users could bet if the hash of a block created after a certain time period is an odd or an even number: if the hash comes up as even, then one user pays a certain amount of cryptocurrency units to the other user and vice versa. To achieve such behavior of the ledger, a contract description is stored in the ledger and both parties attach their digital signatures (which are also stored in the ledger) to the contract. The task of the operator is to monitor the smart contract and to update the balances of the contract partners when agreed-upon time has elapsed and the hash of the next block has been calculated. Here, it is assumed that the ledger blocks contain the time of their creation as entries. If it is required by the smart contract technology that the predicate term may depend only on the (bit) content of the ledger (i.e., must not depend on any external parameters), it would be even necessary, that the blocks contain their creation times.

As another example, if we would like to create a sports results tote-board based on blockchain technology, the program that implements the predicate term could access a trusted external sports result database. As another solution, the ledger could also store the sport results itself. Of course, the validity predicate cannot include result verification logic, however, it may require that the stored results are signed and verifiable using a fixed set of trusted public keys. Still, also in the latter solution, the values need to enter the ledger and, therefore, also in the latter solution, the predicate term in dependent on external values, albeit indirectly.

We might describe smart contracts that are dependant on external values as *embedded* smart contracts ("contracts embedded in the world" [8]). Embedded smart contracts are smart contracts *per se*, as they show in envisioned application scenarios as found in [8] or many white papers of respective ICOs (initial coin offerings).

Level	Purpose	Frequency
L1 (social theory) *Embeddedness*: informal institutions, customs, traditions, norms, religion	Often noncalculative; spontaneous.	100–1000 years
L2 (economics of property rights) *Institutional Environment*: formal rules of the game – esp. property (polity, judiciary, bureaucracy)	Get the institutional environment right. 1st-order economizing.	10–100 years
L3 (transaction cost economics) *Governance*: play of the game – esp. contract (aligning governance structures with transactions)	Get the governance structure right. 2nd-order economizing.	1–10 years
L4 (Neo-classical economics / agency theory): resource allocation and employment (prices and quantities, incentive alignment)	Get the marginal conditions right. 3rd-order economizing.	continuous

Table 1. Economics of institutions; compiled from Williamson 1998 [9].

2.4 New Institutional Economics

We have chosen Williamson's *new institutional economics* [9] as a theoretical reference framework to strengthen (to provide anchors for) many of our arguments throughout the paper. In Table 1, we have compiled the "four levels of social analysis" of new institutional economics (from Fig. 1 in [9]), L1 trough L4, which continuously evolve, at different pace and with different volatility; where they all influence each other (back and forth, even across several levels) in this evolvement. Level L1 is about culture at the societal level; level L2 is about laws, regulations and government; level L3 is about organizational governance in so far it concerns inter-organizational transactions; whereas level L4 is the most fine-grained level of individual actors. It makes sense to study large, complex socio-technical systems with the help of such an institutional analysis framework, compare with Koppenjan and Groenewegen [10]. We are interested in two kinds of questions, i.e., in how far emerging technologies impact institutional design at the different levels of analysis, on the one hand, and (vice versa) in how far changes to institutional design are prerequisites (or at least enablers) for the successful introduction of emerging technologies. Here, not only the amount but, in particular, the level of impact makes a difference in the degree of *disruptiveness* of a technology.

3 Related Work

The multi-author volume [11] (edited by Malcolm Campbell-Verduyn) provides an interdisciplinary assessment of Bitcoin and cryptocurrency from several perspectives of social sciences as well as technology studies. The discussion ranges from new forms of organizations over the cashless society to the impact of global governance. The potential impact of blockchain technology (as an emerging technology) is analyzed in terms of normative implications, dis-/empowerment, decentralization, ethics, legitimacy etc.

In [12, 13], Arvind Narayanan and Jeremy Clark review a series of seminal papers and contributions in cryptography that are related to Bitcoin technology[11] (as described in [1]): linked timestamping [14] (including the Guardtime solution described in Sect. 7.1), merkle trees, byzanthine fault tolerance, proof-of-work, Hashcash. This way, the paper also increases the understanding of the building blocks of blockchain technology.

With [15], Jan Mendling et al. provide a mature analysis of the potential of blockchain technology for business process management (BPM) [16] (and the challenges thereof). Due to its de-centralized nature, the significant potential of blockchain technology for enabling cross-organizational business process is explained. First, the authors eleborate a thorough understanding of the utiliziation of blockchain technology in terms of the established BPM lifecycle. Next, they reflect on the potential of blockchain technology beyond the boundaries set by the established BPM lifecycle, i.e., in terms of the BPM capability areas of Rosemann and vom Brocke [17]: strategy, governance, information technology, people, and culture. The authors conclude with identifying and characterizing seven future research directions relevant to the adoption of blockchain technology for (and its impact on) business process management, i.e., (i) executing/monitoring

blockchain-based systems, (ii) analysing/engineering blockchain-based processes, (iii) redesigning processes for blockchain readiness, (iv) evolution/adaptation of blockchain technology, (v) identification/discovery of processes that are amenable for blockchain technology, (vi) impact of blockchain technology on strategy and governance, (vii) cultural shifts needed for (enacted by) the adoption of blockchain technology.

In [18], Rikken, Janssen and Kwee have compiled a catalogue of governance challenges of blockchain technology, in particular decentralized autonomous organizations. The findings are based on a combination of structured literature review (51 papers) and interviews (two industry experts; two researchers). The challenges has been categorized with respect to several layers (infrastructure, application, company, institution/country) and stages (design, operate, evolve/crisis) to have them prepared for further analysis. Based on that, the key findings are as follows. As opposed to classical applications, blockchain-application show complex relationships ("entanglement") between applications and infrastrusture; governance models can become complicated for a blockchain application, as the application might be "dependent on the governance of the infrastructure"; applications based on permissioned blockchain technology are amenable to established governance models (even if challenging with respect to governance); whereas, research is needed on (new) governance models for applications based on permissionless blockchain technology (in particular, decentralized autonomous organizations)

In [19], Janssen et al. have developed a conceptual framework for the analysis of blockchain technology adoption (as reference point for both practioners and researchers in the field). Through a focused literature review (of a mix of 30 relevant research papers and expert reports) they have identified various relevant factors grouped into the three categories (institutional, market, technical) of the institutional analysis framework of Koppenjan and Groenwegen [10], i.e., *institutional*: norms and cultures, regulations and legislations, governance; *market*: market structure, contacts and agreements, business process; *technical*: information exchange and transactions; distributed ledger; shared infrastructure. Furthermore, they characterize the key challenges that come with each of the several factors and indicate that the factors depend on each other in mutual, complex relationships.

In [20], Nitin Upadhyay compiles a framework for adoption of blockchain technology based on a systematic literature review (805 initially found papers; 89 resulting papers) regarding three research questions, i.e., (i) challenges, (ii) opportunities, and (iii) kind of applications addressed by research on blockchain technology. From the resulting papers, 23 propositions on the adoption of blockchain technologies have been compiled an categorized according to the following groups: innovation characteristics, organizational characteristics, environmental characteristics, user acceptance characteristics. The paper concludes with a compilation of open research questions in the categories of management, impact and application.

With [21], Shermin Voshmgir provides a comprehensive treatment of the past, present and future of the blockchain vision. Based on a thorough review of distributed ledger technologies in its various facets (network, security, control etc.), she aims to explain the disruptive potential of blockchain technology to transcend the Web (as we currently know it) into the vision of a next generation Internet (Web3) based on distributed ledger technology as "collectively maintained public infrastructure"[21] encompassing smart contracts, decentralized autonomous organizations (DAOs), token economics and decentralized finance (DeFi). The proposed concepts are evaluated by a large series of conceptual and real-world use cases from various categories (asset tokens, purpose-driven tokens, tokenized social networks, basic attention tokens (BAT), token-curated registries (TRCs)).

With [6], Andreas M. Antonopoulos provides a comprehensive description of the Bitcoin implementation. After an explanation of Bitcoin basics (terminology, use cases, wallets etc.), he explains, in detail, the data structures and algorithms of the Bitcoin implementation (on the basis of the Bitcoin Core reference implementation), and how they work together in realizing transactions and blockchain consensus. Several further, auxiliary topics such as scripting, security and example blockchain applications are treated as well.

4 Bitcoin as an Uncollateralized, One-Tier M1 Money

4.1 Today's Monetary System

Governments do not print money out of thin air, simply by fiat; instead, they are steering the money supply via a set of complex measures in a tiered, collateralized monetary system – in these endeavors they are supported by resp. team together with independent, legally trusted, accountable institutions. Today's monetary system is tiered. It relies on the interplay of a central bank with commercial banks in guaranteeing the money supply needed for the functioning of a country's economy. Different countries might have different monetary systems, not only with respect to their concrete steering parameters, but also with respect to their design; however, the basic mechanism of money supply is always the same. The crucial point is in the distinction between central bank money and commercial bank money. Only a small fraction of money exists as physical currency (cash/coin), actually, less than 10% of the money is cash/coin. Most of the money exists as pure deposit in banks. Money creation is a permanent process. Money is created and destroyed continuously by the commercial banks via their credit function. Whenever a credit is granted, new money is created. Whenever a credit is payed back, money is destroyed. This means, that it is the commercial bank that creates money out-of-nothing (not the government). A common, often heard, but *false* explanation of the credit function of the commercial bank is as follows: the bank collects money from customers who want to deposit their money; then the bank redistributes the collected money via granting credits; for this service they get paid via the interests of the credits. No; actually, the bank can just grant credits out-of-nothing; however, only within in the narrow boundaries of regulated mechanisms, mainly: collaterals (as described in the sequel) and capital adequacy ratios.

When granting a credit, the bank creates a number in its books: new book money has emerged. Henceforth, the customer who got the credit can use this book money freely in financial transactions; no cash is needed for that. Only institutions that have the legal, official status of a commercial bank – and therefore belong to the network of commercial banks of a country – can grant credits and create book money. Basically, everybody can found a commercial bank. But becoming a commercial bank is not easy, i.e., an institution needs to proof that it is capable of complying with all the necessary regulations and laws of the financial system. Therefore, founding a bank needs a major investment. Whenever a commercial bank grants a credit, it requires a collateral for the credited amount from the debtor. The banks needs that as a security. And the bank is obliged to. It is not up to the bank to decide about collaterals. It is regulated, which kind of assets may be accepted for which kind and which amount of credits; likewise, with which kind of procedures the trustworthiness of the debtor has to be checked, with which kind of procedure the collateral in question has to assessed. Furthermore, a commercial bank needs to hold deposit money at the central bank. This way, the central bank has the role of a *reserve bank*. Only a small fraction of the credit amount granted by a commercial bank has to be deposited by the commercial bank at the central bank, e.g., with respect to the European central bank, this is currently 1%. All this is regulated, and the commercial bank needs to comply; otherwise, it risks its status as a commercial bank and can even become subject of criminal investigations. Banks have regular audits (specialized audits with respect to financial institution laws; far beyond the usual financial audits that are conducted anyhow in each enterprise that is listed on the stock exchange); and they are surveilled by national financial surveillance authorities.

As a result, almost all the money is collateralized in today's monetary systems. This is a crucial concept meant to create particularly stable monetary systems; an un-speculative money is the ideal target. The narrative (which is a mantra told by cryptocurrency evengalists) that, with the abolishment of the gold standard back in the days, governments started to print money out of thin air is not just something like an oversimplification: it is simply wrong. If a credit cannot be paid back, the bank gains ownership over the collateral and tries to turn it into money in order to erase the credit – this is how the collateral works as a security. The buyer of a collateral in this process will usually need a credit for this purchase, and s/he will need to provide a collateral for that new credit: this way the credit system is not merely about the creation and destruction of money, it is rather about a steady transformation of collateral anchors. It is fair to say that almost all money is collateralized; some small credits might not, at least they are usually secured with the reputation of the debtor, i.e., the debtor's creditability, depending on proven income and credit history.

Also, some of the physical currency of the country might not have collaterals. Again, it would be wrong to say (as the cryptocurrency evangelists want to teach us) that with the abundance of the gold standard all physical currency collaterals vanished. Actually, many countries still have gold reserves, e.g., Germany a gold reserve of 3.3367 t (as of December 2019), which amounts to more than 170 Mill. USD[15] (e.g., way more than 10% of cash in Germany, that can be estimated as 1.4 Bill. USD in 2019). But countries do not only hold gold reserves; they hold also other reserves, i.e., reserve currency (i.e., forex: currency of a foreign country) and stock.

