Challenges and solutions on architecting blockchain systems

Gregory Fournier
Polytechnique Montreal
Montreal, Canada
gregory.fournier@protonmail.com

Fabio Petrillo
University of Quebec at Chicoutimi (UQAC)
Chicoutimi, Canada
fabio@petrillo.com

ABSTRACT

Despite the fact that companies are gravitating more and more towards the use of blockchains in their systems, it is clear that the blockchains is no silver bullet. Many challenges such as scalability issues and frustrating trade-offs most notable in public decentralized blockchain systems are currently holding back blockchain's huge potential. In this paper we conduct a Systematic Literature Review in order to explore the current challenges of blockchain while presenting possible solutions to each of these challenges. We come to the conclusion that current challenges can be summarized in three categories: Scalability issues, security issues and a choice of consensus protocol. We also briefly discuss the use of blockchain in current systems, concluding that while blockchains current immaturity makes it hard to recommend for most projects, blockchains in their current state could be used in the Internet of Things.

CCS CONCEPTS

•Software and its engineering → Software architectures;

KEYWORDS

Blockchain, bitcoin, crytpocurrency, scalability, security, consensus protocol

ACM Reference format:
Gregory Fournier and Fabio Petrillo. 2018. Challenges and solutions on architecting blockchain systems. In *Proceedings of 28th Annual International Conference on Computer Science and Software Engineering, Markham, Ontario Canada, October, 2018 (CASCON'18),* 8 pages.
DOI:

1 INTRODUCTION

Ever since the rise of Bitcoin, the blockchain as a data structure has become more and more popular. Companies are eagerly looking to use blockchains outside the world of cryptocurrencies in order to replace current data structures or for future endeavors. Blockchains fundamental property of maintaining immutable information is very enticing for companies who wish to defer malicious users from tampering with data. However, due to the fact that blockchains are a rather new subject, there exists little material on architecting and designing software with blockchains compared to more traditional data structures. The challenges behind implementing or integrating a blockchain are not always emphasized, which are important motivators for this paper.

The objective behind this systematic review is to present the various challenges of architecting and implementing blockchains. By extracting popular trends and useful findings amongst the existing papers on architecting blockchain systems, this paper can serve as a guide for future architects who wish to efficiently inform themselves before designing their own system. Rather than focusing on one problem or solution, this paper seeks to give readers a brief overview of current challenges concerning blockchains. The research question is thus **Q1: What are the current challenges behind architecting and implementing blockchain systems?** As a side question, based on the results of the previous question, we will also try to briefly answer the following question: **Q1.1: Should companies adopt the blockchain considering its current state of affairs?**

Section 2 gives basic information about blockchains in order to understand the rest of the paper, while shortly discussing related works. Section 3 discusses the strategy used in order to obtain the final papers that were reviewed. Section 4 discusses the main ideas that stood out in the initial review of the papers. Sections 5, 6, and 7 answer research question **Q1** while section 8 answers question **Q1.1**. Finally, section 9 concludes the paper with the threats to validity and possible future work.

2 BACKGROUND

A *blockchain* is a basic data structure first proposed by Satoshi Nakamoto [30] in 2008 for the peer-to-peer currency known as *Bitcoin*. A blockchain is composed of many blocks, which can contain any type of data, though they are most often used to keep a record of various transactions between peers. These blocks are linked together backwards, and each block verifies the integrity of its previous block through its hash. Tampering with a previous block will invalidate its hash, making it easily noticeable. Calculating a new hash, also known as *mining* is a very demanding process, and the modification of one block has an effect on every younger block linked to it. While mining is very difficult, verifying the validity of a mined block is very easy for peers. This property of blockchains deters malicious users from modifying block data to their advantage.

Mining blocks is a CPU-intensive task. As such, users who mine (*miners*) are compensated for their work. This is commonly referred to as *Proof-of-Work*. However, as it will be discussed further on, Proof-of-Work is not the only consensus protocol.

In a *distributed blockchain*, such as bitcoin, every peer contains a copy of the complete blockchain, and several of those peers contribute to the addition of new blocks through mining. The peers also serve as judges, working together to ensure the validity and

CASCON'18, Markham, Ontario Canada

DOI:

integrity of the blockchain. Blockchains can be distributed (peer-to-peer), decentralized (not one but several points where data is transfered), and centralized (one central point of data transfer).

One of the motivations in creating the blockchain was to mitigate a common problem with distributed currencies proposals known as the *Double Spending Attack*. In a Double Spending Attack, a user is able to use an amount of currency two times, enabling him to essentially obtain twice the value of the currency spent. In a traditional centralized monetary system such as a bank, all transactions are validated by the same entity, the central bank. However, in a distributed monetary system, each and every peer can potentially validate a transaction, providing the opportunity for an attacker to make the same transaction twice and have it validated by two different peers, one being himself. Further explanation and discussion of double spending attacks can be found in section 6

The current bitcoin policy in accepting new blocks on the blockchain prevents malicious users from effecting a double spending attack, as long as the user owns less than 51% of the blockchains computing power.

