On-chain vs. off-chain storage for supply- and blockchain integration

Article *in* it - Information Technology · November 2018

DOI: 10.1515/itit-2018-0019

5 authors, including:

Thomas Hepp
Universität Konstanz
11 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Philip Ehret
Universität Konstanz
4 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Alexander Schoenhals
University of Konstanz
7 PUBLICATIONS **35** CITATIONS
SEE PROFILE

Bela Gipp
Bergische Universität Wuppertal
138 PUBLICATIONS **2,266** CITATIONS
SEE PROFILE

Thomas Hepp*, Matthew Sharinghousen, Philip Ehret, Alexander Schoenhals, and Bela Gipp

On-chain vs. off-chain storage for supply- and blockchain integration

https://doi.org/10.1515/itit-2018-0019
Received July 14, 2018; accepted November 12, 2018

Abstract: Supply chains are the basis of most everyday life products. Both data integrity and authenticity of related information have severe implications for quality and safety of end-products. Hence, tamper-proof storage is necessary that prevents unauthorized modifications. We examine peer-reviewed blockchain technologies according to four criteria relevant to supply chains: On-chain storage, off-chain storage, verification cost and secure data sharing. Our evaluation yields an overview of concepts for modeling supply chain processes and points out that on-chain storage is currently not practical.

Keywords: decentralized storage system, blockchain, supply chain

ACM CCS: Information systems → Information storage systems

1 Introduction

Everything from the clothes on our backs to the food we eat is the end result of a supply chain. Even something as simple as a toothbrush requires the coordination of materials, intermediate processes, and distribution before it can be purchased by the consumer in the grocery store. More complex products such as computers or medications rely on similar, if not more complex, supply chains. Figure 1 shows a high level example of a supply chain. The physical product is modified or transported at each step and eventually the chain terminates. The information pertaining to the chain, however, is bi-directional and exists more statically. It is this information that needs to be verified and stored immutably.

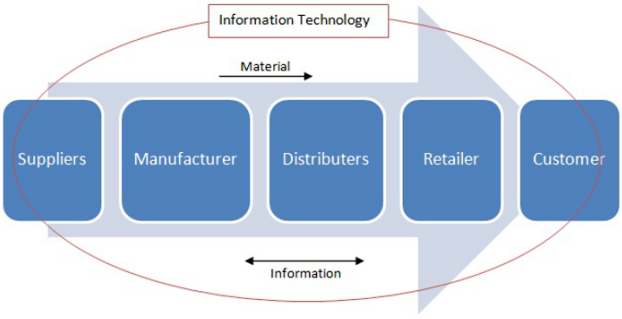

Figure 1: A high level supply chain from a retail environment showing the relationship between the physical product and the information for each link in the chain. Each link can be further abstracted into its own supply chain [20].

When considering a long and complex supply chain, how can we be sure of what was performed during each intermediate step? First approaches are already introduced for tracing products in supply chain cycles, such as [13, 15, 23, 14, 28] Further, if a problem is identified with the final product, how can we effectively determine which link in the chain produced the problem? Finally, once we have answers to these questions, how can we be sure that the information is correct? To answer this question, the following criteria need to be fulfilled: All parties in the chain can trust the information, there is no opportunity to make unauthorized modifications, and the information is available to all authorized parties. This essentially describes a blockchain.

There are several problems, however, with using a blockchain to maintain the integrity of and store supply chain information. First, a blockchain technology may have a limit to the amount of data that can be stored in a single block. Second, the cost of committing a transaction to a block may be prohibitively expensive. Third, it may be necessary to share information for computation or verification, but the data itself is confidential or should not be readable by all parties in the chain.

The remainder of this paper examines a selection of available technologies in order to gain an overview of which features are currently available, and discuss how these features can be applied to supply chains to answer the open questions from above. Section 2 outlines how the

***Corresponding author: Thomas Hepp,** University of Konstanz, Information Science Group, D-78464 Konstanz, Germany, e-mail: thomas.hepp@uni-konstanz.de, ORCID: http://orcid.org/0000-0002-7671-0602
Matthew Sharinghousen, University of Konstanz, Distributed Systems Laboratory, D-78464 Konstanz, Germany, e-mail: matthew.sharinghousen@uni-konstanz.de
Philip Ehret, Alexander Schoenhals, Bela Gipp, University of Konstanz, Information Science Group, D-78464 Konstanz, Germany, e-mails: philip.ehret@uni-konstanz.de, alexander.schoenhals@uni-konstanz.de, bela.gipp@uni-konstanz.de

examined technologies were chosen and discusses criticism each is subject to. Section 3 presents the findings of the investigation. The findings are broken down into four sections ranging from storage methodology to information sharing in order to relate back to the questions listed above. Section 4 considers the high level application of blockchains as a method for guaranteeing data integrity in supply chains. Finally, Section 5 presents some derivative questions and future works.