The idea behind having (almost all) money collateralized is about stabilizing the respective monetary system. Money owners can trust in the money as they can trust in the real-world assets that are bond to money via the collaterals. This way, money is turned from a purely speculative asset into a less speculative asset. The idea is, of course, that a collateral asset has the value of the money amount that it is securing. This way, we defer our trust into the value of some virtual, monetary asset to trust into the value of some real-world asset. The real-world asset is there and can be assessed, by the human decision maker; and this is where the stabilization of the monetary system comes from. The system has flaws, of course. Indeed, it leads to a recursive problem as follows. Even in the light of existing regulations and laws, it cannot be guaranteed that the collateral assets actually have the value that they secure, i.e., the fact that they do so is again speculative. Still, the circumstance that a collateral asset might not have the value as it supposed to have, is less speculative, than trusting into the degree of acceptance of a purely monetary value. The described recursive trust problem is intrinsic, and cannot be resolved easily. A further threat to the system is in failure of the collateral assessment procedures that the commercial banks are obliged to – but this is an external threat to the concept of the tiered monetary system. An over-assessment of collateral assets is exactly what have happened in the subprime mortgage financial crisis in the years of 2007–2010.

The stability of today's monetary system can be threatened – by its still intrinsic speculative components (which cannot be eliminated completely due the recursive nature of the collateral trust problem) and by non-compliance issues of human actors. Still, even if the monetary system cannot absolutely guarantee stability: it is designed for stability; and each new designed monetary system needs to be as stable as the established one; only after that, it can be even more stable. Cryptocurrency evangelists sometimes teach us about further threats to the established monetary systems, e.g.: that a criminal government might deliberately provoke a financial crisis in order to gain absolute control over its citizens (the conspirative nature of such and similar theories makes it hard for us to bring arguments against or in favor of such theories).

4.2 Cryptocurrency and Today's Monetary System Compared

The tiers in today's monetary systems are called M0, M1, M2 and M3. The exact definitions of which kind of money is included in which tier varies from country to country (also, their might be further tier names such as MB and MZM in the US system), however, this should not concern us too much here, as there are strong similarities between the systems. Roughly, it can be said that M0 money consists of commercial bank deposits plus cash deposits at commercial banks, M1 money consists of demand deposits (book money that can be instantly accessed by the owner for financial transactions) at commercial banks plus circulating cash (in wallets and cash registers), whereas M2-M3 contains saving, time deposits, large-time deposits and others. So, it is fair to say that M0 is the central bank money, M1 is the money that is used in *purchasing* (buyer/seller financial transactions), and M2-M3 is about all kinds of other money. For us, this rough (imprecise) categorization is enough to characterize Bitcoin (and any kind of cryptocurrency that follows the Bitcoin paradigm) as M1 money: it can be used in purchasing (in so far and as long as it is accepted in payments in exchange for goods or services, compare with Sect. 2.1). In so far, it is like cash and book money. It is an electronic asset. In so far, it is like M1 money, as, in practice, today's book money is always realized electronically.

We characterize Bitcoin as a *one-tiered, uncollateralized money*. Bictoin lacks a crucial function of today's regulated monetary systems: the *credit function*. In today's monetary systems the credit function is bound *sine-qua-non* to the creation of collateralized money. And this is the crucial insight: as everyone is used to

[15] with a gold price of 1519.50 USD as of 31th December 2019

talk about collaterals of a loan, it is important to see that it is the *issued* money that is collateralized – collateralized money. In the same vein, Bitcoin is an uncollateralized money. Money creation (money issuance) in the Bitcoin system is a side-effect of the proof-of-work consensus mechanism (as incentive for the so-called mining efforts). But via mining, no collateral is bound to the created (mined) Bitcoin. Therefore, Bitcoins are *purely speculative* assets.

5 Modes of Business Communications

From a purely observational perspective, economy manifests in interactions between actors. Economy can be considered as a the entirety of these interactions. Looking at an economy merely as a huge play of interactions would not equip us with any better understanding of it. Here is, where theory building needs to start, eventually, to come up with hypotheses on structures and laws governing the interactions. The transactions of transaction cost economics are interactions. But they are also composed of many smaller interactions. Business interactions show in business communications. When it is said that a certain (emerging) IT technology would lower transaction costs, this can mean several things – depending on which *institutional level* is primarily affected.

It could mean that the technology is disruptive in a *revolutionary* manner (the "blockchain revolution" is something that is heard often) at Williamson institutional level L2 – compare with Table 1. Its emergence would allow for opening *protected* rooms in the society/economy in which players do not need to adhere to the established "formal rules of the game – esp. property (polity, judiciary, bureaucracy)" [9]. Opening such protected (private) rooms is in the tradition of *cypherpunks* (see "A Cypherpunk's Manifesto"[16] by Eric Hughes [22] and "The Crypto Anarchist Manifesto"[17] by Timothy C. May [23]) and it is fair to count leading figures of the blockchain community to the cypherpunk scene, e.g., Hal Finney (inventor of RPoW (reusable proof-of-work; receiver of the first Bitcoin transaction).

Then, in these opened rooms, new forms of organizations and co-operations would become possible at level L3 ("get the governance structure right" [9]) that would drastically decrease or even eliminate transaction costs (a recent example in that vein is the Ethereum *decentralized autonomous organization* (DAO) [24]). We call such arguments L2/L3-level arguments. We learned form transaction cost economics why people might form organizations at all: to separate transactions into external and internal transactions with the purpose of *economizing*. Now, L2/L3-level arguments (on reducing transactions costs) seem to have an opposite direction: decreasing transaction costs by *deconstruction* of established forms of companies hand-in-hand with deconstructing the established institutional stack.

The other strand of argumentation is more straightforward. It is about increasing the *efficiency* and *effectiveness* of business communications, directly, by the exploitation of best available tools. If established forms of communications are simply replaced, the changes are just about "getting marginal conditions right" [9], i.e., happen at level L4, compare with Table 1. Sometimes, business processes need to be re-engineered on behalf of the introduction of new IT technology. Actually, technology is a major driver in the *business process re-engineering* of Hammer and Champy [25], compare also with [26–28]. Then, there are not only cost-savings with respect to the business communications themselves, but indirect cost-savings due to the improvement of business processes. Business process re-engineering can amount to deep organizational and cultural changes of the organization [29] and, therefore, can also show impact at level L3, but still in the framework of the institutional environment set by level L2. Therefore, we call these kinds of arguments L3/L4-level arguments.

In this section, we conduct the discussion *bottom-up*: we identify essential business communications and analyze how and why they are realized with standard IT technology approaches, see Table 2 for an overview. We call these essential business communications *modes of business communications* or just *modes of communications* for short. The aim is, that (on the basis of such an analysis) it becomes easier to assess the potential of blockchain technology. The idea is to bring the arguments about established and emerging IT solutions that are around into a more systematic and coherent form. The modes of communication are:

[16] https://www.activism.net/cypherpunk/manifesto.html
[17] https://www.activism.net/cypherpunk/crypto-anarchy.html

	Mode of Communication	Addressed Problem
i	Message Authentication	forgery-proof message
ii	Digital Document	*sender-proofs-sender* timestamp
iii	Digital Signature	*sender*-non-repudiation
iv	Digital Registered Letter	*receiver*-non-repudiation
v	Digital Contract	*sender/receiver*-non-repudiation
vi	Digital Order	repeatable *sender/receiver*-non-repudiation
vii	Smart Contract	*automatic* contract enforcement

	Mode of Communication	Classical Trust/Enforcement Anchors
i	Message Authentication	trustworthy public key exchange
ii	Digital Document	trusted TSA (timestamping authority)
iii	Digital Signature	trusted CA (certification authority); (trusted TSA)
iv -a	Digital Registered Letter	trusted VAN (value-added network) provider; framework contract (VAN with receiver)
iv -b	Digital Registered Letter (partially de-centralized)	legally trusted CA/TSA; obligatory (legally enforced) de-centralized transaction logs
v	Digital Contract	trusted VAN provider; framework contract (VAN with all parties)
vi -a	Digital Order (symmetric)	trusted VAN provider; message-related framework contract (VAN with all parties); business-related framework contract between all parties (justiciable business rules)
vi -b	Digital Order (asymmetric)	(trusted or (at least) trustworthy CA); business-related framework contract between dominating consumer and dominated supplier; supplier portal at consumer site (via certified IT service provider)
vii	Smart Contract	– xxx –

Table 2. Modes of (trusted) digital business communications together with addressed problems and classical trust and enforcement anchors.

message authentication, digital record, digital signature, digital registered letter, digital contract, digital order, and smart contract. We will walk through all of them step-by-step, but first we need to explain the importance of the concept of *trust* for business communications.

Trust is essential in business communications, but we need a deeper understanding of the difference between just *trustworthy* and *trusted* parties, in order to understand, how the issue of trust actually shapes business communication solutions. If parties trust each other *unconditionally*, we usually would say that they consider each other as *trustworthy*. Scenarios of unconditional trust ("gentlemen agreements" [10]) can be considered as non-standard scenarios in the professional business world. Rather, parties would like to co-operate with each other, even if they do not trust each other unconditionally. Actually, when it comes to *auditability*, a party must show that it does not simply trusts other parties unconditionally! Here is, where the concept of *trusted party* emerges. The concept of trusted party makes sense only with respect to the judiciary system. A party is trusted, if other parties have "enough reason" to *believe* that the party's witness statements would be *believed* in court cases (dealing with disputes involving business communications). The level of formality of what is considered as "enough reason to believe" can greatly vary. For example, it might be formalized by requiring certain certificates on the (technical) maturity of the trusted party, such as would be required by auditors. At a very formal level, we would say that a party is a *legally trusted party*, if it has been directly granted the status of an officially trusted party by respective legal regulations, or (indirectly) granted by a state authority in accordance with respective legal regulations. A standard example for such legally trusted parties are the certification authorities of national eID (electronic identification) solutions [30–33].

Less precise, but in the same vein, we can talk about a trusted IT solution, if it allows for reliable/stable/replicable argumentation in court cases involving business communications that are supported by the solution. Digital

business communication solutions are large-scale solutions that consist of a technical design and an institutional design, compare with Koppenjan and Groenewegen, 2005 [10] (with respect to [10], compare also with [34]). We call the components (trusted parties; auxiliary technical/non-technical assets) of a trusted solution also *trust anchors*.

Business parties turn to court to enforce their interests (in case of dispute). The judiciary serves as enforcement system. The judiciary system might not be the only possible enforcement system. Interests can be enforced by dominating (powerful) players in asymmetric business relationships. Similarly, the *risk of damaged reputation* can serve as (trustworthy) self-regulating enforcement mechanism in some business sub communities. Such business communities are often (relatively) small, at least, they show a low degree of anonymity; often they show also an asymmetric distribution of powers among their players. Both of the described scenarios are independent of the judiciary, instead, they rely on informal rules at level L3. Now, the envisioned business models (products, platform etc.) based on smart contracts usually heavily rely on the idea of establishing an enforcement system independent of the judiciary. In general, IT solutions for business communications can be considered as composed of enforcement anchors. Trust anchors are a special, ubiquitous kind of enforcement anchors. (Enforcement anchors are relevant to an existing enforcement system. The enforcement system of trust anchors is the judiciary.) Table 2 shows the different modes of communication together with the business communication problems that they address, on the one hand, and the classical trust resp. enforcement anchors of their stereotypical IT solutions, on the other hand. Message authentication (i) is a basic mode of communication that is independent of the existence of a judiciary. The vision of smart contracts (vii) targets the independence of the judiciary and so do the usual smart contract business models (products, platforms etc.), although, smart contracts could also be useful in emerging (classically) trusted solutions. Solutions for the modes of communication (ii) through (vi) are usually always designed against the background of the judiciary; although, some solutions might not be (perfectly) designed with respect to it and/or not oriented towards it – we will discuss such examples also, with solutions (iv-b) and (vi-b).