These various properties of blockchains vastly reduce the possibility of hiding a tampered block by *remining* its hash value and assigning it to the correct blocks, as to attack one block a malicious user must rehash the desired block as well as every younger block linked to it. Thus, blockchains are very useful when immutability is a desired property, as long as pseudonymity is sufficient for the end users.

The most popular use of blockchain is without a doubt bitcoin, the most popular cryptocurrency which debuted the rapidly augmenting interest in blockchains and cryptocurrencies. However, there exists others besides crytocurrencies who are using blockchains, such as **Ring**, a communication platform[37].

2.1 Related works

A Taxonomy of Blockchain-Based Systems for Architecture Design regroups many important dimensions and categories for classifying blockchains and ways of using them in systems[48]. The researchers divide decision making into three main categories: architectural design regarding decentralisation, architectural design regarding storage and computation and architectural design regarding blockchain configuration. This paper gives insight on how reparametrization can solve certain issues. However, further research concludes that while reparametrization can mitigate certain scalability issues, their use should only be considered as a short term solution, as we quickly come to a point where modifying blockchain parameters can no longer be beneficial due to existing network and computing issues.

An overview of blockchain technology: Architecture, consensus, and future trends, *Blockchain Challenges and Opportunities: A Survey* and *Blockchain for the Internet of Things: a Systematic Literature Review* are also very interesting papers which explore the different challenges of blockchains. While these papers explore in depth one or two issues, this paper aims to combine these information in addition to other findings, and focus more on the different possible solutions currently available to each problem as of now[49][50][7].

Table 1: Overall Paper Subject Distribution Concerning Challenges and Solutions

Topic	Papers	Total
Scalability	[50][15][39][44][13] [2][49][47][28][24][9] [31][1][18][10][48][8] [45][32][23][42][20][27][40]	24
Security	[50][22][49][47] [41] [17][9][1] [18][7][10] [48][19][14][25][38][4] [42][20][36][12][11]	22
Consensus Protocol	[15][39][44] [29] [22][3][33][28][24][1] [10][19][25]	13
Other	[9][6][43][46][32][23][21][5][34] [35][26]	11

3 METHODOLOGY

3.1 Data source

3.1.1 Initial Source. In order to assess the feasibility of a systematic review on blockchain architecture, an initial study of 3 key papers was done[7][10][48]. Once the pertinence of this paper was established, a search was done with the query *"(architect* OR design*) AND blockchain AND system*"* which yielded 163 papers.

3.1.2 Impurity Removal and Application of Selection Criteria. The 70 top papers were analyzed, of which 20 papers were retained based on their title and credibility. Credibility was determined by number of citations per year and total amount of citations. Selection criteria favored papers that pointed out things such as problems, issues, challenges, and solutions to these aforementioned problems, but we also looked for generalized papers on blockchain architecture and papers discussing the use of blockchains besides crytocurrency.

3.1.3 Forward Snowballing. Finally, forward snowballing was done on the two most valuable papers. Since *A Taxonomy of Blockchain-Based Systems for Architecture Design* is a main pivot point of this paper, forward snowballing was done on this paper and on *Bitcoin-NG: A Scalable Blockchain Protocol*. 28 papers were kept from this snowballing. Overall, 48 papers have been retained for systematic review.

3.2 Tools

Publish or Perish was used for researching, refining queries, and keeping a tab on what articles were retained for study[16].

4 PRELIMINARY RESULTS

Of the 48 retained articles, 24 articles either pointed out the difficulties concerning blockchain scalability or offered possible solutions. Scalability is in fact one of the biggest constraints currently holding back blockchains. Most papers discuss the current limitations in the bitcoin protocol which limits its possibility of scaling via