2 Methodology

Ideally, the comparison of all varieties of blockchains in respect of their feature sets, advantages, and disadvantages is desirable. This is, however, not practical, as around 1000 known crypto-currencies [5] and additionally, more blockchains without an associated currency exist. Therefore, this paper focuses primarily on the blockchains associated with the most valuable crypto-currencies and a subset of storage-based blockchain products. "Most valuable" denotes such currencies with the highest market capitalization as listed by coinmarketcap.com [5]. The subset of products is based on Google searches with the keywords "blockchain", "storage", and "distributed" in various combinations and with various filler words. Requiring the availability of an academic paper on the product further reduces the set.

Four important feature considerations relevant to supply chains were determined [29].

1. *On-chain Storage* – the ability to store various amounts of information in the blockchain itself. This is an important consideration, because the stored information will be necessarily available to all parties in the system.
2. *Off-chain Storage* – The storage of information in various forms off of the blockchain. This becomes necessary when a party wants to verify information with the blockchain, but does not necessarily want to make the information available. Additionally, the information stored may be larger than the blockchain itself supports.
3. *Cost of Verification* – the computational, temporal, and monetary cost of verifying a new block [16].
4. *Secure Data Sharing* – The ability to make data available for computation, without making the plain-text data available for copy and potential redistribution, such as [16, 32].

For each feature, the selected blockchain technologies are compared in a corresponding section in the next chapter. Three main criteria are used to decide which papers are considered: popularity, peer-reviewed, and availability. For features with a limited pool of candidate papers, blockchains with peer-reviewed papers are given priority. The findings are presented in the form of a matrix giving an overview of the considered features and blockchains.

3 Results

The two main blockchains considered, based on value and popularity, were Bitcoin [19] and Ethereum [4]. Bitcoin was an obvious choice for this paper, as it is leading crypto-currency and paved the way for other varieties of blockchains to be developed based on its features or, in some cases, lack thereof. Ethereum is a suitable counterpart to Bitcoin, because of its equally large market capitalization and popularity. Given its more feature-full design, Ethereum also has applications outside of on-chain storage. Regarding off-chain storage, less well-known blockchain technologies such as BigChainDB [18] and Hawk [16] were used as examples of alternative methods of storage and secure data sharing. In addition, Enigma [32] and BlockDS [7] show how data can be shared via a blockchain with permissions.

3.1 On-chain storage

The simplest way of storing information with a blockchain is to simply store the data in the chain itself. For example, binary data can be stored as part of a transaction and will then be distributed to the community along with all the other transactions. This leads to the question how much information can be stored on-chain within transactions and what to do with a piece of data, which equals or exceeds a given size limit.

Fixed block size. As of the submission of this paper, Bitcoin is the most valuable crypto-currency [5], and is an example of a fixed-block-size blockchain. In particular, a lot of attention has been paid to the semi-fixed block-size of Bitcoin, which is 1 megabyte [3]. The size is semi- instead of hard-fixed, because the actual size of the block can be anywhere less than 1 megabyte as well as slightly larger. Part of the controversy comes from the limited number of transactions which can be stored in a single block. This, combined with the current popularity of Bitcoin, forces users to pay high transaction fees. These fees incentivize miners to include the transaction in their block [1]. These costs and their effects will be discussed in more detail later in this paper. For on-chain storage, the effect of the fixed block size

is clear: even a moderately sized file of 5 megabytes, which is small, is impractical and expensive to store. An alternative are increased block sizes. This allows for larger pieces of data to be stored, while mitigating the negative effect of taking that space away from regular transactions. There are a number of side-effects to increasing block size, which need to be taken into account. On the one hand, increasing the size would increase the number of transactions processed per second. At present, the throughput of Bitcoin caps out at around 7 transactions per second (tps) [6]. This pales in comparison to the average 2000 tps of Visa [22]. On the other hand, there are many arguments against increasing the size [3]. First, a full-node would need to spend more on storage and computation resources in order to handle the increased size. [9] A theorized consequence of this is that the number of full-nodes would decrease which would lead to a more centralized system. Second, changing the block size requires a hard-fork of the chain. This means that current tools would need to be updated in order to support the new system. If too few organizations agree on the new fork, then the new version could fail, or the two versions could continue side by side. Both cases could negatively impact the usability of the blockchain.