Message authentication (i) is a basic mode of communication that addresses the problem of forgery-proof message. The business parties unconditionally trust each other, but they have the problem that some intruder might tamper a sent message, or might sent a message claiming to be one of the parties. The problem can be solved simply by trustworthy public key exchange so that, henceforth, a sender can authenticate a message with its private key. We could call the authentication of the message a digital signature, but it is not. The authentication of the message serves only the protection against a forgery attack and is (not yet) feasible to proof that a sender has sent a message, exactly, because of the lack of a trust anchor that will be part of *digital signature* solutions (iii). The simplest form of trustworthy public key exchange is to meet in the real world. Also, they could be send via any other channel that is already considered sufficiently trustworthy (e.g., snail mail).

Several definitions of electronic documents exist, each coming with different intentions: from the domain of libraries (ISO 2709, MARC , Dublin Core etc.) over technical exchange formats (ODF, PDF etc.) to enterprise computing (ISO 10244, Moreq etc.), compare also with [35, 36]. In this paper, we use *digital document* (ii) as the name of a communication mode. The purpose of a digital document is to add a timestamp to it, so that a party can proof that it created/possessed a *record of information* at a certain point in time. A standard use case of this is about saving *intellectual property rights*. A timestamp is a trusted piece of information about such point in time. To stay in the picture of communication modes, we call the addressed problem *sender-proofs-sender* timestamp, because it is about one party (sender) that wants to timestamp a information record (message). This becomes clear, when we look into the standard solution for this problem. In the standard solution, the interested party sends an information record to a *trusted timestamping authority* (TSA). This TSA adds a timepoint information to is, authenticates the resulting document with its private key and sends it back to the interested party. Actually, the TSA has *digitally signed* (iii) the information record, just, the addressed problem is slightly different: the senders themselves want to proof that they have sent messages, not a third party receiver. An alternative solution (Guardtime) that exploits a publication mechanism is described in Sect. 7.1.

Digital signatures (iii) address the problem of *sender*-non-repudiation as follows. The receiver of a message wants to proof that the message has been sent by the sender, often in combination with proving the timepoint

of when the message has been sent. Use cases emerge whenever the sender has confirmed/approved something (e.g., a business certificate) or has committed to something (e.g., an order, compare with (vi) below). The standard solution works as follows. The senders sign the messages with their private keys. The role of the trusted CA is to confirm and to witness that a public key belongs to a sender. The sender and the trusted CA together generate the private/public key pair of the sender. Henceforth, the senders uses their private key to sign documents. A receiver can request a *public key certificate* (public key of the sender plus sufficient identity information about the sender; signed by the trusted CA with its private key) from the trusted CA (technically, typically via OCSP (Online Certificate Status Protocol) [37]). So far, with this confirmation, a scalable solution for message authentication (i) has been achieved. Requesting a public key certificate is a means of scalable, trustworthy public exchange. But with the trusted CA even more has been achieved. The trusted CA (or the issued public key certificate on its behalf) can serve as witness at court; and this way, *sender*-non-repudiation is achieved. Often, a digitally signed document also needs to contain a timestamp. But the timepoint of signature can be added by the sender before signing, there is no need for a trusted TSA, because the addressed problem is *sender*-non-repudiation. It is the receiver who wants to prove that s/he has received a message at a certain time; if s/he receives a message with a wrong timepoint s/he can just ignore it as if not sent (getting a document with the correct timepoint is a different issue; however, it makes no difference whether the receiver aims to get that for the first time or after s/he has received a message with a wrong timepoint). Practically, we see that digital signature solutions also include a trusted TSA, and typically the trusted CA would (at the same time) take the role of the trusted TSA. This is so, because digital signatures usually appear in solutions for the more complex business communications *digital contract* (v) and *digital order* (vi) that we will discuss in due course.

The digital registered letter (iv) addresses the problem that a sender would like to proof that the receiver has received a message at a certain timepoint (in snail mail it is the resp. postal services organization that serves as witness: registered letter). We call the problem *receiver*-non-repudiation. The problem of *receiver*-non-repudiation is dual to the problem of *sender*-non-repudiation. It is often overlooked, that it is important to distinguish cleanly between *receiver*- and *server*-non-repudiation. Actually, the problem of *receiver*-non-repudiation is much harder to solve than the problem of *sender*-non-repudiation. It is not a sufficient solution that the sender proofs that s/he has sent a message (as in *sender-proofs-sender*-timestamping (ii) above) – even if s/he can proof that, this does not proof that the message actually arrived at the receiver. The receiver could always deny that s/he has received it, if the message is simply sent via an ordinary open network. The *ad-hoc* idea to require a signed acknowledgment message from the receiver makes no sense: the problem is exactly about the receiver claiming that s/he has not received the message (so s/he just would not send the acknowledgement message as promised). A solution to this problem can be provided by the concept of *value added network* (VAN) that has been used in typical EDI (electronic data interchange) [38] solutions in the 1990s.[18]

In the VAN-based solution (iv-a), the sender does not send a message directly to the receiver, instead, s/he sends it to the trusted VAN provider with the receiver as addressee. The VAN provider stores the message in a persistent message queue (i.e., technically, the VAN is a MOM (message-oriented middleware)) where it waits for being picked from the receiver. This means, that the message delivery to the receiver is switched from *push* to *pull*. Now, in case of dispute, the VAN provider can witness that the receiver has *either* picked the message (at a certain timepoint) *or* did not try to pick it from the queue. In both cases, the receiver cannot simply claim any more that s/he has not received the message, which provides *receiver*-non-repudiation. It is the responsibility of the receivers to pick their messages from the VAN. Such responsibility can be formalized in a framework contract, in which the receiver commits to pick messages regularly. Here, it needs to be the interest of the receiver to commit to such responsibility. The interest exists, because the

[18] The uprise of EDI (electronic data interchange) was hand-in-hand with the *deconstruction of the value chain* as described by the Boston Consulting Group. After the millenium, EDI was superseded by B2B (business-to-business). Technologically, B2B was about a change to the SOA (service-oriented architecture stack) [39–41, 36]. Conceptually, it was about the vision to transform asymmetric business relations into more symmetric business relations on a large scale – as such, B2B never took off. Independent, of whether EDI or B2B, the conceptual building blocks of IT solutions remain the same.

receiver is interested to get engaged in certain business communications (and therefore transactions), i.e., contracts (v) and orders (vi) below. No trusted CA is needed in this solution. The trusted VAN can identify both the sender and the receiver by their *user credentials*. Sender *"send"* messages to the VAN provider by logging in into the provider's systems (of course, as part of a secured API call) with their user credentials, same with the receivers when they pick their messages. Therefore, the VAN provider exactly knows, who and when has sent/received which messages via the VAN.

Other solutions can be designed to solve the problem of *receiver*-non-repudiation. Here, we describe a *partially decentralized solution* (iv-b) as found in the nation-wide data exchange layer X-Road[19,20] [42–44], which is the enabling backbone of the Estonian e-government ecosystem, compare also with [45–47, 34]. X-Roads is based on a PKI (public key inftrastructure) for nation-wide e-identities and digital signatures and therefore has a *legally trusted* CA that at the same time serves as legally trusted TSA. Now, each authority that participates in X-Road as an *information system node* maintains a transaction log (complete log of all ingoing/outgoing X-Road messages) [44]. The transaction logs provide sufficient proof to ensure *receiver*-non-repudiation. The maintenance of transaction logs is implemented in the standard implementation of the X-Road security server.

We define the digital contract (v) as the communication mode, in which both parties agreed on and committed to the content of a (timestamped) information record, so that the addressed problem is both *sender*- and *receiver*-non-repudiation in sending and signing the contract back and forth. A solution can be designed, e.g., again with a VAN as for *receiver*-non-repudiation; this time, all parties need to be obliged to pick their messages from the VAN provider.

The *digital order* (vi) is the essential communication mode in B2B e-commerce. A digital order is a *contract* between an *consumer* (sender) and a *supplier* (receiver). The contract is considered accepted by *both* parties, if it has been sent by the sender and has been received by the receiver. Both parties need to commit to that mode; therefore, it needs a business-related framework contract that also regulates further justiciable business rules with respect to the orders (how much will the consumer order at least per month? how much can the supplier deliver at most per month? how fast will products be delivered? etc.). If the business-related framework contract is not a digital contract, a solution for digital orders just needs a solution for digital registered letters (iv) as a basis. Otherwise, it needs a solution for digital contracts (v) as a basis. In practice, digital orders are more complex and contain also messages send to the consumer (e.g., order acknowledgement, compare with resp. EDI standards) that need to be provable. Therefore, digital orders usually need a digital contract solution as their basis anyhow, compare with (vi-a) in Table 2.

In practice, other solutions for *digital orders* (vi) exist. (In practice, even in mission-critical supply chains, orders are often just sent by email with little to no systematic measures for resilience – the confidence in the enforceability is accordingly low.) Today's supply chains often show asymmetric business relationships, i.e., business relationships with a dominating (powerful) consumer and a dependent (much smaller) supplier.[21] In such scenarios, we can see solutions (vi-b) as follows. Again, there is a business-related framework contract, however, this time the suppliers commits to pick orders directly from a server (as part of a supplier portal) at the customer site (e.g. via Web service, REST service etc.). A sufficiently trustworthy CA, as well as a certified IT service provider who runs the supplier portal, can be part of the solution. The objective of such solutions is to streamline and advance the consumers procurement processes (supplier "enablement") and not, in first place, the legally solid proof of messages – as other enforcement mechanisms (concerning dependency/reputation of the supplier) gain more importance.

Smart contracts (vii) address the problem of *automatic* contract enforcement, i.e., they aim at providing a contract enforcement system as part of their solution, please see Sect. 2.3 for an explanation of smart contracts. Smart contracts (vii) are fundamentally different from the other modes of communication (i) through (vi); and there exists no classical solution for them. Rather, smart contracts can be identified with

[19] https://www.ria.ee/en/state-information-system/x-tee.html

[20] https://x-road.global/

[21] A great deal of the millennium B2B vision with its UDDI (Universal Description, Discovery, and Integration) *yellow pages* approach was about *"democratizing"* business relationships. Democratizing business relationships is an ongoing theme. Actually, many ICOs rely on it in their business model visions.

today's canonical solution that is provided for the problem that they address. The smart contract solution consists of

- a (typically permissionless) distributed ledger (blockchain for short)
- a cryptocurrency (realized by the blockchain)
- a formal contract language that is rich enough to express *self*-enforcable (*smart*) contracts (in the boundaries of the solution) in terms of
  - information in the ledger
  - accessible information outside the ledger (*basic* real-world embededdeness)
  - accessible (steerable) assets outside the ledger such as locks, signals, machines, i.e., all kinds ot IoT (Internet-of-Things) devices (real-world embeddedness)
- Smart contracts that are written in the contract language and stored in the ledger

A prominent smart contract implementation is the distributed ledger platform Ethereum with the cryptocurrency Ether and the contract language Solidity. Now, how can a smart contract solution help in implementing the essential business communication modes that we have identified in Table 2? We discuss this for the communication mode *digital order* (vi) as it is the most encompassing one and can be considered the essence of a stereotyical ICO aiming at *disrupting* the "way business is done" in a certain business domain or in general. More concrete, we review the enforcement anchors in (vi-a) to conduct the discussion.

First, we want to discuss a potential solution that is "maximally" disruptive, i.e., aims at enabling business independent of the established judiciary and currency, and independent of any dominating player. The goal has to be that neither a trusted CA nor a trusted VAN is needed. Consequentially, the message-related contracts (VAN with all parties) will also become obsolete in the solution. Now, both business-related framework contracts and orders are stored as *smart contracts* in the blockchain. Now, the problem of *sender/receiver*-non-repudiation that is addressed by the *digital order* (vi) is simply not an issue any more. The messages are recorded – tamper-proof – in the ledger. The information in the ledger can be used in contract enforcement. A question could be, whether the ledger information would be accepted by the judiciary. We will turn to that later, but for now, it is not a question at all: we have said that the solution is about getting independent of the judiciary and establishes its own enforcement system.