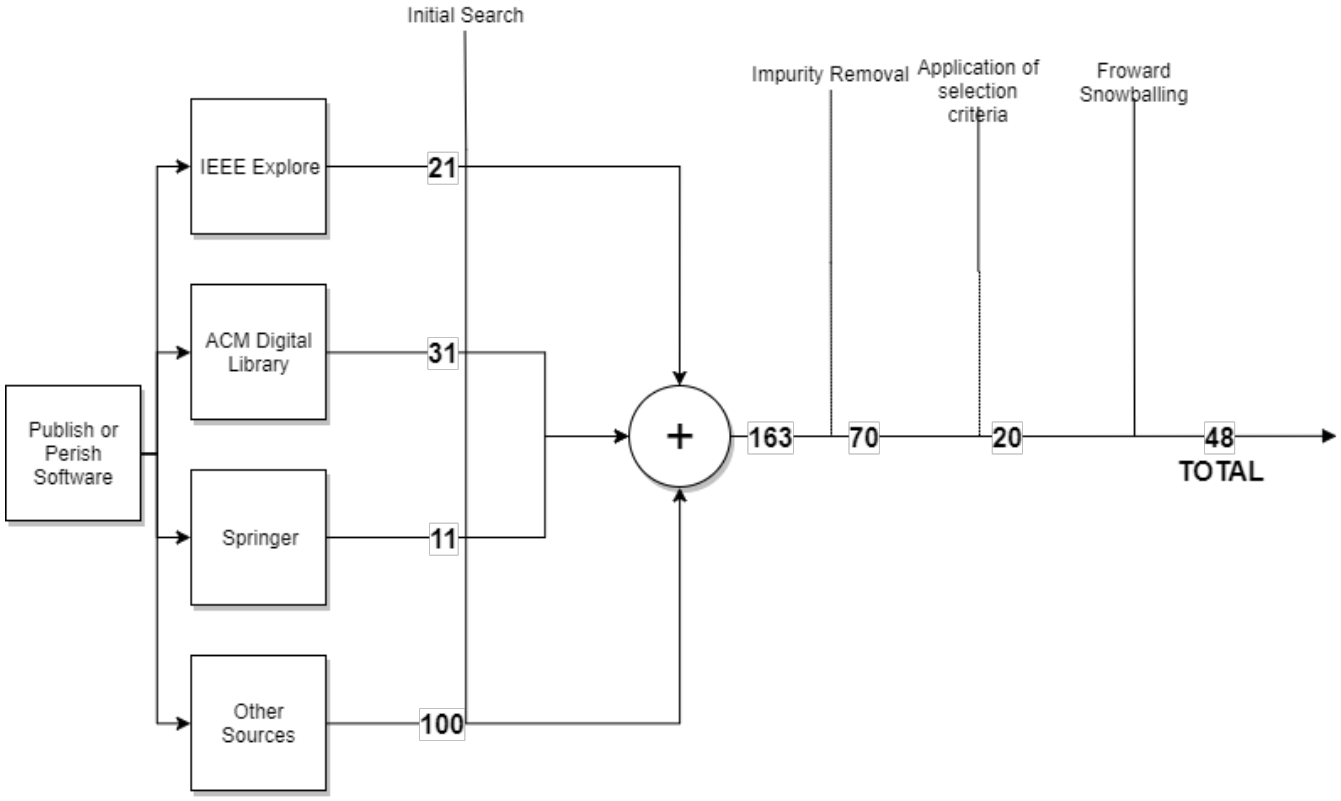

Figure 1: Overview of the search and selection process

reparametrization (eg. increasing block size) without reducing security.

Furthermore, security and user privacy was also a big concern, discussed in almost half of the researched papers. Several papers revealed the possibility of tracing blockchain transactions back to their original users, and demonstrated the possibility of certain attacks, especially on public distributed blockchains. Not only is security another huge concern, but it is also very heavily tied to scalability. As we will discuss further on, there often exists a trade off between scalability and security. The issue of *double spending* is also often brought up, saying that while the possibility of a double spending attack were currently very low, the fact that Proof-of-Work makes it possible forces blockchains to limit block time and block size.

Finally, choosing a consensus protocol, which is to say the manner in which peers may achieve consensus on the authenticity and integrity of the blockchain, can be very complicated. While bitcoin's consensus protocol, a version of Proof-of-Work, solved the problems previously associated with distrusted monetary systems, it has its limits. Novel takes on Proof-of-Work often came up in research papers, as well as other consensus protocols which aren't as dependent of computing power.

Based on these results, the following results will be divided between these 3 topics: Scalability, Security, and the consensus protocol.

5 ON THE ISSUE OF SCALABILITY

One of the biggest challenges that researchers are currently trying to solve is the issue of scalability. The current hard coded limit of 1MB in the bitcoin blockchain limits bitcoin transactions to around 3-7 transactions per second, which is considerably slower than traditional credit/debit card transactions which have speeds of almost 2000 transactions per second[45].

Proof-of-Work, the idea that by making miners spend a lot of CPU resources peers may achieve consensus on the integrity and validity of data being transmitted on the blockchain goes against scalability by definition, especially in completely decentralized systems.

Proof-of-Work is a value (proof) that is very time consuming (work) to come up with, but is easily verifiable by others. In the case of bitcoin, the value that a miner must find is what is called a *nonce* which, when concatenated to the block data and passed through a SHA256 hashing function, must generates a hash under a threshold value known as the *difficulty*. Finding the nonce is a very complicated task, but the result can be easily verified by others, as they only need to verify if the hash value is inferior to the current difficulty. In the case of bitcoin, this difficulty can be modified in relation to the average computing power of nodes on the blockchain to keep block time at a constant value despite the variances in node computing power over time. Since increasing node power will simply increase the mining difficulty, vertical scaling is practically impossible.

In a completely decentralized peer-to-peer blockchain system, such as bitcoin, every participant must keep an up-to-date copy of the blockchain. As such, the whole system is held back by the weakest nodes with the most latency, which is why difficulty must be adjusted over time. Difficulty is adjusted in order to keep the block time at around 10 minutes, a value deemed a reasonable trade-off between speed, stability and security.

In most distributed systems, computational issues can be resolved by adding more nodes or more powerful nodes to a the system. However, this isn't something that is feasible as the difficulty will change accordingly, keeping the block time at 10 minutes despite efforts to improve overall computing power on the blockchain. As such, horizontal scaling is also a difficult hurdle for bitcoin. That is why most new proposals for cryptocurrencies tend to avoid using the Nakamoto consensus, bitcoin's take on Proof-of-Work, as a consensus protocol, preferring either other protocols such as Proof-of-Stake, Hybrid consensus and Byzantine fault-tolerant protocols or using other variations of Proof-of-Work[45].

