An overview of blockchain scalability, interoperability and sustainability

Kaihua Qin, Arthur Gervais

Hochschule Luzern
Imperial College London
Liquidity Network

1. Introduction

Since Bitcoin’s introduction, have the non-custodial financial and programmable possibilities of blockchain attracted extensive worldwide attention. Similar to other groundbreaking and novel technologies, blockchain still suffers from fundamental internal and external issues. This high level report revolves around three themes – scalability, interoperability and sustainability of nowadays blockchains.

Before starting this discussion, a preliminary definition of blockchain is essential. A blockchain is generally seen as an append-only ledger, or timestamping service of events. It is the very immutability which provides most value that a blockchain offers - once something has been written to it, it cannot be removed.
The next question to answer would be, how to write on the blockchain? It is here that the world separates into two fundamentally different types of blockchains. First, the open and freely accessible blockchain where anyone with sufficient capital can choose to join and become a writer of the ledger - often referred to as **permissionless blockchain**, and second, the closed and typically by consortia governed blockchain - so called **permissioned blockchain**.

Naturally, the answers to the questions of scalability, interoperability and sustainability are fundamentally different depending on which blockchain type is observed. Permissionless blockchain are inherently more costly in terms of maintenance, as they allow the freedom for new participants to leave and join at any point in time. Permissioned blockchains on the other hand, are in practice built out of legally binding documents between different parties, which attest to each other their intent to form a blockchain network. Intuitively, scaling a rather static and pre-agreed network is understandably more affordable and less challenging than a dynamic and open network.

Over 70% of the current market capitalization is held by open and permissionless blockchains, run by so called Proof of Work (PoW). Proof of Work allows someone with sufficient capital to solve a computationally hard problem in order to validate blockchain transactions and write them to the blockchain. Notably anyone is free to join this network and contribute towards it’s processing power.

2. Blockchain Scalability Opportunities

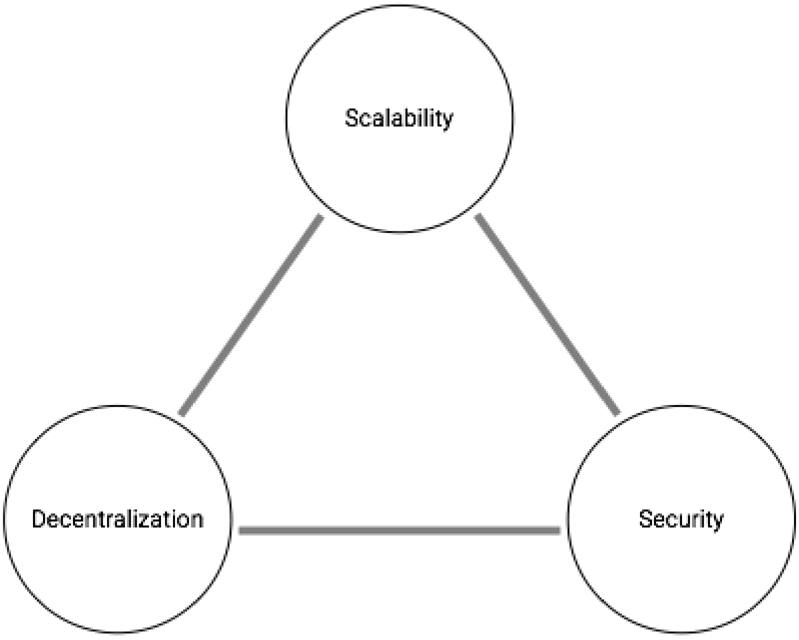
Proof of Work based permissionless blockchains currently offer a transaction throughput of about 10 transactions per second. With simple technical tweaks, those could potentially be improved to about 100 transactions per second, without deteriorating their security provisions of an open and decentralized network[1]. This comparably low throughput is a challenge for blockchain adoption, which however already is solved for e.g. financial applications, as we will outline further below.