What happens if a contract is not fulfilled? It is the solution itself that reacts, according to what has been specified in the smart contract. Types of reactions can be about *enforcing* well-behavior, *punishing* misbehavior (penalty) or *compensating* misbehavior (escrow). All cases of potential misbehavior need to be understood and appropriate reactions have to be programmed. For example, the product of an order could delivered with an IoT lock (that is accessible from the contract language). The lock is only unlocked, after the payment has been made. All of this can be programmed before the smart contract is signed. But what happens if the product (that has been unlocked) is not working? The idea is that then the payment for the product is *automatically* paid back. But how can the ledger decide that the product is not working according to its specification? Can we program an automatic surveillance of the product (via sensors) that decides that? Rather not. The parties can decide beforehand about a neutral *arbitrator*, whom they both trust and who is asked (automatically) to decide in that case by reviewing the product in the real world. Does this still adhere to the initial vision of fully automatic self-enforceable contracts? Is it not too complicated, is it not even impossible to understand all potential eventualities and to design and program appropriate reactions to them? Each transaction could be secured with an escrow. Escrow mechanisms are well-known from established business communications. But only a small fraction of business transactions is actually secured by escrows in today's business world (rather, penalties are fixed in contracts – to be enforced by the judiciary). Securing each and every order in B2B is simply not possible, because for this, the needed financial *liquidity* would be way beyond what could be considered reasonable – we would like to talk about *escrow-superheaviness* in case of such an approach.

To overcome the questions posed in the previous paragraph, a middle position is possible that aligns a smart contract solution with the established judiciary and official currency. Here, the underlying ledger would be permissioned. In the extreme case, in which the ledger is provisioned by a single player, the solution does not differ a lot from a classical VAN-based solution. It still could be argued that a chosen smart contract

platform is (with its specific features) particularly appropriate for the indicated domain of digital orders and/or that is favorable for other reasons (scalable, resilient, manageable, contemporary, future-proof etc.). Conceptually, stepping from a single provider to a group of providers does not change a lot. It could be argued that the resilience and/or the "trustworthiness" of such a permissioned solution is particularly high. However, also the opposite can always be claimed, in particular, because there exist (to our best knowledge) yet no documented approaches to make distributed ledger technologies auditable (to be assessed positively in operational risk management audits).[22] Similarly, a solution can only be as trusted as it is *believed to be believed* in court cases (see the discussion earlier in this section) and again, from our current perspesctive, we claim that there is still a severe lack of proven experience/approaches/standards that would provide a sufficient level of *trustability* for distributed ledger technologies.

6 Intrinsic Technological Barriers

6.1 Scalability and Denial-of-Service

In this section, we investigate blockchain technology that follows the original blockchain paradigm, i.e., proof-of-work [1]. The findings are also relevant in discussions of alternative blockchain paradigms (following other consensus mechanisms), as they reveal fundamental *requirements* for (permissionless) blockchain-based solutions.

There are several, mutual dependent technical barriers intrinsic in blockchain technology, i.e., protocol-related *denial-of-service* (DoS) attacks (both from the *user* and from the *operator*), protocol-related *scalability* (both with respect to *usage* and with respect to *operation*) and the possibility of *double-spending*. We say that the potential DoS attacks are protocol-related (and therefore intrinsic), because they arise from the design of blockchain technology itself, i.e., they are not just DoS attacks that threaten every IT system that operates in an open network. The same is for protocol-related *scalability*, where the potential of *double-spending* is specific to blockchain technology anyhow.

If the block size of a blockchain technology is not limited, trivial DoS attacks are possible (for the user, if transaction costs are achievable; for the operator always). Limiting the block size leads to limited scalability. We define:

N	maximal (limit) number of transactions per block
N'	average number of requested transactions per block
T	average time needed for mining a bock
N/T	available performance of the blockchain
N'/T	requested performance of the blockchain
N'/N	load of the blockchain
$N'/N > 1$	overload of the blockchain
τ	threshold value of N'/N rendering the blockchain impractical
τ'	threshold value of N'/N collapsing the blockchain

(1)

The available performance of the blockchain N/T in (1) is neither a peak performance nor a maximal performance measure. As T itself, N/T is an average measure and makes sense only when considering a sufficiently large time period (sufficiently large number of (mined) blocks).[23] The same is with the measures N'/T, N'/N etc. The requested performance of the blockchain N'/T comes from transactions *requesting* to be added to the blockchain. Whenever N'/T is larger then N/T, we talk about an *overload* of the blockchain ($N'/N > 1$), characterized by the existence of waiting/dangling transactions. An overload of the blockchain is not a problem *per se*, it becomes a problem, when the requested performance is way larger than the available

[22] The fact that many ICOs might propose blockchain technology as a solution to increase accountability in organizations does not mean that blockchain technology itself is auditable.

[23] For an ongoing statics of the Bitcoin blockchain, its size, average blocks sizes, average transactions per block, total number of transactions etc., see:
https://www.blockchain.com/charts/

performance ($N'/T \gg N/T$). The threshold τ marks the value of N'/N, from which the blockchain is *perceived* as impractical. It is not possible to determine the value of τ theoretically, only *empirically*, i.e., it is a value determined by the users of the blockchain. The threshold τ' marks the value of N'/N that would lead to a collapse of the blockchain, meaning, that the value of its cryptocurrency assets massively diminishes (e.g., by users starting to withdraw their cryptocurrency assets on a massive scale). Obviously, we have that $\tau' \geqslant \tau > 1$. Typically, we are interested in τ, when analysing scalability, whereas we are interested in τ' rather when analysing potential DoS attacks. More precisely, let us assume that N' is composed of transactions n and n' (i.e., $N'=n+n'$) as follows:

$$\begin{array}{ll} n & \text{average number of requested non-incidental transactions per block} \\ n' & \text{average number of requested incidental transactions per block} \end{array} \tag{2}$$

Now, we could say that both τ and τ' are analyzed as scalability issues if we assume that $n' \approx 0$, whereas we would analyse them in service of understanding potential DoS attacks if we assume that $n' \gg n$. A typical countermeasure against incidental/critical overload $N'/N \gg 1$ can be seen in the introduction of a lower bound (so-called *dust limit*) for the amount transferred by a single transaction;[24] – at least against *ad-hoc* occurences/attempts. Another protection against incidental/critical overload can be provided by transaction fees, as a side-effect of their development, and only in so far as they have been developed.[25] In both cases (dust limits and transaction fees) there exists the question of the *sweet spot* between protection against overload and enablement of microtransactions. Too high transaction costs contradict microtransactions. Note, that the enablement of microtransactions was among the main objectives of the original Bitcoin vision [1].

An operator (or group of operators who conspire) with sufficient mining power can attack the blockchain. Actually, more precisely, each operator can always (try to) attack the blockchain, but s/he needs sufficient mining power to be successful with it. A well-known kind of attack is the so called *double spending* attack [1]. Here, the attacker replaces a previously done transaction by an updated one (here, the replaced/deleted transaction was spent for a good/service that has been delivered to the attacker in the mean time). The essence of the double-spend attack is that the attacker changes the blockchain history permanently; the fact that this might be done in service of double-spending is just an instance of that more general scheme. We would like to speak of an *overwrite attack* instead. If an attacker is successful with an overwrite attack, the attacker can overwrite not just one certain transaction in the first block of the recent changed history (double-spend) but all transactions in all blocks in the recent history. For instance, if the attacker creates blocks that consist only of (void/dummy) transactions, this would amount to a severe DoS attack. The described attack is more than a plain denial-of-service attack. It has some negative impact in terms of availability, but its actual damage is in the level of *business confusion* that arises from all the reverted/lost transactions. We assert that, in case of Bitcoin, it is rather unlikely that an *overwrite* attack is tried in purpose of a *double-spend* attack – given the expensiveness of an overwrite attack (given the current overall mining power currently spent in Bitcoin) as opposed to realistic amounts in business transactions. If an overwrite attack occurs, we would rather assume that it has been done in purpose of a DoS attack (the lost transaction fees can be considered marginal as compared to the costs of the needed mining efforts in such case). An overwrite attack is expensive. The more mining power, the faster it will succeed (in the sense of: expected time in case that it succeeds). To show that, we develop a statistical model of overwrite attacks as follows. We model the mining of blocks in the main blockchain and the attacker's blockchain as two **i.i.d.** sequence of random variables X (main blockchain) and Y (attacker blockchain) that are also *mutually independent w.r.t. each other*, i.e.:

[24] Today, 30th August 2020, the dust limit of Bitcoin is 546 Satoshi, amounting to ≈0.065 USD.

[25] Today, 30th August 2020, the average Bitcoin transaction fee is ≈2.60 USD, i.e., 40 times higher than the dust limit of 0.065 USD. Transaction fees can greatly vary over time, in August 2020 they have been oscillating roughly between 1USD and 7USD. For an ongoing statistics of Bictoin transaction fees, see here: https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

$$X = (X_i : \Omega \longrightarrow \mathbb{R})_{i \in \mathbb{N}} \tag{3}$$
$$Y = (Y_i : \Omega \longrightarrow \mathbb{R})_{i \in \mathbb{N}} \tag{4}$$

We have that Ω is the set of all possible *evolvements* of the blockchain (with respective probability space $(\Omega, \Sigma, \mathsf{P})$ as usual) after the attempt to attack has started. The attempt to attack starts at timepoint 0. Assume that we name the blocks in the main chain and the attacker chain as follows:

$$\cdots x_{-3} x_{-2} x_{-1}\ x_0\ x_1 x_2 x_3 \cdots \tag{5}$$
$$\cdots x_{-3} x_{-2} x_{-1} \qquad y_1\, y_2\, y_3 \cdots \tag{6}$$

The attack starts with mining x_1 and y_1. The target of the attacker is to overwrite x_0. Therefore, y_1 in the attacker block points to the block before x_0, i.e., x_{-1}. This way, y_1 "overwrites" x_0.

Now, a random variable X_i models the time that has been needed to *mine and distribute* the *i-th* block x_i of the main chain (more precise: the *i-th* block after the start of the attack). Now, we make a typical simplification (in service of coming up with a model): we assume that the mining effort in the mining/distribution is so dominant that we can neglect the time needed for distribution of blocks through the network. This assumption greatly simplifies the scenario and our model. It immediately implies that we do not need to model the differences between the timepoints at which a block arrives at different nodes of the blockchain network – we can simply assume that a block arrives at (is distributed to) all nodes at the same time point. Now, the random variable $X^n = X_1 + \cdots + X_n$ models the *timepoint* at which (or: overall elapsed time until) the *n-th* block of the main chain x_n has been mined. Similarly, a random variable Y_i models the time that has been needed to mine the *i-th* block of the attacker chain y_i and so forth.

Note that X and Y are both **i.i.d.** individually. Furthermore, they are mutually independent w.r.t. each other. In general, they are not identically distributed as compared to each other. In the special case that the owner of the main blockchain and the attacker have the same computational power, we have that X and Y are also identically distributed as compared to each other. In all other cases, they are not. If the computational power of the attacker is weaker (stronger) than the computational power of the main blockchain owner, the *mean* of Y is larger (smaller) than the mean of X, as then, *on average*, the attacker needs more (less) time to mine a block than X.

At each point in time, the blockchain protocol decides for the longest chain in our model.[26] The attacker chain needs to pace up with one additional block, i.e., the block x_0 which is omitted from the attacker chain, compare with (6). Therefore, the probability that the attacker succeeds with an *overwrite* has the following value:

$$\mathsf{P}\big(\, \exists n \in \mathbb{N} \,.\, Y^{n+1} < X^n \big) \tag{7}$$

In order to specify the average time needed to succeed with an overwrite, we first define the random variable H of *first overwrite success times* as follows:

$$H \ : \ \Omega \longrightarrow \mathbb{R} \cup \infty \tag{8}$$
$$H(\omega) = \min\big\{ n \mid Y^{n+1}(\omega) < X^n(\omega) \big\} \tag{9}$$

Now, we can specify the *average time needed to succeed with an overwrite* as a conditional expected value as follows:

[26] At each point in time, the blockchain protocol decides for the strongest chain. We assume that the attacker and the main chain proceed with the same difficulty in our model. Note, that a chain of equal length that is formed later does not replace the main chain. (Often heard explanations such as that "the aggreed-upon blockchain is always the chain that represents the most accumulative work" is not true. It is true in the long run, statistically, as alternative chains get orphaned.)

$$\mathsf{E}\big(\, H \,\big|\, \exists n \in \mathbb{N} \,.\, Y^{n+1} < X^n \big) \tag{10}$$

Whenever $\mathsf{P}\big(\, \exists n \in \mathbb{N} \,.\, Y^{n+1} < X^n \big) = 1$, we have that $H : \Omega \longrightarrow \mathbb{R}$ and, therefore, the *average time needed to succeed with a an overwrite* is obtained by $\mathsf{E}(H)$. Given a fixed X, we see that (10) decreases, whenever $\mathsf{E}(Y)$ decreases. $\mathsf{E}(Y)$ decreases, whenever the mining power of the attacker increases.