There have been heavy debates amongst the bitcoin community concerning the augmentation of the current 1mb block size limit in order to improve transaction speeds. However, most researchers agree on the fact that there are many limits on the actual effects of reparametrization. Given the current overlay network and today's desirable 10 minute average block interval, the block size should not exceed 4MB. The 10 minute average block interval is a compromise decided by the original creator of bitcoin. While a shorter block time would mean faster transactions, this would require larger bandwidth for users and the increased number of forks could cause instability within the blockchain. As such, a 10 minute delay was decided, and to maintain that delay the block size should not exceed 4MB. However, A 4MB block size corresponds to a throughput of at most 27 transactions/sec, which is still far from traditional payment methods[8] .

Another paper on the security of blockchains concluded that decreasing the average blocktime to 1 minute while keeping the block size at 1mb wouldn't have much of an impact on security. However, even this suggestion would only increase bitcoin's throughput to 60 transactions per second, which is still fairly low compared to traditional monetary systems[14].

There exists solutions to improve the current Proof-of-Work constraints. For example, rather than resolving conflicts by choosing the longest fork, the GHOST (Greedy Heaviest-Observed Sub-Tree) protocol uses weighted subtrees in order to choose which fork to continue on, providing more secure means of increasing the block frequency and the block size [44]. Bitcoin-NG is another proposal which would improve current Proof-of-Work, which uses an alternative blockchain protocol[8][18].

The easiest solution is to avoid using Proof-of-Work as a consensus protocol. There exists other consensus protocols, such as Proof-of-Stake, Proof-of-luck, and Byzantine Fault Tolerant protocols, which do not rely on miners executing intensive tasks. As such, these protocols often lend themselves more easily towards scalability, usually being limited only by network latency. Improvements to scalability and other aspects of blockchains via novel consensus protocols will be further discussed in section 7

Most of bitcoin's scalability issues come from the fact that its current protocol is hard coded into its system. Changing bitcoin's block size to something bigger would require a *hard fork*, essentially making previous nodes useless until being updated, as they would be incompatible with the new bitcoin block version. However, most researches agree that even if this was done it would only barely improve bitcoin's throughput[3]. That is why most new cryptocurrency proposals steer away from Proof-of-Work[8] .

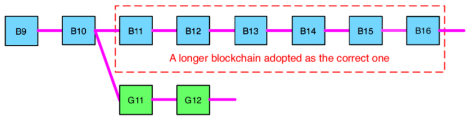

Figure 2: Reselection process of Bitcoin. The longest branch leading by four blocks will always be selected as the main chain.[1]

6 ON THE ISSUE OF SECURITY

As previously mentioned, in a distributed blockchain network, many nodes are mining blocks in parallel, fighting to be the first to add a new block to the main chain of the blockchain. When a miner starts mining new blocks, he creates a branch of the main chain. This of course means that the blockchain has many branches coming from its main chain. When several chains are formed, Bitcoin nodes accept the longest chain leading by at least four blocks as the record of transactions. The reselection of the record of transactions may cause some payments to be canceled, which when done deliberately is known as a double spending attack. As such, merchants are advised to wait that their transaction is included in a mined block, and that several blocks are chained on top of it. Ideally, a merchant would wait until three blocks are mined on top of their block, which would assure at 100% that the block on which the transaction is included on the main chain and thus valid, but waiting for this can take upwards of one hour, which isn't always desirable for merchant who handle several hundred transactions per second, As such, many merchants will accept simply one confirmation, meaning that their transaction has been included in a block, but is not guaranteed to be on the accepted chain[41].

These are the principals on which bitcoin was founded, in order to solve the double spending problem. However, what Nakamoto fails to highlight in his original paper is the possibility for attackers to *pre-mine* before launching an attack, making his analysis only approximative[41].

Pre-mining, or selfish mining, is the act of secretly mining blocks without distributing them to the system until the miner decides to. A selfish miner could mine a certain number of blocks, wait for his competitors to mine until they are only one block away from being chosen as the main chain while increasing the size of his branch, and then distribute his long chain of blocks. If the selfish miner mined enough blocks, his chain would instantly become the main chain, invalidating the other miner's chains, making them waste resources while maximizing the revenue obtained from publishing his branch. Pre-mining in order to execute a double spending attack is very similar to selfish mining. In a double spending attack, the malicious

[1] Zheng, Zibin and Xie, Shaoan and Dai, Hong-Ning and Chen, Xiangping and Wang, Huaimin. (2017). Blockchain Challenges and Opportunities: A Survey. International Journal of Web and Grid Services.