Clearly, while expanding these considerations for more general application, using a fixed size blockchain for on-chain storage is not practical. Either the blocks are small, decentralized, and expensive for storage or they are large, centralized, and expensive to verify.

Variable block size. In contrast to Bitcoin with its fixed block size, Ethereum lends itself as a well-known example of variable block sizes in a blockchain [26].

The given variability of the block size makes Ethereum much more attractive as an option. Especially in the case of very large files, the block size could be increased sufficiently to store the entire file in one block. The advantage to having the file in a single block, is that only the single block needs to be stored in a lightweight-node for access to the data. Despite the larger block size, the hash necessary for validation in the next block will still be 32 bytes, thus having no negative impact on the rest of the chain [26].

There are three critical problems with storing files in Ethereum, however, two of which revolving around cost. First, the cost of a transaction/contract increases proportional to the size thereof. The technical definition of Ethereum defines the fee for a 256 bit word at 20,000 gas (tx currency), meaning 1 megabyte of data requires a fee of 625 million gas [26]. Based on prices in early January, 2018, this transaction would cost thousands of dollars. Second, assuming someone is willing to pay for the transaction, a miner needs to be willing to process the block. Finally, while there is theoretically no limit to the size of an Ethereum block, the block size is based on the demand of the system and votes cast by miners. The size of a block is tied to the gas limit on a block, rather than vise-versa. If miners choose to increase the gas limit, then the block size increases as a result. In order to store large files in a single block, the miners must be willing to allow such a block size. As shown in Figure 2, the block sizes in 2016 grew proportionally to the amount of transactions, while only growing large enough to store moderately sized pieces of data. Despite this growth, the size of a single block has never exceeded 35 kilobytes – this is significantly smaller than the 1 megabyte Bitcoin block size.

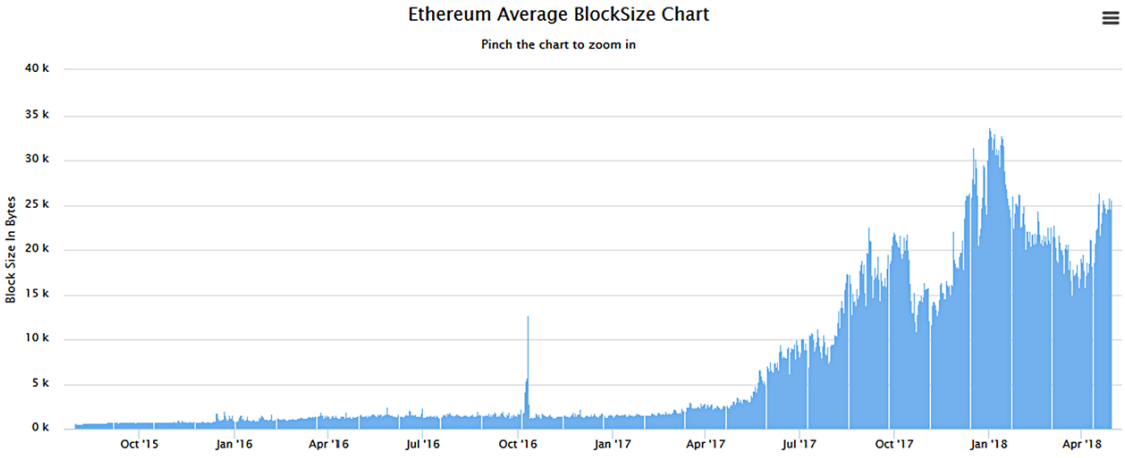

Figure 2: The average block size of Ethereum through 2016. The size of blocks is adjusted to maintain a roughly constant number of pending transactions, regardless of total transactions [8].

3.2 Off-chain storage

While on-chain storage is neither financially nor technically practical, the advantages that the blockchain provide can be applied to off-chain storage methods. As a general example, the hash of a piece of data, which is quite small, can be stored in the chain. Due to the relatively small size of a hash, the corresponding cost for storage will also be low. The challenge, then, is to provide a link between the hash in the chain and the physical storage location. To that end, the two methods considered are smart contracts and distributed hash tables (DHT), such as [31]. The two are differentiated, because a smart contract defines what, who, and how, while a DHT more specifically handles how data is distributed and accessed [30].