The value of (10) depends on the kind of distribution of block mining times X (and Y). A natural assumption is that X is distributed approximately *geometrically* as follows. Mining is about repeatedly choosing a value (by random, or by incrementation) as a nonce in the block – until the block's hash meets a specified difficulty target. Now, we need to make two assumptions. First ("unit time"), we assume that the time needed to compute a block's hash is always the same. Second ("memorylessness"), we need to assume that the probability that a block's hash meets the specified difficulty target is equal for each chosen value (which we can, approximately, if the set of values from which we draw nonces is sufficiently large). Based on theses assumption, we would have that X shows a *geometric* distribution ("number of tosses till success").

6.2 Characteristic Neglection of the Physical Network

Blockchains are peer-to-peer networks [48]. Peer-to-peer networks are a distributed application model, in which nodes are clients and servers at the same time. The nodes work together as equal partners (peers) to achieve a common goal/service. Peer-to-peer networks often come with the attitude of *democratizing* the Internet. But peer-to-peer networks are an application model, and nothing but a application model. They are defined, established and discussed merely at OSI layer 7. They are *overlay networks* [49] and they take for granted the existence of a functioning physical network. Figure 1 shows a typical visualization of a peer-to-peer network, but Fig. 2 shows how the *network* actually looks like.

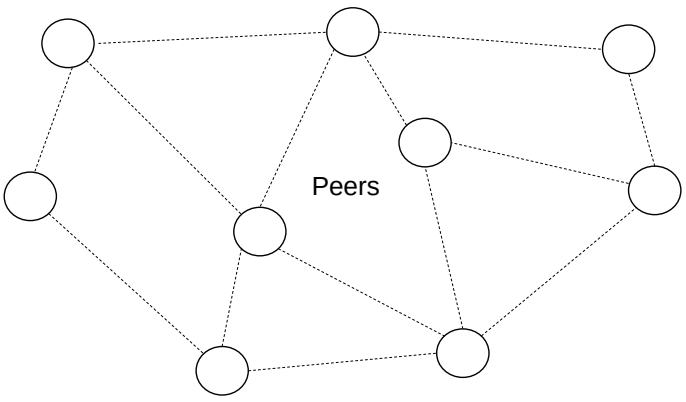

Fig. 1. Typical visualization of a peer-to-peer network.

Sending a transaction from one node to a neighboring node in a peer-to-peer network (Fig. 1) might amount to sending the transaction actually (Fig. 2) via LANs (local area networks), WANs (wide-area networks), IXPs (Internet exchange points), and ISPs (Internet service providers) with their typical NAT translation (network address translation) with all their devices and equipment which are not owned and therefore not controlled by the players in the peer-to-peer network. There seems to be (to be investigated) a characteristic attitude of neglecting the existence of owners of the Internet machinery. The players of the peer-to-peer network are the owner of the nodes. A network failure is considered a failure of a peer. Or vice versa: it is not distinguished between whether a failure of a node has occurred *accidentally* of *intentionally* (i.e., as a an attack conducted by the node). There seems to be a certain attitude to consider the underlying

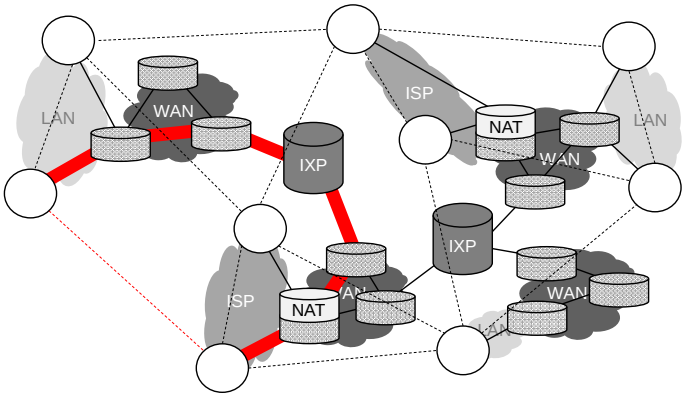

Fig. 2. Peer-to-Peer network together with its underlying physical network.

Internet as an endless *ocean of devices*, but it is not: the Internet is strictly hierarchical, think of the IANA (Internet Assigned Numbers Authority) alone.

What is the problem? For example, there exist only a couple of hundreds of IXPs worldwide[27]. That is not a lot. In a usual discussion of proof-of-work consensus it is considered unlike that 51% of the miners conspire to drive an attack, because there are 1.000.000 miners. But what if the IXPs conspire? For example, by just switching off at the same time (much more sophisticated kinds of attacks can be devised easily). They cannot, you would say, because they are bound by thousand upon thousands of legal contracts (with the autonomous networks). But is it not the peer-to-peer community that mistrusts all central authorities and centralized organizations? Why not the IXPs? Actually, it does not need many IPXs; three or four IXPs might be enough to cut off a whole country from the Internet. The whole point is, that the Internet companies (ISPs, backbone providers, IXPs) are no *peers* in the peer-to-peer network, they are just *suppliers*. The core of the *cypherpunk* scene is totally aware of all that. Playing with mems such as 'piracy' and 'hacking' is actually an essential ingredient of their culture, and the neglection of the physical network (if it exists as perceived by us) is rather a phenomenon in a extended, new generation of the peer-to-peer community. So or so, when it comes to business decisions with respect to blockchain technology, it is important to maintain a technological full stack perspective.

7 Working Blockchain Solutions

The solutions discussed in the section are related to permissioned blockchains. The Guardtime solution in Sect. 7.1 is a document timestamping solution that has been created as early as in 2007 and consists of concepts (distributed Merkle trees) that are today associated with blockchain technology and therefore, the solution is knows as KSI blockchain (keyless signature infrastructure blockchain). The solutions in Sects. 7.2, 7.3 and 7.4 use a permissioned blockchain platform as a component in their software architecture.

7.1 A Tamper-Proof Timestamping Service

Document timestamping solutions that implement *sender-proofs-sender* timestamping (communication mode (ii) in Table 2) are mission-critical in many organizational contexts. Organizations want to have tamper-proof and provable document logs not only in the communications with other organizations; they also want to be safe against failure (accidental or intentional) of their own members/employees. Equally, the state wants to be confident with the operations of its authorities and, again, the authorities want to be confident with the operations of their employees. Since 2007, Guardtime offers a document time stamping solution as a

[27] https://www.datacentermap.com/ixps.html

service, i.e., the so called KSI blockchain (keyless signature infrastructure blockchain). In the Estonian e-government ecosytem, the solution is successfully used to secure the healthcare registry, the property registry, the succession registry, the digital court system and the state gazette [50].

Fig. 3. Root hash of Guardtime's KSI blockchain hash calendar published in the Financial Times (Picture taken from: https://guardtime.com/).

The KSI blockchain achieves a practical implementation of an idea that goes back to Stornetta et al. in 1993 [14], i.e., it stores timestamped document hashes in a Merkle tree [51] and publishes the root hash of the tree *periodically* (e.g., once a month) in a newspaper (e.g., in the Financial Times, among others, in case of the Guardtime solution), see Fig. 3. The idea is straightforward and effective: the published root hash serves as a *real-world trust anchor*. However, to turn this idea into a robust solution that is highly performant and at the same time highly available is a challenge, and this is exactly what is achieved by the KSI blockchain [52, 53] as follows. Time is divided into rounds (e.g., one second). Hashes of all documents signed during a round are aggregated into a per-round hash tree. The root hashes of the per-round trees are collected into a perpetual hash tree, the so-called *hash calendar*. It is the root hash of this hash calendar that then is published periodically to newspaper(s). Now, the Merkle tree is distributed and replicated to make the solution resilient, i.e., the nodes are deployed on several distributed servers. The servers are operated by the KSI blockchain service provider, with a mix of *owned* servers at the premises of the KSI blockchain service provider and servers of trusted *third-party* compute service providers, to make the solution even more resilient.

7.2 An Automated Manufacturing Software Ecosystem

There is a global trend and paradigm shift in manufacturing towards *personal manufacturing* [54]. In this new paradigm, people and organizations would not buy a ready-made product [55]. Instead, they would obtain raw material and produce products using their own (or locally accessible) automated manufacturing (AM) machinery. People and companies have access to a lot of different types of automated manufacturing (AM) machinery such as 3D printers, CNC mills, laser jets, and robotics to manufacture products locally, at the point and time of need. The impressively fast adoption of these technologies, once more boosted during the recent COVID-19 outbreak, even more amplified this increasing need – strongly indicating that this novel approach to manufacturing can become a key enabler for the real-time economy of the future [56].

Already now, software platforms such as 3DPrinterOS [57, 56, 58] and ZAP [59] are available, which cover the whole personal manufacturing workflow from the initial idea to the physical object using real-time command-and-control of AM machinery to manufacture production-grade functional parts. Such platforms automate the whole process, and allow for tracking different parts of a product at different stages of manufacturing such as CAD, simulation, conversion, slicing, 3D printing, supports removal, post-processing, surface

finishing, delivery of parts to an assembly line, assembly process, quality assurance, packaging and – finally –shipping the product to a customer. An important problem is how to ensure information integrity and consistency through all the stages of AM; i.e., an intentional or accidental corruption or loss of information can happen, which can affect the final quality and reliability of product parts and the final product. And if such a product is used in a mission critical environment, this can even affect human life.

In this section, we further explain the 3DPrinterOS platform[28]. The platform realizes a mechanism to securely deliver content to 3D printers from the cloud. The first generation of the solution [60] has been introduced in 2015. Today, the 3DPrinterOS cloud has more than 123.000 users who have generated over five million CAD designs and machine codes and have produced more than 1.500.000 physical parts on 40.000 3D printers in 100 countries; these values double every six months [61]. The technology is licensed to Bosch, Kodak, and other popular desktop 3D printer manufacturers. Currently, the solution is completely reworked [57] and extended to any type of manufacturing machine or complex IoT device with command, control, and telemetry.

In the sequel, we describe how permissioned blockchain technology is used at 3DPrinterOS to ensure the integrity of several critical manufacturing parameters of product parts and end products. We use Hyperledger Fabric [62] as a microtransaction ledger solution for the 3DPrinterOS software ecosystem [56]. The software ecosystem can be understood as a kind of broker: its purpose is to enable several organizations to bring value to end users. The solution is sketched in Fig. 4.

The platform orchestrator *R1* (using the configuration *NC1*) sets up and runs multiple ordering service nodes $O_1 \cdots O_n$ on different infrastructure providers such as AWS, Azure, GCE [63], this way forming the private blockchain network *N*. The orchestrator of the ecosystem *R1* is a network initiator in terms of Hyperledger Fabric and has administrative rights over the network *N*. Both the nodes and administrators of orchestrator *R1* access the network *N* through the usage of X.509 [64] certificates issued by *R1*'s certificate authority *CA1*. Certificates issued by *CA1* are also used for signing the transactions to be accepted into the ledger.

Other players (vendors, end users, external actors) of the ecosystem [56] are added to the network via creation of consortia $X1 \cdots Xn$ (e.g., *X1* and *X2* in Fig. 4); then establishing channels $C1 \cdots Cm$ (e.g., *C1* and *C2* in Fig. 4) for storing information in corresponding ledgers $L1 \cdots Lm$ (e.g., *L1* and *L2* in Fig. 4). The participating players *R2*, *R3* and *R4* establish certificate authorities $CA2 \cdots CAk$ to grant their employees and their technology (IT systems, machines etc.) access to the ecosystem through channel configurations *CC1* and *CC2*. Certificates issued by the organizations' certificate authorities are used to sign all transactions. Chain code is stored in smart contracts $S1 \cdots Sn$ (e.g., *S1* and *S2* in Fig 4). The applications of participating organizations *A2*, *A3* and *A4* can access corresponding channels *C1* and *C2*, as well as the software ecosystem's application platform *A1*. This way, all participating users and assets such as 3D printers, CNC mills, material batches can be identified; and therefore, it becomes easy to track responsibilities even on a large scale. This increases the transparency of the processes, e.g., in case of part malfunctioning, it is easy to track the participants and the involved processes. Established channels are private to participants (thus only participating parties have access). The blockchain keeps all information in the ledgers. Ledger data is updated using smart contracts. Smart contracts are executed only if they are confirmed by all channel participants using a voting-based consensus mechanism. All information in the ledger is tamper-proof, and all information is entered only if digitally signed by a responsible organization. All of these issues would be hard to solve without permissioned blockchain technology.