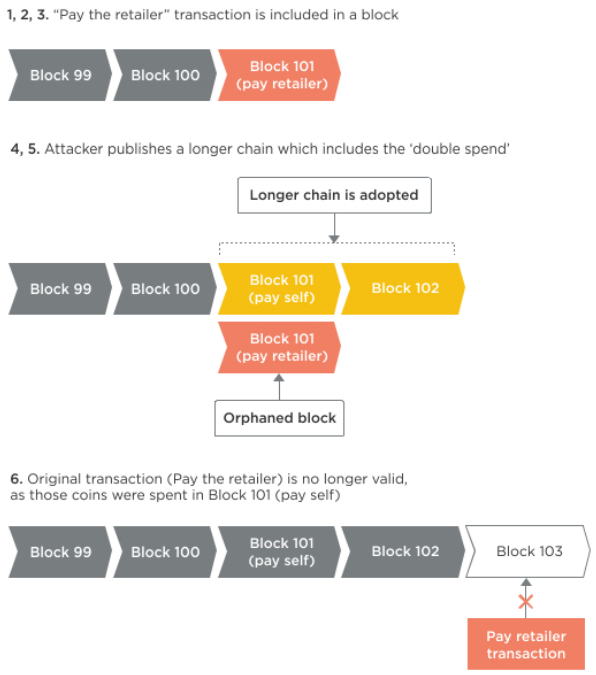

Figure 3: Overview of a double spending attack.[2]

user starts by completing a transaction with a merchant, spending a quantity 'x' of crytocurrency. The merchant then waits for one confirmation, before sending the malicious user their goods. While this is happening, the malicious user makes another transaction with the same 'x' amount of bitcoins on another branch, which he himself started mining secretly off the main chain. If the malicious user can mine enough blocks to be considered the main chain before the chain on which he made the first transaction can, he then publishes his chain, invalidating his first transaction with the merchant. He has thus successfully executed a double spending attack[2] [31] (Figure 3).

One would hope that a failed double spending attack would be costly, discouraging miners from performing them. However, assuming the miner has a reasonable advantage (a computational power of representing at least 33%), he can always keep mining to try to catch up to the main chain and eventually submit his chain as the main chain. With simple game theory, a miner with computational power representing upwards of 33% of the blockchain obtains overall better profit from always tempting double spending attacks rather than adopting honest behavior[17].

Ideally, every miner would have the same computational power, making mining a much fairer game and essentially eliminating the possibility of double spending attacks. Originally, it was assumed that no miner had an incentive to deviate from the honest strategy if the majority of miners were honest. However, this is no longer the case. A miner with computational power of at least 33% of the total blockchain power will obtain strictly better rewards by following selfish strategies rather than mining honestly. [17] Even more dangerous, if a miner (or a group of miners) were to obtains > 50% of the total computational power of the blockchain, they could execute double spending attacks with a success rate of 100%[41]. This has become an increasingly concerning problem as mining pools, groups who share their processing power and split their rewards, increase their number of miners and as such overall computational power. Collusion between the largest mining pools could result in the possibility of a > 50% attack[2][7][49].

While double spending attacks are technically doable, they are currently next to impossible to do on public blockchains as large as bitcoin. However, the existence of this vulnerability introduces an important trade-off between scalability and security. For example, increasing the block size will increase latency and block propagation time. By increasing block propagation time you increase the window of opportunity for a malicious user to execute a double spending attack, while at the same time discouraging honest miners with poor networks, decreasing overall security of the blockchain[50].

The major factor enabling double spending attacks are the fact that in a public blockchain, every miner contributes to the consensus determination. Selfish miners who execute double spending attacks are also the miners who accept their chain as being the main chain. While this is a fundamental property of public blockchains it makes scalability an issue without compromising security. One way of countering this is adjusting the level of centralization of the blockchain. In a consortium blockchain, rather than all miners participating in the consensus determination, consensus is reserved to a selected set of nodes. Depending on the consensus protocol used, this selection may be static or dynamic. Bitcoin-NG is a novel blockchain protocol which uses a random leader selection process. Spectre is another protocol which tries to eliminate the trade-off between scalability and security. As such, moving away from the Nakamoto protocol is the current solution in mitigating double spending attacks while at the same time eliminating the aforementioned trade-off.

While fairly impossible with cryptocurrencies and their desired principals, the use of a completely centralized blockchain could be interesting for uses outside cryptocurrencies on private networks. By centralizing the consensus determination efficiency greatly increases. If the central node it to be assumed to be honest, we can be assured that consensus will be done in a way that favors honest behavior. However, it becomes harder to detect if blocks have been tampered with or not. As such, private centralized blockchains should be considered in scenarios where tampered data has less of an impact: the Internet of Things is often brought up as a favorable scenario for private blockchains [7].

7 CHOOSING A CONSENSUS PROTOCOL

The majority of research on the scalability and security issues of blockchains all agree that the underlying problem lies with the current Proof-of-Work consensus protocol hard coded into bitcoin. As long as the Nakamoto protocol isn't either heavily modified or outright replaced by another consensus protocol bitcoin will

[2] Bits on blocks. (2018). A gentle introduction to blockchain technology. [online] Available at: https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/ [Accessed 22 Mar. 2018]

always be plagued with slow transaction times. As such, choosing an adequate consensus protocol plays a much bigger role than the reparametrization of blockchains (bigger block size, shorter block time, etc.). Ideally, a new consensus protocol would completely eliminate the trade-off between scalability and security. Here we present alternatives to blockchain's Nakamoto protocol.