Permissioned blockchains typically offer much higher transaction throughputs. The reason being that those in control, those who validate the blockchain transactions, are known upfront and are typically a fixed set of corporate instances. Existing consensus protocols for closed sets are naturally much faster, than an open and permissionless network. It is unclear whether permissioned blockchains offer the non-custodian property, as all trades and applications are effectively controlled by the predetermined validators.

Comparable traditional centralized payment systems like VISA and PayPal, offer significantly higher transactions per second (TPS), however remain custodian of the user's assets[2].

2.1 The Scaling Trilemma

There is a loose trilemma law that blockchains can have at most two of the three properties: (i) decentralization, (ii) scalability and (iii) security. In a permissionless blockchain, all the transactions and blocks are broadcast, verified and recorded among all participants in a decentralized peer-to-peer network. This process ensures that the whole system is immutable, stable and resistant as long as more than half of the computing resources remain honest.



Honest majority is required for an appropriate security property, which however is very costly on the scalability side as all participants need to be informed and to implicitly agree.

Let's define decentralization as a property similar to redundancy and no single

[1] Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016

[2] https://usa.visa.com/about-visa/visanet.html

point of control. Then, permissioned blockchains make the tradeoff to only allow a fixed set of participants to control the underlying ledger.

Figure 1: The Scaling trilemma - it's hard to achieve scalability, security and decentralization at the same time.

Scalability of PoW blockchain is inherently affected by some endogenous factors, such as the *block time interval* and the *block size*. Intuitively, reducing the time interval does improve performance, however might increase the risk of blockchain forks and thereby reduce security. Block size is another contributing factor - the larger a block, the more transactions can be carried within, which will lead to a slower network propagation, which itself again leads to lower security properties. Blind modification of the said parameters might make the whole system vulnerable to a variety of attacks, such as selfish mining and double spending.

To better understand the aforementioned parameter selection, we have built an open-source blockchain simulator to test and verify how different consensus and network parameters affect the whole system[3]. We also constructed an explicit mathematical model, fed by the stale block rate which is the output of the simulator and some other inputs like the adversarial fraction of the total mining power, to evaluate security. Our study shows that, by appropriately adjusting some parameters, a scalability factor of 10 can be achieved, without sacrificing security. Namely, Bitcoin could increase the current transaction throughput by ten-fold, with 1 simple parameter change, and without sacrificing security[4].

2.2 Blockchain layers

Blockchains, whether permissioned or permissionless, are typically organized in different layers (cf. Figure 2). When referring to a blockchain, we're often discussing layer 1, where the chain of blocks are formed - the blockchain. For this to happen, there are important communication requirements in place - those typically occur over the internet, respectively the network layer. The hardware layer implicitly allows to compute on the received data and to produce appropriate outputs.

What's interesting, in particular regarding the scalability opportunities, is layer 2. Layer 2 would not only allow to perform financial transactions, but potentially enables new forms of applications, as well as opportunities to improve the scaling challenges of layer 1. Having a slow layer 1, therefore does not imply to have a slow layer 2 blockchain! Note that layer 2 can also be a separate blockchain in itself.

Examples of scaling solutions on the different layers are the following. Scaling solutions on layer 0 would correspond to faster network propagation mechanism as in to optimize the communication throughput. On layer 1, we could think about different blockchain structures, such as a directed acyclic graph (DAG), which however might introduce additional

[3] https://github.com/arthurgervais/Bitcoin-Simulator

[4] Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016.

complexities. Possible scalability solutions on layer 2 would be off-chain channels or hubs, as well as side chains. A side chain is an independent chain, governed by e.g. a consortia, whereby an off-chain protocol typically does not require an additional consensus mechanism.