7.3 An Electronic Identity and Digital Signature Solution

There are just a few countries in the world who offer convenient, fully integrated electronic identity (eID) and digital signature solutions [32] to their citizens. The majority of e-signature services worldwide [65] still use "flinstone" techniques for signing documents such as drawing a signature on the screen, or uploading a picture with signature. There are many barriers to implement contemporary eID and digital signature solutions in many countries in the world including EU in a secure, user-friendly, and scalable manner and

[28] https://www.3dprinteros.com/

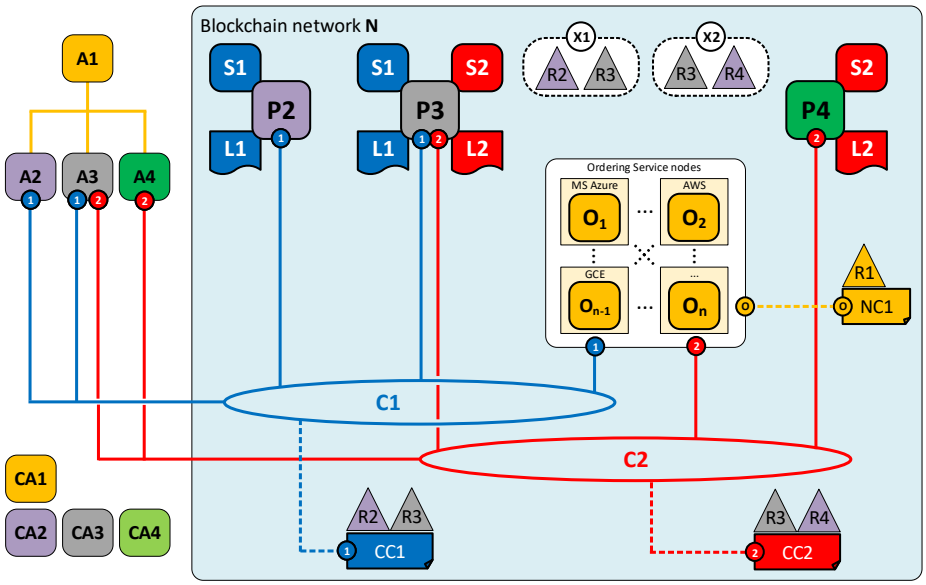

Fig. 4. Example Hyperledger Fabric setup of the 3DPrinterOS software ecosystem.

– as a consequence – to have them adopted by the citizens [66, 33, 67, 31, 68, 69]. As indicated by several studies [70–72], eIDs and digital signatures are used mostly by major enterprises, whereas their usage by SMEs (small/medium enterprises) and ordinary citizens is marginal – due to the lack of a broad range of e-government services and a low usability of the systems. This marginal use of eIDS and digital signatures is mostly in online payments; and only a fraction is in contract signing [70, 65]. With the advent of blockchain technology, easier and more secure solutions potentially became possible, which would allow for a seamless user experience and would show (at the same time) a high scalability.

An example of such a solution is Agrello ID [73][29] – a blockchain-based solution for eID and digital signature. We describe the solution in detail in Fig. 5. The solution is based on elliptic curve key threshold encryption [74, 75], where one part of the key is kept inside a smart device such as a smartphone, and the other part is stored in the cloud. When these keypairs are combined, they are capable of providing a fully functioning digital signature behaving in accordance with the eIDAS regulation [76].

Each physical object in the world is matched with a digital twin in an instance of the Ethereum [77] blockchain. A person's digital identity is a smart contract in the blockchain; also, objects such as cars or apartments have digital twins in the form of chaincodes (smart contracts). Each participant of the private network has a right to publish information about the identities of such objects. More specifically, the identity of a person is confirmed by a *know your customer* (KYC) provider [78]. Each confirmation and validation is done by executing chaincode on the digital identity smart contract, which activated the digital identity, see Fig. 5. By using private keys, the owners of a digital identities can perform operations such as signing service or sales contracts.

We analyzed numerous CA software products and found that none of them is scalable enough for our purposes: country-wide authentications and digital signatures with millions of citizens performing many thousands of operations per day. Also, the used solution should be future-proof. We found that solutions based on permissioned blockchains have these capabilities, as they are already tested in the field. For Agrello, we use a private deployment of Ethereum[30] with proof of authority.

[29] https://www.agrello.id/
[30] Ethereum verions 1.7.3 and 1.8.1

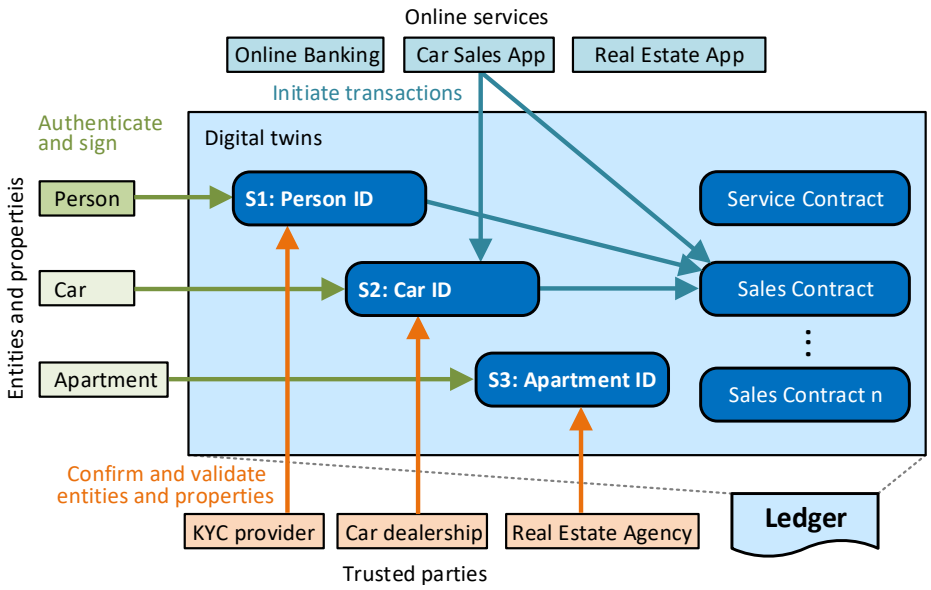

Fig. 5. The Agrello eID and digital signature solution – conceptual view.

7.4 A Sensors-as-a-Service Marketplace

Permissioned blockchains can enable completely new business models. In particular, they have the potential to allow for calculating and accounting microtransactions more precisely. Still, today's mictotransactions are often quite costly to maintain and audit. Blockchain technology has the potential to make this easier. An example is Thinnect[31], a SolaaS (Solution as a Service) platform that allows to use a comprehensive hardware utilization and monitoring solution (including cloud-based connectivity dashboards for hardware such as temperature sensors or CO2 sensors) using a pay-per-reading business model. Often, high quality hardware is expensive and value can be created for the end customers only by utilizing a full-fledged solution; therefore, businesses who need to make specific parameter measurements (food logistics, smart cities etc.) need to invest a considerable amount of money to acquire hardware and to keep networks of sensors up and running reliably – outsourcing is not a solution here, as it is equally expensive or even more expensive.

A way out of this is yielded by a SolaaS model where companies only pay for the value they get - they *pay per measurement* and are not involved at any stages of hardware purchasing, delivery or deployment. Hardware is subsidized by the hardware manufacturer or a third party willing to invest and get interest as shown in Fig. 6. Each party is paid trough micro-payments for each measurement. Solutions based on classical database products and traditional ways of calculating and auditing the ledgers can take a lot of time and money, because separate databases at different locations (at participants' premises) can create a lot of trouble in synchronization and finding out truth in auditing processes. Blockchains have the potential to make all of this much easier due to their consensus mechanisms, uniquely identifiable and digitally signed transactions, as well as the fact that they continuously check balances by their design. Thinnect uses Hyperledger Fabric [62] blockchain as a micro-transaction ledger solution to keep track of the sensors readings batches, nonpayments from actual users of hardware to Thinnect, and micro-payments to hardware manufacturers and investors subsidizing the cost of hardware. To summarize, blockchain technologies have the potential to bring many advantages to micropayments and nanopayments as needed in solutions such as Thinnect [79].

[31] https://www.iot-inc.com/portfolio-page/thinnect/

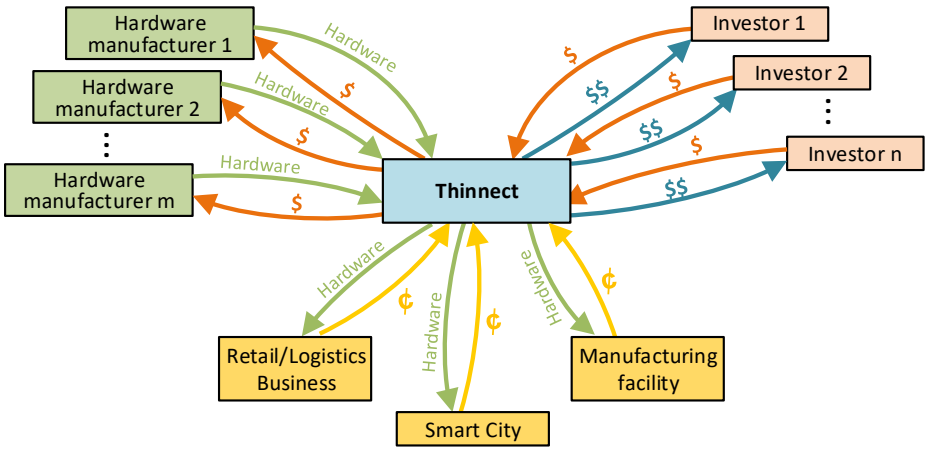

Fig. 6. Thinnect's innovative pay-per-measurement business model.

8 A Decision Maker Discussion

Permissionless ledgers are currently hyped due to their assumed, so-called disruptive potential. IT decision makers are looking into the technology to understand its potential for their own IT strategies. In concrete projects, this boils down to the question whether a distributed ledger should be considered as the required technology. System that may potentially benefit from distributed ledger technology can be, very generally, characterized as *data exchange systems*. In a data exchange scenario, we have several users who send messages to each other. Each user has a personal ledger, which serves as a basis for some important decisions. Decisions can be made by operators and users (both passive and active). For example, government agencies grant social benefits to citizens on the basis of digitally signed applications that have been received. Now, the entirety of all the individual ledgers can also be viewed as an aggregate ledger that reflects all information about data exchange between parties. Given a data exchange scenario, distributed ledger technology is only one possible solution among the following options:

- *distributed data exchange* (without a central database, peer-to-peer),
- *trusted centralized service*,
- *permissioned ledger*, or
- *permissionless ledger*.

Now, the question of whether a distributed ledger is required in a concrete data exchange scenario can be approached more systematically in terms of the above options and respective requirements that make them necessary. We have compiled this approach into a questionnaire-schema in Fig. 7. For the sake of comprehensiveness, the questionnaire-schema needs to introduce some necessary over-simplifications (the actual area of discourse is much more complex) as becomes clear in the subsequent discussion. Still, we are convinced that the questionnaire-schema is helpful in concrete decision scenarios.

The first question to be posed is whether all users can make their decisions solely on the basis of their personal ledgers. Occasionally, data from other parties is needed in decisions, but these data are requested from the other parties to become part of the individual ledger – as least as long as needed for a particular decision. The parties in this scenarios trust in the correctness and accuracy of each other's data without further measures, i.e., they are peers. Similarly, non-repudiation is not an issue in this scenario. In such a scenario, the system does not require the use of a trusted centralised service and distributed data exchange is sufficient. Centralized services are still necessary in such scenario to synchronize distributed data in long-running business transaction and coordinate cross-organizational business processes [39–41, 36]. However, such centralized services are just another kind of "trustworthy" party (peer) in that scenario, i.e., they are no trusted centralized services in the sense of this discussion.