GHOST: GHOST (Greedy Heaviest-Observed Sub-Tree) is a Proof-of-work consensus protocol with a modified policy in the selection of the main chain created in order to mitigate the double spending risk. Rather than choosing the chain with the longest amount of blocks, GHOST will instead evaluate the whole chain, choosing the chain with the move work having been done onto it. In order to accomplish this, GHOST uses a DAG (Directed Acyclic Graph) rather than simply using a linked list to maintain it's blockchain. It can thus evaluate the whole chain, choosing the chain on which the most work has been done. This can eliminate selfish mining situations that are possible with Nakamoto consensus, as can be seen in figure 4. While the GHOST protocol succeeds in increasing the difficulty of double spending attacks, the author notes that GHOST does not completely eliminate the threat. [40].

Byzantine Fault Tolerant Protocol (BFTP): The reasoning behind Proof-of-Work is that we can never assume a miner to be honest. Providing the proof-of-work is a way for miners to show that they are indeed honest miners. This problem exists because distributed blockchains are subject to what's know as the Byzantine Generals' Problem, and must achieve consensus despite that. However, researchers have been able to create Byzantine fault tolerant protocols : protocols that are robust to arbitrary types of failures in distributed algorithms. *Algorand* is a cryptocurrency using such an algorithm capable of confirming transactions with latency on the order of a minute while scaling to many users. With it's consensus protocol, Algorand ensures that users never have divergent views of confirmed transactions, even if some of the users are malicious and the network is temporarily partitioned [15].

Proof-Of-Luck: Proof-of-Luck uses a TEE (Trusted Execution Environment) platform's random number generation to choose a consensus leader, which offers low-latency transaction validation, deterministic confirmation time, negligible energy consumption, and equitably distributed mining. Proof-of-Luck is an example of a consortium blockchain, where leaders that execute the consensus determination are chosen by protocol. In the case of Proof-of-Luck, a TEE such as an Intel SGX-enabled CPU is mandated with randomly assigning leaders to achieve consensus. By removing the obligation for nodes to execute intense work, Proof-of-luck enables underpowered consumer-grade hardware to participate in mining on the blockchain, and by distributing equally the work amongst miners we can avoid selfish mining. Scaling now becomes very easy, being limited only by network latency and the amount of nodes on the network[29].

[3]Sompolinsky, Y., & Zohar, A. (2015, January). Secure high-rate transaction processing in bitcoin. In International Conference on Financial Cryptography and Data Security (pp. 507-527). Springer, Berlin, Heidelberg.

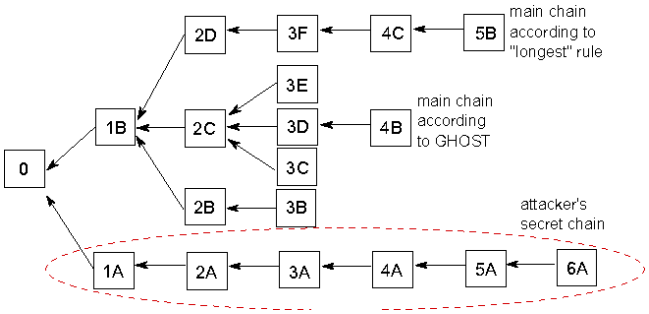

Figure 4: A block tree in which the longest chain and the chain selected by GHOST differ. An attackerfis chain is able to switch the longest chain, but not the one selected by GHOST.[3]

8 THE FEASIBILITY OF BLOCKCHAINS IN ITS CURRENT STATE

As highlighted in the previous results, blockchains still have several security and scalability issues which limit their ability to scale indefinitely, both horizontally and vertically. We can further pinpoint performance problems with an evaluation framework, such as the one proposed by Blockbench [9]. Their case study demonstrates with the help of Blockbench that consensus protocols are the main bottlenecks in the cryptocurrencies Hyperledger and Etherium. This problem is not unique to these protocols, consensus protocol is the main factor in determining the scalability of future blockchains. While many researchers agree that in its current state blockchains are not yet ready for mass usage, more research with the help of Blockbench on other examples of blockchains could lead to more accurate results [31].

Using the blockchain to face data integrity threats seems to be a natural choice, but its current limitations of low throughput, high latency, and weak stability hinder the practical feasibility of any blockchain-based solutions [13]. As such, it is hard to recommend using blockchains in its current state.

9 CONCLUSION

In this Systematic Review we discussed the various challenges of architecting blockchain systems and possible future solutions to mitigate those challenges problems. We discussed how current blockchains such as Bitcoin are fundamentally limited in scalability due to their underlying protocol and the limits of mitigation through reparametrization. We discussed current blockchain security issues such as double spending attack and how novel crytocurrencies and consensus protocols were being developed to further eliminate the possibility of a double spending attack. Consensus protocols were then presented, demonstrating the various choices besides Proof-of-Work that currently exist. Finally, we discussed the feasibility of blockchains outside of cryptocurrencies in their current chain, recommending companies should wait on the more widespread adoption of novel consensus protocols such as GHOST and BFTP.

While this paper brings out the challenges and solutions concerning blockchains, it only scratches the surface of the whole subject. A more in depth of each problem category (scalability, security and

consensus protocols) would provide researchers and practitioners with a better comprehension of the existing challenges and the solutions available to them.

Blockchains are a very new technology. Advances are being done every day. As such, it would be interesting to do this further analysis when blockchains achieve a more mature status. This could potentially provide the future paper with more solutions to the existing problems.

REFERENCES

[1] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2016. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. *arXiv preprint arXiv:1612.02916* (2016). Query date: 2018-02-26.