Smart Contracts. Storing the hash of a piece of data in a blockchain has the important benefit of immutably verified integrity of the data. An example of this application is Trusted Time-stamping described by Gipp et al. [12, 11, 10]. However, storing the hash is only part of the necessary information. In order to access the data, one also needs to know who has a copy of it. Such information is stored in a smart contract. The contract, like a real-world contract, defines the specific terms of the agreement including, but not limited to, when something should happen, what should happen, and how much the operation is worth (the cost). Smart contracts, in opposition to the real world, are programmed and irreversible. This characteristic is a side-effect of the designed immutability of a blockchain. As a simple example, a smart contract can be made with the following conditions: User A stores 20GB worth of data with User B. User B agreed to store the data for 6 months at a fee of $5. Once the fee has been paid, User A transfers the data to User B, where it is stored for the agreed amount of time. Most notably, this contract does not specify any guaranteed access to the stored data. If User B is offline, then User A loses access to their data, while the data is still technically being stored and fulfilling the contract. In such storage contract situations, it is typical that one must not only pay for storage of their data, but also pay for access and download of the data [24, 25], thus giving the storage provider an incentive to be online. Another issue is the event that something unforeseen happens to User B's storage location. In this case, the data is also lost, with potentially no possibility of recovery.

Distributed Hashtables. There are several possibilities to overcome the shortcomings of using smart contracts for simple peer-2-peer storage. For example, multiple copies of a file could be stored in a storage network, so that if one storage location becomes unavailable, there is still at least one copy of the data available. This solves both the problem of a storage location temporarily being offline, and removes a location as a single-point-of-failure for the data. This leads to the topic of distributed hashtables (DHT). In general, a DHT is a network of storage locations with a centralized index. The index stores the information which tells which piece of data is stored in which storage location. The storage locations can either store the data entirely, or, like a RAID system, store a piece of the distributed data. In the context of blockchains, the index is stored decentralized on the chain. The technologies Storj, Sia, and BlockDS were considered to give a brief overview of currently available DHT solutions. Storj and Sia are two examples of commercial products, where a storage location is made available, and capacity is purchased by consumers [24, 25]. Both technologies function on the principle of a DHT, where a piece of data is split and distributed to multiple locations.

BlockDS. Both of the above technologies apply security through encryption of the stored data. The consumer holds the decryption key, so no matter where the data is stored, it is secure. However, neither system handles the case of sharing the data with others. In order to do so, one would need to pull a local copy of the data, send it to another user, and the user would need to re-upload the data into the network for storage. To combat this use-case there exists BlockDS. A piece of data is stored across the DHT with primary access rights for the owner of the data. In contrast with Storj and Sia, however, the owner can assign additional access rights to the encrypted data to additional consumers. The data is accessed by the consumers via a static set of keywords. This set is necessary for the search, due to the data itself being encrypted and unreadable [7]. Once the data has been accessed, however, the consumer could potentially download a local copy of the data and redistribute it at all, meaning the original owner of the data no longer controls it.

Distributed Databases. Similar to DHTs are distributed databases (DDB). The difference with a DDB, though, is the addition of a query language on top of the storage backend. EthDrive and BigchainDB [18] are two examples of DDBs integrated with blockchains [18, 27]. In general, a DDB functions just like a classic database. In order to solve problems of scalability, additional nodes can be added to a DDB in order to further increase storage and processing capacity. A publisher starts a contract and pushes data into the network. The DHT information is stored in the blockchain, and the data itself is distributed throughout the DDB (P2P file system network). In order to access the data, a publisher or consumer first issues a query against the DDB, and once the data is found can

download a copy. BigchainDB works in more or less the same way, but with more focus on commercial-like applications [18].

3.3 Cost of verification

In the most simple case, there are inherent costs related to submitting a transaction as well as verifying a block. There are additional associated costs, however, which are necessary but not always immediately clear. For a brief overview, the financial and computational costs are most relevant to supply chains.

Financial Cost. When comparing on- and off-chain storage, given currently available technologies, the financial cost per transaction is a pressing issue. If an individual product is to be tracked at 10 stages from start to finish, multiplied by 10.000 units, then there is already demand for 100.000 transactions. Assuming one transaction per stage per unit is, certainly, a naive approach, but highlights the necessity for very low single-transaction costs. In practice, considering a batch per stage and transaction is more likely, such as OriginStamp [12].

Computational Cost. In addition to the financial costs of a transaction, block verification incurs computational costs. First, storing data on the blockchain and increasing block sizes have consequences. Off-chain storage also has costs, especially when considering the potentially huge number of transactions per second in an industrial setting. For simplicity, Bitcoin will be used as an example for on-chain storage and the associated computation. A full node contains all of the blocks in the chain, whereas a partial node only contains the blocks in which the owner is interested [1]. In a fixed block size situation, the hardware requirements of a full node are related to this size. Bitcoin's 1 megabyte blocksize is a good example [19]. If the blocksize is increased to 5 megabytes to facilitate data storage, that has a consequence of multiplying hardware requirements for full nodes by 5 times. An area requiring further research is the effect of this increase cost on the number of full nodes. A high number of full nodes is necessary in order to maintain the decentralized nature of blockchains. In the case of off-chain storage, the block size can remain small, so the storage impact on full nodes remains as it is. However, in an industrial setting, as mentioned, the number of transactions to be committed can be very large. This means that in order to maintain a low wait time for verification, the number of transactions verified per second needs to increase proportionally. Modifying complexity to adjust verification time has been considered since the release of Bitcoin's original paper [19].