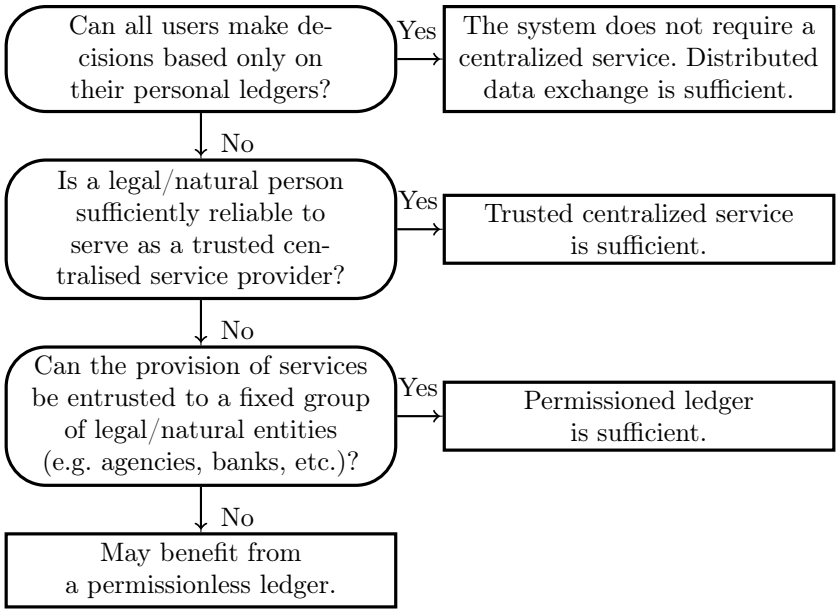

Fig. 7. Scheme of requirements for the potential use of distributed ledger technology.

If distributed data exchange is not sufficient as a solution, we need to ask whether a legal/natural person would be sufficiently reliable to serve as a trusted central service provider as the next question. Criteria for reliability depend on the context, the application, the legal dispute resolution framework etc. They also depend on a country's laws, and therefore, the applicable laws of the country in question. If the answer to this question is yes, then a trusted centralised service is sufficient as a solution. If a single legal/natural person is not capable to serve as a trusted central service provider due to scalability issues, it is a standard solution to build a hierarchy of trusted distributed services. As there is still a *root* trusted service, we count such hierarchy of trusted services still as a "single" centralized service for the sake of this discussion.

If no single legal/natural person can be sufficiently trusted, we need to ask whether centralized services can be entrusted to a fixed group of legal/natural persons (e.g. several public authorities, banks, etc.). In the context of a country, this question essentially aims at whether the country as such can be trusted. If the answer is yes, then a permissioned ledger is sufficient.

To conclude, a permissionless ledger may be required only to resolve tasks where neither a country nor any consortium of private individuals nor a consortium of public authorities or private parties are considered to be sufficiently reliable, or there is a request for protection against major powers that might affect any consortium.

9 Conclusion

We live in the postmodern age, which still might be characterized best through the lenses of the original analysis provided by Francois Lyotard [80, 81]: as a paralysis of the grand narratives by the emergence of a plethora of small, individualized mini narratives and sub-cultures. We feel very much, that the grand narratives and the mini narratives are not only in tension; also, they are in dialogue. It is not just so that the grand narratives are cannibalized by the new, emergent narratives [82]; rather, our social reality swifts between the several grand and the many mini narratives, steadily transforming and re-adjusting all of them and through each other. Furthermore, it seems as if we oscillate between more vibrant and less vibrant interaction around the many narratives over the decades. Where the 1990s have been roaring [83], we saw some slope of disillusion in the first decade after the millennium. Then, in the 2010s, we again saw, more

and more, a split of perceptions; mini narratives became micro narratives hand-in-hand with the emerging fake news debate. One of the narratives that received massive attention in the past decade – from industry, academia and the public – is the *blockchain revolution*.

In her speech at the Bank of England Conference in September 2017, Christine Lagrange said: "To be clear, this [virtual currencies] is not about digital payments in existing currencies—through Paypal and other »e-money« providers such as Alipay in China, or M-Pesa in Kenya. Virtual currencies are in a different category, because they provide their own unit of account and payment systems. These systems allow for peer-to-peer transactions without central clearinghouses, without central banks. For now, virtual currencies such as Bitcoin pose little or no challenge to the existing order of fiat currencies and central banks. Why? Because they are too volatile, too risky, too energy intensive, and because the underlying technologies are not yet scalable. Many are too opaque for regulators; and some have been hacked. But many of these are technological challenges that could be addressed over time. Not so long ago, some experts argued that personal computers would never be adopted, and that tablets would only be used as expensive coffee trays. So I think it may not be wise to dismiss virtual currencies." [84]

With this paper we wanted to contribute to a more informed discussion of the potential of blockchain technology. We wanted to look at blockchain technology from a neutral, analytical perspective. Our aim was to understand technological and socio-economic barriers to envisioned blockchain technology solutions that are intrinsic in the blockchain technology stack itself. We looked into the *permissionless blockchain* and the *permissioned blockchain*, where the *permissionless blockchain* gained our major interest – at least in this paper (we have designed successfull permissioned blockchain solutions with large customer bases ourselves, we have glimpsed over them shortly in this paper, in service of the big picture that we addressed). The permissionless blockchain came first, i.e., it is the blockchain *per se*. Permissioned blockchain solutions have a higher *capability-to-perform*, at the same time they have less fundamental disruptive potential; still their corresponding initiatives gain (from a marketing perspective) from the perceived immense (seemingly boundless) disruptiveness of the permissionless blockchain.

References

1. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008) https://bitcoin.org/bitcoin.pdf.
2. Buldas, A., Saarepera, M., Steiner, J., Draheim, D.: A Unifying Theory of Electronic Money and Payment Systems. TechRxiv. Preprint, 2021. (2021) https://doi.org/10.36227/techrxiv.14994558.v2.
3. Draheim, D.: Blockchains from an e-governance perspective: Potential and challenges. In: Proceedings of EGOSE'2020 – the 7th International Conference on Electronic Governance and Open Society – Challenges in Eurasia. Communications in Computer and Information Science 1349, Springer (2020)
4. Buldas, A., Draheim, D., Nagumo, T., Vedeshin, A.: Blockchain technology: Intrinsic technological and socio-economic barriers. In: Proceedings of FDSE'2020 – the 7th International Conference on Future Data and Security Engineering. LNCS 12466, Springer (2020) 3–27
5. Arne Ansper, Ahto Buldas, J.W.: Cryptographic Algorithms Lifecycle Report 2017. Technical Report Doc. A-101-9, AS Cybernetica (May 2018) Procurer: Information Systems Authority, Republic of Estonia.
6. Antonopoulos, A.M.: Mastering Bitcoin: Programming the Open Blockchain. O'Reilly (2017)
7. Szabo, N.: Smart Contracts: Building Blocks for Digital Markets. Nick Szabo (1996)
8. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)
9. Williamson, O.: Transaction cost economics: How it works; where it is headed. De Economist **146** (1998) 23–58
10. Koppenjan, J., Groenewegen, J.: Institutional design for complex technological systems. International Journal of Technology, Policy and Management **5**(3) (2005) 240–257
11. Malcolm Campbell-Verduyn (ed.): Bitcoin and Beyond: Cryptocurrencies, Blockchains and Global Governance. Routledge (2017)
12. Narayanan, A., Clark, J.: Bitcoin's academic pedigree. Communications of the ACM **60**(12) (2017) 36–45
13. Narayanan, A., Clark, J.: Bitcoin's academic pedigree. ACM Queue Magazine **15**(4) (2017) 1–30
14. Bayer, D., Haber, S., Stornetta, W.: Improving the efficiency and reliability of digital time-stamping. In Capocelli, R., De Santis, A., Vaccaro, U., eds.: Sequences II. Springer (1993) 329–334
15. Mendling, J., Weber, I., van der Aalst, W., et al.: Blockchains for business process management – challenges and opportunities. ACM Transactions on Management Information Systems **9**(1) (2018) 1–16

16. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, 2nd ed. Springer (2018)
17. Rosemann, M., vom Brocke, J.: The six core elements of business process management. In vom Brocke, J., Rosemann, M., eds.: Handbook on Business Process Management 1. Springer 105–122
18. Rikken, O., Janssen, M., Kwee, Z.: Governance challenges of blockchain and decentralized autonomous organizations. Information Polity **24**(4) (2019) 397–417
19. Janssen, M., Weerakkody, V., Ismagilova, E., Sivarajah, U., Irani, Z.: A framework for analysing blockchain technology adoption: Integrating institutional, market and technical factors. International Journal of Information Management **50** (2020) 302 – 309
20. Upadhyay, N.: Demystifying blockchain: A critical analysis of challenges, applications and opportunities. International Journal of Information Management **54** (2020) 1–26
21. Voshmgir, S.: Token Economy – How the Web3 reinvents the Internet, 2nd. ed. BlochainHub Berlin, Berlin (2020)
22. Hughes, E.: A Cypherpunk's Manifesto. In Schneier, B., Banisar, D., eds.: The Electronic Privacy Papers: Documents on the Battle for Privacy in the Age of Surveillance. John Wiley & Sons, USA (1997) 285–287
23. May, T.C.: The Crypto Anarchist Manifesto. In Ludlow, P., ed.: Crypto Anarchy, Cyberstates and Pirate Utopias. The MIT Press (2001) 61–64
24. DuPont, Q.: Experiments in algorithmic governance: A history and ethnography of "The DAO," a failed decentralized autonomous organization. In Campbell-Verduyn, M., ed.: Bitcoin and Beyond: Cryptocurrencies, Blockchains and Global Governance. Routledge (2017) 1–18
25. Hammer, M., Champy, J.: Reengineering the Corporation. HarperCollins (1993)
26. Draheim, D.: Business Process Technology – A Unified View on Business Processes, Workflows and Enterprise Applications. Springer (2010)
27. Draheim, D.: Smart business process management. In Fisher, L., ed.: 2011 BPM and Workflow Handbook, Digital Edition. Future Strategies, Workflow Management Coalition (2012) 207–223
28. Atkinson, C., Draheim, D., Geist, V.: Typed business process specification. In: Proceedings of EDOC'2010 – the 14th IEEE International Enterprise Computing Conference, IEEE (2010) 69–78
29. Schein, E.H.: Organizational Culture and Leadership. Wiley (2016)
30. Lips, S., Tsap, V., Pappel, I., Draheim, D.: Key factors in coping with large-scale security vulnerabilities in the eid field. In: Proc. of EGOVIS'2018 – the 7th Intl. Conf. on Electronic Government and the Information Systems Perspective. LNCS 11032, Springer (2018)
31. Lips, S., Pappel, I., Draheim, D.: Designing an effective long-term identity management strategy for a mature e-state. In: Proc. of EGOVIS'2019 – the 8th Intl. Conf. on Electronic Government and the Information Systems Perspective. LNCS 11709, Springer 2019
32. Pappel, I., Pappel, I., Tepandi, J., Draheim, D.: Systematic digital signing in Estonian e-government processes – influencing factors, technologies, change management. Transactions on Large-Scale Data- and Knowledge-Centered Systems **16** (2017) 31–51
33. Tsap, V., Pappel, I., Draheim, D.: Key success factors in introducing national e-Identification systems. In: Proc. of FDSE'2107 – the 4th Intl. Conf. on Future Data and Security Engineering. LNCS 10646, Springer (2017) 455–471
34. Bharosa, N., Lips, S., Draheim, D.: Making e-government work: Learning from the Netherlands and Estonia. In: Proc. of ePart 2020 – the 12th IFIP WG 8.5 Intl. Conf. on Electronic Participation. LNCS 12220, Springer (2020) 41–53
35. Pappel, I., Tsap, V., Pappel, I., Draheim, D.: Exploring e-services development in local governemnt authorities by means of electronic document management systems. In: Proc. of EGOSE'2018: the 5th Intl. Conf. on Electronic Governance and Open Society – Challenges in Eurasia. Communications in Computer and Information Science 947, Springer (2019)
36. Draheim, D., Koosapoeg, K., Lauk, M., Pappel, I., Pappel, I., Tepandi, J.: The design of the Estonian governmental document exchange classification framework. In: Proc. of EGOVIS'16 – the 5th Conference on Electronic Government and the Information Systems Perspective. LNCS 9831, Springer (2016) 33–47
37. Myers, M., Ankney, R., Malpani, A., Adams, C.: RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. Internet Engineering Task Force (IETF) (2013)
38. Emmelhainz, M.A.: EDI: A Total Management Guide. Van Nostrand Reinhold (1993)
39. Schulte, R., Natis, Y.: Service Oriented Architectures, Part 1. Technical Report ID Number SPA-401-068, Gartner Group (1996)
40. Schulte, R.: Service Oriented Architectures, Part 2. Technical Report ID Number SPA-401-069, Gartner Group (1996)