[2] Emmanuelle Anceaume, Thibaut Lajoie-Mazenc, Romaric Ludinard, and Bruno Sericola. 2016. Safety analysis of Bitcoin improvement proposals. *Network Computing and Applications (NCA), 2016 IEEE 15th International Symposium on* (2016), 318–325. Query date: 2018-02-26.

[3] Ethan Buchman. 2016. Tendermint: Byzantine fault tolerance in the age of blockchains. (2016). Query date: 2018-02-26.

[4] E CHEN. 2016. *An Approach for Improving Transparency and Traceability of Industrial Supply Chain With Block-chain Technology.* dspace.cc.tut.fi. https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/25401/Chen.pdf?sequence=3 Query date: 2018-01-31.

[5] JB Cholewa and AP Shanmugam. 2017. Trading Real-World Assets on Blockchain-An Application of Trust-Free Transaction Systems in the Market for Lemons. *Business & Information Systemsfi* (2017). http://aisel.aisnet.org/bise/vol59/iss6/4/ Query date: 2018-01-31.

[6] Konstantinos Christidis and Michael Devetsikiotis. 2016. Blockchains and smart contracts for the internet of things. *IEEE Access* 4 (2016), 2292–2303. Query date: 2018-02-26.

[7] Marco Conoscenti, Antonio Vetro, and Juan Carlos De Martin. 2016. Blockchain for the Internet of Things: A systematic literature review. *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of* (2016), 1–6. Query date: 2018-02-26.

[8] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, and Emin Gn Sirer. 2016. On scaling decentralized blockchains. *International Conference on Financial Cryptography and Data Security* (2016), 106–125. Query date: 2018-02-26.

[9] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. 2017. Blockbench: A framework for analyzing private blockchains. *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), 1085–1100. Query date: 2018-02-26.

[10] Ittay Eyal, Adem Efe Gencer, Emin Gn Sirer, and Robbert Van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol. *NSDI* (2016), 45–59. Query date: 2018-02-26.

[11] MA Ferrag, LA Maglaras, H Janicke, J Jiang, and … 2017. Authentication Protocols for Internet of Things: A Comprehensive Survey. *Security andfi* (2017). https://www.hindawi.com/journals/scn/2017/6562953/abs/

[12] P Fremantle and P Scott. 2015. *A security survey of middleware for the Internet of Things.* peerj.com. https://peerj.com/preprints/1241.pdf

[13] Edoardo Gaetani, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. 2017. Blockchain-based database to ensure data integrity in cloud computing environments. (2017). Query date: 2018-02-26.

[14] Arthur Gervais, Ghassan O Karame, Karl Wst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), 3–16. Query date: 2018-02-26.

[15] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. *Proceedings of the 26th Symposium on Operating Systems Principles* (2017), 51–68. Query date: 2018-02-26.

[16] Anne-Wil Harzing. 2016. Publish or Perish Website. (2016).

[17] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. 2016. Blockchain mining games. *Proceedings of the 2016 ACM Conference on Economics and Computation* (2016), 365–382. Query date: 2018-02-26.

[18] Aggelos Kiayias and Giorgos Panagiotakos. 2016. On Trees, Chains and Fast Transactions in the Blockchain. *IACR Cryptology ePrint Archive* 2016 (2016), 545. Query date: 2018-02-26.

[19] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. 2016. Enhancing bitcoin security and performance with strong consistency via collective signing. *25th USENIX Security Symposium (USENIX Security 16)* (2016), 279–296. Query date: 2018-02-26.

[20] B KOTESKA, E KARAFILOSKI, and A MISHEV. 2018. Blockchain Implementation Quality Challenges: A Literature. (2018). http://ceur-ws.org/Vol-1938/paper-kot.pdf Query date: 2018-01-31.

[21] SU Lee, L Zhu, and R Jeffery. 2018. Designing Data Governance in Platform Ecosystems. *fiof the 51stfi* (2018). https://scholarspace.manoa.hawaii.edu/handle/10125/50515 Query date: 2018-01-31.

[22] Joshua Lind, Ittay Eyal, Peter Pietzuch, and Emin Gn Sirer. 2016. Teechan: Payment channels using trusted execution environments. *arXiv preprint arXiv:1612.07766* (2016). Query date: 2018-02-26.

[23] Q Lu and X Xu. 2017. Adaptable Blockchain-Based Systems: A Case Study for Product Traceability. *IEEE Software* (2017). http://ieeexplore.ieee.org/abstract/document/8106871/ Query date: 2018-01-31.

[24] Loi Luu, Viswesh Narayanan, Kunal Baweja, Chaodong Zheng, Seth Gilbert, and Prateek Saxena. 2015. SCP: A Computationally-Scalable Byzantine Consensus Protocol For Blockchains. *IACR Cryptology ePrint Archive* 2015 (2015), 1168. Query date: 2018-02-26.

[25] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. 2016. A secure sharding protocol for open blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), 17–30. Query date: 2018-02-26.