3.4 Secure data sharing

Outside of information technology, securely data sharing is often implemented with Non-Disclosure Agreements (NDA). This means that, while the owner of the information may seek reparation or punishment for a leak, the information was leaked none-the-less, and is no longer in control of the owner. In the context of IT there exists a technical means to ensure ownership – encryption. This presents the challenge, however, of sharing encrypted information between parties, without compromising ownership. Two potential solutions to this problem, with blockchain integration, are Enigma [32] and Hawk [16].

Enigma. The Enigma system is a peer-to-peer network, which allows multiple parties to share data and code securely [33]. Additionally, the system allows for the distributed computation of complex code. The technical specifics by which the computation and information sharing is implemented are beyond the scope of this paper. However, Figure 3 shows how Enigma proposes handling computation. First, a publicly available link to the storage location is stored on the blockchain. Second, the data itself, stored encrypted, is split into many chunks. These chunks of data are distributed throughout the Enigma node network, with one chunk per node. The nodes are isolated from each other in such a way that no information about a chunk is shared with any other nodes. Finally, computation is performed on the data, with the results from all the nodes being sent to the owner. In this way, the owner is the only one who has the complete input and result datasets. Since the data is encrypted, theft is difficult. That leaves denial of service as the main avenue of

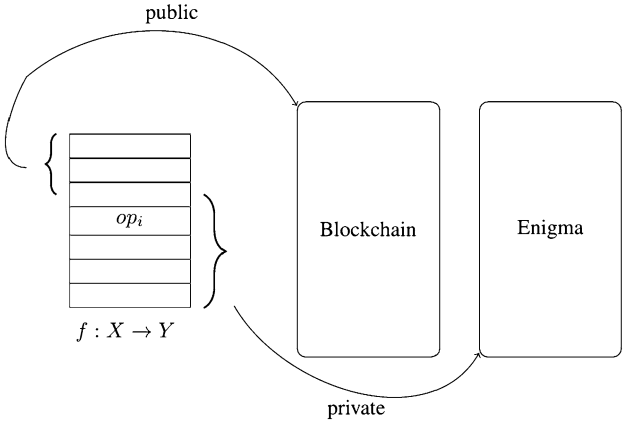

Figure 3: A simple representation of how information is split between the public blockchain and private storage [32].

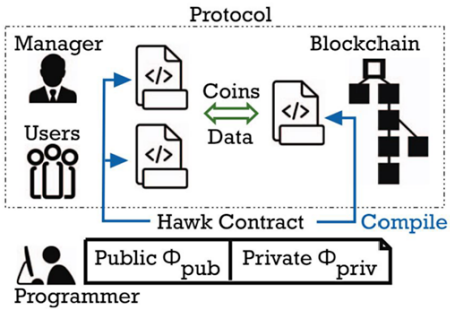

Figure 4: Hawk allows for contracts to be submitted, which are facilitated by the manager [16].

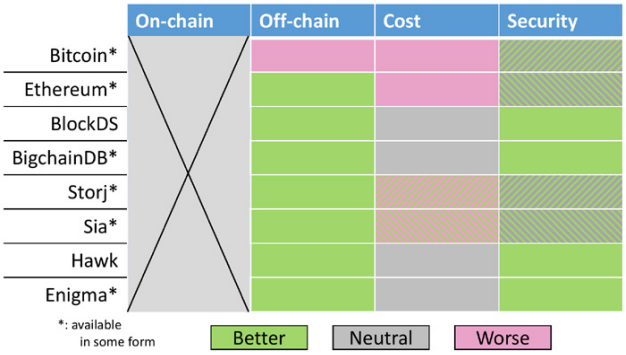

Figure 5: A classification of the discussed technologies. The ratings are based both on technical and subjective conclusions.

attack. To reduce the effect of this attack, Enigma requires that a deposit, in addition to a computation fee, be placed on transactions. Due to a deposit which is placed on transactions, in addition to a computation fee, a DDoS attack would be very expensive to conduct.