41. Draheim, D.: The service-oriented metaphor deciphered. Journal of Computing Science and Engineering **4**(4) (2010) 253–275
42. Kalja, A.: The first ten years of X-Road. In Kastehein, K., ed.: Estonian Information Society Yearbook 2011/2012. Ministry of Economic Affairs and Communications of Estonia (2012) 78—-80
43. Ansper, A.: E-State From a Data Security Perspective, Master Thesis. Tallinn University of Technology, Faculty of Systems Engineering, Department of Automation (2001)
44. Ansper, A., Buldas, A., Freudenthal, M., Willemson, J.: High-performance qualified digital signatures for X-Road. In: Proc. of NordSec 2013 – the 18th Nordic Conference on Secure IT Systems. LNCS 8208, Springer (2013) 123–138
45. Paide, K., Pappel, I., Vainsalu, H., Draheim, D.: On the systematic exploitation of the Estonian data exchange layer X-Road for strengthening public private partnerships. In: Proc. of ICEGOV'2018 – the 11th Intl. Conf. on Theory and Practice of Electronic Governance, ACM (2018)
46. McBride, K., Kütt, A., Ben Yahia, S., Draheim, D.: On positive feedback loops in digital government architecture. In: Proc. of MEDES'2019 – the 11th Intl. Conf. on Management of Digital EcoSystems, ACM (2019)
47. Saputro, R., Pappel, I., Vainsalu, H., Lips, S., Draheim, D.: Prerequisites for the adoption of the X-Road interoperability and data exchange framework: A comparative study. In: Proc. of ICEDEG 2020 – the 7th Intl. Conf. on eDemocracy & eGovernment, IEEE (2020)
48. Andy Oram (ed.): Peer to Peer: Harnessing the Power of Disruptive Technologies. O'Reilly (2001)
49. Tanenbaum, A.S., Wetherall, D.J.: Computer Networks, 5th Edition. Pearson (2012)
50. Martinson, P.: Estonia – the Digital Republic Secured by Blockchain. PricewaterhouseCoopers (2019)
51. Merkle, R.: Protocols for public key cryptosystems. In: Proc. of S&P'1980 – the 1st IEEE Symposium on Security and Privacy. (1980) 122–122
52. Buldas, A., Saarepera, M.: Document Verification with Distributed Calendar Infrastructure. US Patent Application Publication No.: US 2013/0276058 A1 (2013)
53. Buldas, A., FKroonmaa, A., Laanoja, R.: Keyless signatures' infrastructure: How to build global distributed hash-trees. In: Proc. of NordSec'2013 – the 18th Nordic Conference on Secure IT Systems. LNCS 8208, Springer (2013)
54. Mota, C.: The rise of personal fabrication. In: Proc. of C&C'11 – the 8th ACM Conference on Creativity and Cognition, ACM (2011) 279–288
55. Tao, F., Cheng, Y., Zhang, L., Nee, A.Y.: Advanced manufacturing systems: socialization characteristics and trends. Journal of Intelligent Manufacturing **28**(5) (2017) 1079–1094
56. Vedeshin, A., Dogru, J.M.U., Liiv, I., Draheim, D., Ben Yahia, S.: A digital ecosystem for personal manufacturing: An architecture for a cloud-based distributed manufacturing operating system. In: Proc. of MEDES'2019 – the 11th Intl. Conf. on Management of Digital EcoSystems, ACM (2019) 224–228
57. Vedeshin, A., Dogru, J.M., Liiv, I., Ben Yahia, S., Draheim, D.: A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method. IEEE Access **October** (2019)
58. Vedeshin, A., Liiv, I., Ben Yahia, S., Draheim, D.: Smart cyber-physical system for pattern recognition of illegal 3d designs in 3d printing. In: SADASC'2020 – the 3rd Intl. Conf. on Smart Applications and Data Analysis for Smart Cyber-Physical Systems. Communications in Computer and Information Science 1207, Springer (2020) 74–85
59. Brian Heater: Zap brings the manufacturing process to the cloud. https://techcrunch.com/2016/09/12/zap/ (2016) [Online; accessed 04-September-2020].
60. Isbjörnssund, K., Vedeshin, A.: Secure Streaming Method in a Numerically Controlled Manufacturing System, and a Secure Numerically Controlled Manufacturing System. U.S. Patent 2014111587 A3, Dec. 3, 2015.
61. 3D Control Systems, Inc.: 3DPrinterOS cloud world statistics. https://cloud.3dprinteros.com/dashboard/#/world-statistics (2020) [Online; accessed 04-September-2020].
62. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proc. of EuroSys'2018 – the 13th European Conference on Computer Systems, ACM (2018) 1–15
63. Li, A., Yang, X., Kandula, S., Zhang, M.: CloudCmp: comparing public cloud providers. In: Proc. of IMC'10 – the 10th ACM SIGCOMM Conference on Internet Measurement, ACM (2010) 1–14
64. Housley, R., Polk, W., Ford, W., Solo, D.: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, RFC 3280, April 2002 (2002)
65. FinancesOnline: 47 Essential e-Signature Statistics: 2020 Market Share Analysis & Data. https://financesonline.com/25-essential-e-signature-statistics-analysis-of-trends-data-and-market-share/ (2020) [Online; accessed 05-September-2020].

66. Everis: Study on the Use of Electronic Identification (eID) for the European Citizens' Initiative. Final Assessment Report. Everis, European Commission (2017)
67. Tsap, V., Pappel, I., Draheim, D.: Factors affecting e-ID public acceptance: a literature review. In: Proc. of EGOVIS'2018 - the 8th Intl. Conf. on Electronic Government and the Information Systems Perspective. LNCS 11709, Springer (2019)
68. Lips, S., Barhosa, N., Draheim, D.: eIDAS implementation challenges: the case of Estonia and the Netherlands. In: Proc. of EGOSE'2020: the 7th Intl. Conf. on Electronic Governance and Open Society – Challenges in Eurasia. Communications in Computer and Information Science 947, Springer (2019)
69. Tsap, V., Lips, S., Draheim, D.: Analyzing eID public acceptance and user preferences for current authentication options in Estonia. In: Proc. of EGOVIS'2020 – the 9th Intl. Conf. on Electronic Government and the Information Systems Perspective. LNCS, Springer (2020)
70. Cavallini, S., Bisogni, F., Gallozzi, D., Cozza, C., Aglietti, C.: Study on the supply-side of EU e-signature market: Final Study Report. Fondazione FORMIT, European Commission, Directorate-General Information Society and Media of the European Commission (2012)
71. Alzahrani, L., Al-Karaghouli, W., Weerakkody, V.: Analysing the critical factors influencing trust in e-government adoption from citizens' perspective: A systematic review and a conceptual framework. International Business Review **26**(1) (2017) 164–175
72. Carter, L., Liu, D.: Technology humanness, trust and e-government adoption. In: Proc. of ACIS 2018 – the 18th Australasian Conference on Information Systems, Association for Information Systems (2018)
73. Agrello: Agrello ID - Identity based digital signatures. https://www.agrello.id (2020) [Online; accessed 05-September-2020].
74. Binu, V., Sreekumar, A.: Threshold multi secret sharing using elliptic curve and pairing. arXiv preprint arXiv:1603.09524 (2016)
75. Liu, Y., Liu, T.: A novel threshold signature scheme based on elliptic curve with designated verifier. In: Proc. of ICAIS'2019 – the 5th Intl. Conf. on Artificial Intelligence and Security. LNCS 11635, Springer (2019) 332–342
76. The EU Parliament and the Council of the Eurpean Comission: Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. Official Journal of the European Union **L 257/73** (2014)
77. Gavin Wood et al.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper 151 (2014)
78. Veriff: Veriff - identity verification done in seconds. https://www.veriff.com (2020) [Online; accessed 05-September-2020].
79. Thinnect: Thinnect – IoT Edge of Network Service Provider. https://www.thinnect.com (2020) [Online; accessed 05-September-2020].
80. Lyotard, J.F.: The Postmodern Condition: A Report on Knowledge. Éditions de Minuit (1979)
81. Lyotard, J.F.: La condition postmoderne: rapport sur le savoir. Manchester University Press (1979)
82. AERO Magazin: How French "Intellectuals" Ruined the West: Postmodernism and Its Impact, Explained. AERO Magazin, March 27, 2017
83. Stiglitz, J.E.: The Roaring Nineties: A New History of the World's Most Prosperous Decade. W. W. Norton (2004)
84. Christine Lagarde: Central Banking and Fintech—A Brave New World? Bank of England Conference, London, 29 September 2017.

Ahto Buldas is professor of cryptography at Tallinn University of Technology. Studied computer science at Tallinn University of Technology (1985-1991). MSc on simulation techniques for Boolean circuits (1992). PhD on computational algebraic graph theory (1999). His research interests are related to applied cryptography. Time-stamping related research started from 1997, during which he has published papers in Crypto, Asiacrypt and PKC conferences. Participated in the development of the Estonian Digital Signature Act and ID-card (1996-2002). His current research interests also include risk analysis methods, including attack-tree semantics and game-theoretical approaches to risk analysis. Ahto Buldas is a co-founder of Guardtime and also of Cybernetica AS.

Dirk Draheim is full professor of information systems and head of the Information System Group at Tallinn University of Technology (TTÜ). Dirk holds a Diploma in computer science from Technische Universität Berlin, a PhD from Freie Universität Berlin and a habilitation from the University of Mannheim. From 2006-2008 he was area manager for database systems at the Software Competence Center Hagenberg, Linz. From 2008-2016 he was head of the data center of the University of Innsbruck and, in parallel, Adjunct Reader at the Faculty of Information Systems of the University of Mannheim. Dirk is co-author of the Springer book "Form-Oriented Analysis" and author of the Springer

books “Business Process Technology”, “Semantics of the Probabilistic Typed Lambda Calculus” and “Generalized Jeffrey Conditionalization”. His research interest is the design and implementation of large-scale information systems.

Takehiko Nagumo is Senior Managing Executive Officer of Mitsubishi UFJ Research and Consulting. Also, Takehiko is adjunct professor in strategic management at Kyoto University Graduate School of Management. Takehiko holds an MSc in Development Finance from University of London and an MBA in Strategic Management from Georgetown University. Takehiko is the first BSC Hall of Fame winner from Japan, selected by Professor Kaplan of Harvard Business School. Numerous times, Takehiko’s research papers and case studies on Balanced Scorecard have been published by Harvard Business School. Takehiko is one of the thought leaders in Japan in the field of digital economy and society and is active in designing human-centric AI implementation in smart cities and governments in Japan and globally: he is a member of the Japanese Government’s Regulatory Reform Promotion Committee, invited research fellow at the National Institute of Advanced Industrial Science and Technology (AI Center), co-founder and Executive Director of the Smart City Institute Japan and Fellow of the World Economic Forum, Centre for Fourth Industrial Revolution Japan.

Anton Vedeshin conducts a Ph.D. in cloud computing and cybersecurity and received an M.Sc. degree in computer science from Tallinn University of Technology (TTÜ). He has developed software since the age of 11. Author of numerous cybersecurity and digital manufacturing scientific papers (IEEE, ACM, Springer) and patents. Anton conducted reviews of numerous articles on cloud computing for IEEE Access. He teaches cloud computing at TTÜ and data security at the Estonian Entrepreneurship University of Applied Sciences (EUAS). Anton is the CTO and co-founder of 3D Control Systems, Inc., works in the advanced manufacturing industry and deploys secured automated manufacturing solutions for F500 enterprises. He founded his first software development company at the age of 20 with 25+ employees. Anton developed multiple security and e-government solutions for the Estonian government and the EU Commission and worked in the Mitsubishi electric car data mining project. Anton designed numerous blockchain solutions based on several platforms (Ethereum, Hyperledger Fabric, Corda R3). His research interests include cloud computing, cybersecurity, 3D printing, blockchain, machine learning and AI.