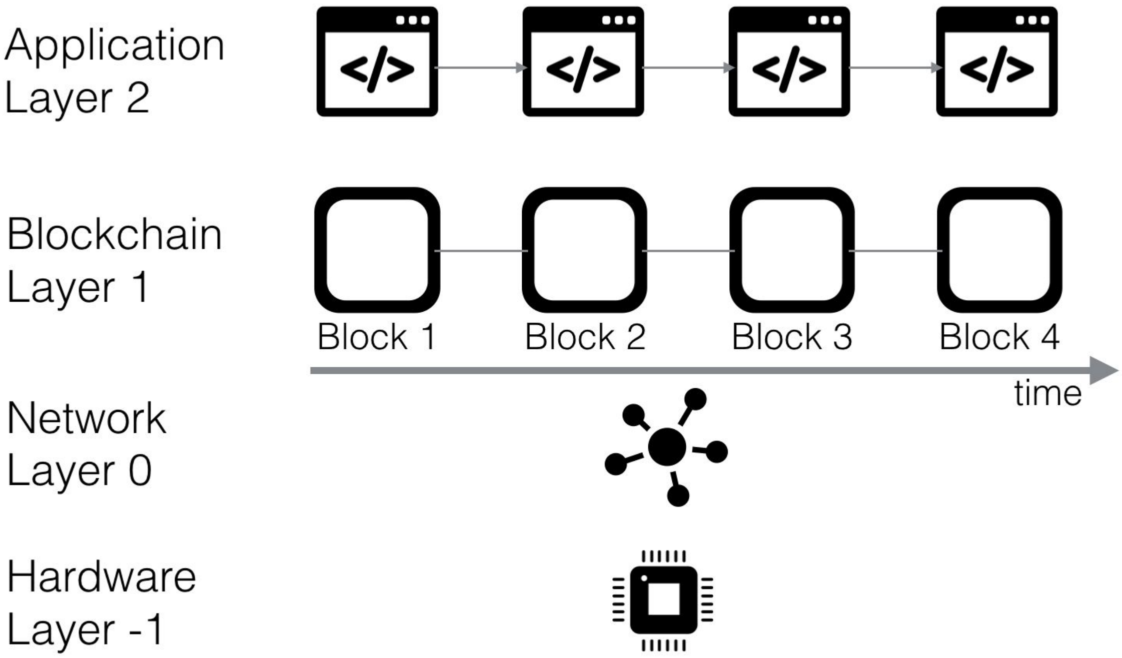


Figure 2: The different layers that a blockchain is constructed with. Note that layer 2 could also be another blockchain, for example an application specific blockchain.

2.3 Blockchain Forks

A fork refers to the point that a blockchain is split into different versions. PoW blockchains may fork accidentally at any point in time when two or more miners simultaneously find a conflicting block. A fork is resolved once subsequent blocks are mined and by determining the chain with the most proof of work.

Some forks are initiated intentionally to modify or upgrading the protocol, usually for the purpose of bug fixing and performance enhancements. The community typically differentiates between two types of forks: (1) a hard fork is a rule change that requires all nodes to update, whereas (2) a soft fork is “backward-compatible”. Under the circumstance that a hard fork is partly accepted, the blockchain would turn into separate chains. For instance, Ethereum hard-forked into Ethereum and Ethereum Classic in 2016.

The remainder of this chapter introduces existing scalability solution attempts.

A. Alternatives to proof-of-work

One of the fundamental differences between centralized ledgers and blockchains is that the bookkeeper is randomly selected from a crowd of decentralized candidates. In Proof of Work, candidates are competing on computationally intensive puzzles and are randomly selected, proportional to their processing power, and then allowed to write to the chain.
In Proof of Stake (PoS) systems those random elections are based on financial stakes. The stakes can take up different forms, e.g. total amount of stake or the age of a particular set of coins.

Proof of Stake, however, is not a scalability solution. PoS is aimed at reducing the energy consumption problem of PoW and likely has different trust assumptions than PoW.

Except for PoS, there exist plenty alternatives of PoW such as delegated Proof of Stake (DPoS) and practical Byzantine fault tolerance (PBFT). A detailed comparison and investigation of such proposals are beyond the scope of this report[5]. A possible infographic can be found in the appendix.

B. Sharding

In a traditional blockchain network, transactions and blocks are validated by all the participating nodes. One approach to scaling would be to parallelize the blockchain, which is called sharding. Sharding is a technique that is widely used in large-scale databases. Data are horizontally partitioned into pieces, each called a shard, in order to provide high concurrency.