Hawk. In contrast to Enigma, Hawk does not focus on the secure computation of data but rather the security of transaction information [16]. In a classic blockchain, smart contracts are stored on the chain, and the terms of the contract are publicly available. Hawk, on the other hand, offers a way for smart contracts to be handled in a confidential manner, while retaining the benefits of the immutable blockchain. First, the programmer defines the terms of the smart contract. As described in the Hawk paper, a contract can be a bid in an auction. The private portion of the contract contains the actual value of the bid, as well as any other private information. The public portion is stored on the public blockchain. The contract is then compiled into three separate programs: one for the consensus network, one for the manager, and one for the user. Most importantly, the consensus program is executed by all nodes in the network, and the result verified. As for attack vectors, the manager serves as a minimally trusted party – as described by the authors. The manager helps to facilitate the execution of the contract, and, as such, has access to the private information therein. While the manager cannot affect the execution parameters of the contract, he/she can access the private information and share it.

4 Discussion

To summarize, none of the discussed technologies are sufficient in order to fully integrate supply chains and blockchains. Figure 5 visualizes these findings. The classification into "better", "neutral", and "worse" is based on both technical features of the technologies as well as a subjective interpretation of their application in a supply chain environment. The technologies highlighted with a star are currently available either as private, commercial, or proof-of-concept products.

On-chain Storage. For fixed-block-size chains, on-chain storage is not practical [17]. Either the size of data needs to be severely limited, or the data needs to be split across multiple blocks. Regarding splitting, the number of blocks in the chain would increase very quickly – even with a relatively small user base. In a commercial setting, the user base could remain small, but the increased amount of data to be stored would offset the difference. In a consumer case, such as with Bitcoin, the increased block count would put additional strain on full-nodes [1]. This could potentially lead to a reduction of full-nodes and centralization of the blockchain. Variable-block-size is not capable of mitigating the mentioned problems. Larger pieces of data could be stored in a single block, but this still puts additional strain on full-nodes. Primarily, the different block sizes would require additional hardware and bandwidth to support them. In both consumer and commercial settings, extensive use of the blockchain for storage purposes could produce a very large number of blocks. As indicated in Figure 5, none of the technologies discussed in this paper solve these problems.

Off-chain Storage. Off-chain storage removes the necessity to consider variable block sizes and only requires a small amount of data to be stored on the chain itself. This has the advantage that small transactions cost less and thus a larger number of transactions fit in a block of particular size. The small amount of data required to be stored on the chain helps to solve two problems: As small transactions cost less, and a larger number of transactions fit in a block of a particular size. Almost all of the technologies support off-chain storage. Almost all of the technologies support off-chain storage, and are either chains

themselves, or protocols sitting on top of chains. Regarding protocols, one simply needs to choose an underlying chain with smart-contract support to retain off-chain storage capability. Bitcoin currently does not support smart-contracts and, therefore, makes it a bad candidate for this type of storage.

Cost of Verification. Bitcoin and Ethereum currently are too cost-intensive to be practical for high-frequency use [5]. The neutral technologies, protocols applied to separate blockchains, are difficult to classify, because most of the computational and financial costs are associated with the underlying chain. The selection of an inexpensive chain from the hundreds of available options, allows to keep costs low. In the case of BigchainDB, where a higher number of nodes correlates to a higher rate of transactions per second, the cost of recruiting new nodes needs to be considered. Storj and Sia are difficult to classify, due to their somewhat low financial costs and somewhat high relative costs. Relative to traditional storage methods, such as on a local hard drive, both Storj and Sia are roughly 50 % more expensive [24, 25].

Secure Data Sharing. All of the technologies support, at the very least, encryption to secure data. However, simply encrypting data introduces the problem of sharing that data with others. In order to share the secured data, a more complex solution is necessary instead of the naive approach of sharing the key – the more people that possess a single key, the less secure it becomes. Enigma and Hawk both introduce relatively secure data sharing methodologies. Enigma is primarily susceptible to compromised nodes. If a sufficient number of nodes collaborate, then the input dataset, as well as the results, can be compiled in an unencrypted state. Hawk is a more high level security application, which uses an interesting method of secret sharing. Its methodology of keeping contract details secret would be significantly more robust if it did not require a manager to facilitate the execution. The authors admit that the manager has access to the plaintext contract details, which Hawk is supposed to protect. This defeats the purpose of the protocol. Perhaps a combination of Hawk and Enigma would be a solution to the above mentioned problems. Enigma could be used as the manager, which as a protocol does not access the data, and Hawk used to keep the contract details secret.

5 Future works

The work done in this paper is the first of three conceived steps in a comprehensive research project. First, the state of the art needs to be researched, so that the best features of available technologies can be evaluated and, potentially, improved upon. Second, in order to tailor the project to domain needs, input from domain experts is necessary. Finally, the state of the art and domain feedback can be combined into a prototype solution.