[26] I Mas and DLEEK Chuen. 2015. Bitcoin-Like Protocols and Innovations. *Handbook of Digital Currency* (2015). https://www.sciencedirect.com/science/article/pii/B9780128021170000217

[27] T McConaghy, R Marques, A Mller, and … 2016. BigchainDB: a scalable blockchain database. *white paperfi* (2016). https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf

[28] Trent McConaghy, Rodolphe Marques, Andreas Mller, Dimitri De Jonghe, Troy McConaghy, Greg McMullen, Ryan Henderson, Sylvain Bellemare, and Alberto Granzotto. 2016. BigchainDB: a scalable blockchain database. *white paper, BigChainDB* (2016). Query date: 2018-02-26.

[29] Mitar Milutinovic, Warren He, Howard Wu, and Maxinder Kanwal. 2016. Proof of luck: an efficient blockchain consensus protocol. *Proceedings of the 1st Workshop on System Software for Trusted Execution* (2016), 2. Query date: 2018-02-26.

[30] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer electronic cash system. (2008).

[31] Christopher Natoli and Vincent Gramoli. 2016. The blockchain anomaly. *Network Computing and Applications (NCA), 2016 IEEE 15th International Symposium on* (2016), 310–317. Query date: 2018-02-26.

[32] B Notheisen, JB Cholewa, and AP Shanmugam. 2017. Trading Real-World Assets on Blockchain. *Business & Informationfi* (2017). https://link.springer.com/article/10.1007/s12599-017-0499-8 Query date: 2018-01-31.

[33] Rafael Pass and Elaine Shi. 2017. Hybrid consensus: Efficient consensus in the permissionless model. *LIPIcs-Leibniz International Proceedings in Informatics* 91 (2017). Query date: 2018-02-26.

[34] M Risius and K Spohrer. 2017. A Blockchain Research Framework. *Business & Information Systems Engineering* (2017). https://link.springer.com/article/10.1007/s12599-017-0506-0 Query date: 2018-01-31.

[35] N Roth. 2015. An architectural assessment of bitcoin: using the systems modeling language. *Procedia Computer Science* (2015). https://www.sciencedirect.com/science/article/pii/S1877050915003026

[36] P Sarigiannidis, E Karapistoli, and … 2017. Modeling the Internet of Things Under Attack: A G-network Approach. *IEEE Internet of Thingsfi* (2017). http://ieeexplore.ieee.org/abstract/document/7956134/

[37] savoirfairelinux. Ring Website. (????).

[38] I Singh and SW Lee. 2017. Comparative Requirements Analysis for the Feasibility of Blockchain for Secure Cloud. *Asia Pacific Requirements Engeneering Conference* (2017). https://link.springer.com/chapter/10.1007/978-981-10-7796-8_5 Query date: 2018-01-31.

[39] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. 2016. SPECTRE: A Fast and Scalable Cryptocurrency Protocol. *IACR Cryptology ePrint Archive* 2016 (2016), 1159. Query date: 2018-02-26.

[40] Yonatan Sompolinsky and Aviv Zohar. 2015. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 507–527.

[41] Yonatan Sompolinsky and Aviv Zohar. 2016. Bitcoin's security model revisited. *arXiv preprint arXiv:1605.09193* (2016). Query date: 2018-02-26.

[42] M Staples, S Chen, S Falamaki, A Ponomarev, and … 2017. *Risks and opportunities for systems using blockchain and smart contracts. Data61.* data61.csiro.au. https://www.data61.csiro.au/~/media/D61/Files/Blockchain-reports/Blockchain-RisksandOpps-HTML.html?la=en&hash=B30AF266CCDE4BC81684C676AE70E61E72E9E995 Query date: 2018-01-31.

[43] P Tasca, T Thanabalasingham, and CJ Tessone. 2017. Ontology of Blockchain Technologies. Principles of identification and classification. *arXiv preprint arXivfi* (2017). https://arxiv.org/abs/1708.04872

[44] Jason Teutsch and Christian Reitwiener. 2017. A scalable verification solution for blockchains. *:(fiff 2017). url: https://people. cs. uchicago. edu/ teutsch/papers/truebit. pdf* (2017). Query date: 2018-02-26.

[45] Marko Vukolifj. 2015. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. *International Workshop on Open Problems in Network Security* (2015), 112–125. Query date: 2018-02-26.

[46] H Wu, Z Li, B King, Z Ben Miled, J Wassick, and J Tazelaar. 2017. A Distributed Ledger for Supply Chain Physical Distribution Visibility. *Information* (2017). http://www.mdpi.com/2078-2489/8/4/137 Query date: 2018-01-31.

[47] Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, and Shiping Chen. 2016. The blockchain as a software connector. *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on* (2016), 182–191. Query date: 2018-02-26.

[48] Xiwei Xu, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, Cesare Pautasso, and Paul Rimba. 2017. A taxonomy of blockchain-based systems for architecture design. *Software Architecture (ICSA), 2017 IEEE International Conference on* (2017), 243–252. Query date: 2018-02-26.

[49] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An overview of blockchain technology: Architecture, consensus, and future trends. *Big Data (BigData Congress), 2017 IEEE International Congress on* (2017), 557–564. Query date: 2018-02-26.

[50] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, and Huaimin Wang. 2016. Blockchain challenges and opportunities: A survey. *Work Pap.fi2016* (2016). Query date: 2018-02-26.