Supply chains, and the information therein, can vary from industry to industry. The consequence is that the requirements of a supply chain and blockchain system will also vary. Ideally, a long-term project would involve a specific industry partner. A comprehensive design study with this partner, regarding both system architecture and interaction design, would ensure a high chance of industry adoption for the developed product. After performing a design study, the process of end-user validation can begin. We propose that, once a prototype has been implemented, the system should be integrated in to a test set of products. Both the functionality and efficiency of the system can be evaluated over a period of time. Such a process of development, including this stage of testing, would need to be conducted over several months, if not years.

6 Conclusion

Blockchains have been developed extensively since the Bitcoin white paper in 2008 [19]. However, there is still plenty of work to be done to allow full supply chain integration [15]. The technologies discussed in this paper individually resolve some of the integration issues, but none of them is capable of solving all of them on their own. Further development in the areas of cost and incentive, data security, and intellectual property preservation is necessary before an industry viable solution can be produced. Industry feedback and an iterative design process can assist further development in these areas.

This manuscript has been timestamped on the blockchain and is verifiable under: https://www.originstamp.org/u/2612b95d-6d79-f591-3fd4-30329392d926

References

1. A. M. Antonopoulos. *Andreas M. Antonopoulos – Mastering Bitcoin_ Programming the Open Blockchain*, O'Reilly Media (2017).
2. J. Benet and J. Ai. IPFS – Content Addressed, Versioned, P2P File System (DRAFT 3). In: Draft 3 ().
3. BitcoinWiki. *Block Size*. URL: https://en.bitcoin.it/wiki/Block_size_limit_controversy, 2018.

4. V. Buterin et al. A next-generation smart contract and decentralized application platform. In: *white paper* (2014).
5. CoinMarketCap. *Cryptocurrency Market Capitalizations*. URL: https://coinmarketcap.com/all/views/all/, 2018.
6. K. Croman et al. On scaling decentralized blockchains (A position paper). In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9604 LNCS (2016), pp. 106–125.
7. H. G. Do and W. K. Ng. Blockchain-Based System for Secure Data Storage with Private Keyword Search. In: *Proceedings – 2017 IEEE 13th World Congress on Services, SERVICES 2017* (2017), pp. 90–93.
8. Etherscan.io. *Blocksize*. URL: https://etherscan.io/chart/blocksize, 2018.
9. I. Eyal et al. Bitcoin-NG: A Scalable Blockchain Protocol. In: (2015). URL: http://arxiv.org/abs/1510.02037.
10. B. Gipp, K. Jagrut and C. Breitinger. Securing Video Integrity Using Decentralized Trusted Timestamping on the Blockchain. In: *Proceedings of the 10th Mediterranean Conference on Information Systems (MCIS)*, September (2016), pp. 1–10.
11. B. Gipp, N. Meuschke and C. Breitinger. Using the Blockchain of Cryptocurrencies for Timestamping Digital Cultural Heritage. In: *TCDL Bulletin Bulletin of IEEE Technical Committee on Digital Libraries* 13(1) (2017), pp. 2–4.
12. B. Gipp, N. Meuschke and A. Gernandt. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. In: (2015), pp. 1–6. URL: http://arxiv.org/abs/1502.04015.
13. T. Hepp et al. Securing Physical Assets on the Blockchain Linking a novel Object Identification Concept with Distributed Ledgers. In: (2018).
14. A. Jeppsson and O. Olsson. Blockchains as a solution for traceability and transparency. In: (2017). URL: https://lup.lub.lu.se/student-papers/search/publication/8919957.
15. K. Korpela, J. Hallikas and T. Dahlberg. Digital Supply Chain Transformation toward Blockchain Integration. In: (2017), pp. 4182–4191. URL: http://hdl.handle.net/10125/41666.
16. A. Kosba et al. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts, In: *Proceedings – 2016 IEEE Symposium on Security and Privacy, SP 2016* (May 2016), pp. 839–858.
17. P. S. Management and H. Sternberg. Master Thesis Aiming for Supply Chain Transparency: Exploring the Potential of Blockchains. In: August (2017).
18. T. Mcconaghy et al. BigchainDB: A Scalable Blockchain Database (DRAFT). In: *BigchainDB* (2016), pp. 1–65.
19. S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. In: *Www.Bitcoin.Org* (2008), p. 9. URL: https://bitcoin.org/bitcoin.pdf.
20. Pavitraa, *Technology in Supply Chain – examples from Walmart, Air-Supply*. URL: http://cmuscm.blogspot.de/2014/09/technology-in-supply-chain-examples.html%7D.
21. A. Schoenhals, T. Hepp and B. Gipp. Design Thinking using the Blockchain Enable Traceability of Intellectual Property in Problem-Solving Processes for Open Innovation, In: (2018).
22. P. Sharma. *The Economics of Bitcoin Block Size*, 2017. URL: https://blog.blockonomics.co/the-economics-of-bitcoin-block-size-e9575272c3ee.
23. K. Toyoda et al. A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in The Post Supply Chain. In: *IEEE Access* 5 (2017), pp. 17465–17477.
24. D. Vorick and L. Champine. Sia: Simple Decentralized Storage, In: (2014).
25. S. Wilkinson et al. Storj a peer-to-peer cloud storage network, In: (2014).
26. G. Wood, Ethereum: A secure decentralised generalised transaction ledger, In: *Ethereum Project Yellow Paper* 151 (2014).
27. Bin Liu, Xiao Liang Yu, Xiwei Xu. EthDrive: A Peer-to-Peer Data Storage with Provenance, in: *CEUR Proceedings*, 2017, pp. 9–18.
28. L. Xu et al. CoC: Secure Supply Chain Management System Based on Public Ledger. In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)* (2017), pp. 1–6. URL: http://ieeexplore.ieee.org/document/8038514/.
29. X. Xu et al. A Taxonomy of Blockchain-Based Systems for Architecture Design. In: *Proceedings – 2017 IEEE International Conference on Software Architecture, ICSA 2017* (2017), pp. 243–252.
30. C. Ye and X. Wang. 1 Introduction. In: *Optimization* (2007), pp. 42–47. arXiv:0007035 [hep-lat].
31. H. Zhang et al. *Distributed hash table: Theory, platforms and applications*. Springer, 2013.
32. G. Zyskind, O. Nathan and A. Pentland, Enigma: Decentralized Computation Platform with Guaranteed Privacy. In: (2015), pp. 1–14, arXiv preprint arXiv:1506.03471. URL: http://arxiv.org/abs/1506.03471.
33. G. Zyskind, O. Nathan and A. S. Pentland. Decentralizing privacy: Using blockchain to protect personal data, in: *Proceedings – 2015 IEEE Security and Privacy Workshops, SPW 2015*, 2015.

Bionotes

Thomas Hepp
University of Konstanz, Information Science Group, D-78464 Konstanz, Germany
thomas.hepp@uni-konstanz.de

Thomas Hepp has been a PhD student in the Information Science group at the University of Constance since 2016. His research focuses on blockchain technology and how it can be used to increase transparency and reproducibility in supply chains. In addition to theoretical knowledge, Thomas is passionate about transferring these research results into an innovative product, which is why he is co-founder and CTO of OriginStamp.

Matthew Sharinghousen
University of Konstanz, Distributed Systems Laboratory, D-78464 Konstanz, Germany
matthew.sharinghousen@uni-konstanz.de

Matthew Sharinghousen is a Master student with the Distributed Systems Laboratory at the University of Konstanz. His area of specialization is in secure network communication on the transport and socket layers. Previously, he researched visualization techniques of network log provenance for expert analysis.

Philip Ehret
University of Konstanz, Information Science Group, D-78464 Konstanz, Germany
philip.ehret@uni-konstanz.de

Philip Ehret is a PhD student in the Information Science Group at the University of Konstanz. His work is mainly focusing on blockchain technology and data science with special personal interest in mobile and web technologies. The combination of technologies allows Philip to address challenges in society and industry to solve real-world problems.

Alexander Schoenhals
University of Konstanz, Information Science Group, D-78464 Konstanz, Germany
alexander.schoenhals@uni-konstanz.de

Alexander Schoenhals is a PhD candidate at Daimler AG, supervised at the University of Konstanz. In the past, he has implemented several interactive systems in VR/AR with the main emphasis on haptic feedback. His current research focuses on interactive methods to recognize, track and protect intellectual property in the very first stage of the innovation cycle with novel technologies. This plan requires an interdisciplinary exchange, therefore he maintains a lively exchange with business representatives, legal experts and also representatives of his area of expertise - computer science.

Prof. Dr. Bela Gipp
University of Konstanz, Information Science Group, D-78464 Konstanz, Germany
bela.gipp@uni-konstanz.de

Prof. Dr. Bela Gipp leads the Information Science Group at the University of Konstanz, Germany. His research lies at the intersection of information science and data science, where he focuses on the retrieval, analysis, and visualization of large volumes of data. The implications of blockchain technology – for the benefit of both industry and society – is another research domain Bela is passionate about. Currently, he serves as a juror and the university partner for the worlds largest Blockchain Competition.