In the case of blockchains, sharding is considered complex it would require the cautious design of a partitioning mechanism to make sure all shards are sufficiently decentralized. Otherwise, a shard could be easily controlled by a single entity – the single-shard takeover attacks.
The next challenge is inter-shard communication. If cross shard communications are too frequent, sharding might become less efficient than a non-sharded blockchain. Further issues like fraud detection and data availability make designing blockchain sharding challenging. Despite the challenges and difficulties, sharding is a promising technique that will make a tremendous difference on the blockchain scalability issue.

Several sharding schemes have been proposed, notably in conjunction with PoS systems[6]. Those for example propose that each shard holds data in shard blocks, or collations, forming a chain structure. Structurally that seems similar to the traditional blockchain, partitioned into sub-blockchains. At the same time, the main chain only stores the summarized global state,

[5] SoK: Consensus in the Age of Blockchains, Bano et al., https://arxiv.org/pdf/1711.03936.pdf
[6] https://github.com/ethereum/wiki/wiki/Sharding-FAQs

for example the collation headers of all the shards. Hence, the cross-links between collations in shards and the main chain are established.

Nodes in a sharded blockchain take on different roles – proposers are tasked with marshalling transactions and creating collations; validators are tasked with verifying those collations and reaching consensus. The validators are randomly selected and shuffled to a new shard every round. This is to prevent validators from knowing which shard they will validate and thus to reduce risks of the single-shard takeover attacks. Although the prototypes of this solution have been modelled like what is described previously, due to their technical complexity, sharding still is in the scoping epoch – there exists no sharding chain that is operational.

C. Off-chain payments and computation

Off-chain technology, as the name implies, does no longer perform transactions on the blockchain. It generally performs transactions off the chain, which however are secured by the underlying chain that acts as a dispute mediator.

Various types of off-chain solutions have emerged in recent years, many of which rely on the micro-payment channels. Payment channels permit users to defer submitting transactions on-chain and settle the final state at a later time.

The life cycle of payment channel falls into three phases. First, to establish the channel, both counterparties fund an anchor transaction broadcast on-chain to lock certain amount of assets between them. Second, the two sides can perpetually exchange commitment transactions off-chain to reallocate the initial locked funds. Every time new transactions are generated, the previous ones will be repealed via for example the Revocable Sequence Maturity Contract (RSMC)[7]. RSMC ensures that any side attempting to profit improperly by submitting an obsolete commitment transaction on-chain will be punished with losing all funds inside the channel.

Third and lastly, a settlement transaction is submitted on the blockchain marking the channel closure.

There are other types of payment channels, e.g. those based on commitment transactions with time locks. The latest transaction with the shortest time lock will “double-spend” previous transactions, which is equivalent to revocation. Payment channels suit the use case scenario where micro-payments happen frequently.

[7] The workflow of RSMC is described as follows. Assume Alice and Bob are transacting through a payment channel. Every time a new off-chain payment is generated, Alice will create a revocation key and prepare a commitment transaction with two outputs for Bod to sign. The first output is sent to Bob directly and the second output is sent to 2-of-2 multisignature address controlled by Bob and the revocation key. Bob will sign this transaction and at the same time sign a transaction to send the second output to Alice after a long time lock of, for example, 1000 blocks. Next time, when another new commitment transaction is generated, Alice will reveal the old key to Bob as the revocation. If Alice tries to broadcast the old state on-chain, Bod will have enough time to redeem the second output with Alice’s revocation key. For details, refer to https://lightning.network/lightning-network-paper.pdf.

Payment channels can be further meshed together into a network, so that payments can flow hop-by-hop via Hashed Timelock Contract (HTLC)[8]. To perform a multi-hop payment, the payee should generate a random number, namely the preimage, and send its hash to the payer. Then, starting from the payer, a payment in every hop will require the recipient to present the preimage, which can be validated by the hash to redeem the funds.
As the payee knows the preimage, the redemptions will proceed hop-by-hop in the opposite direction to finalize the whole payment. HTLC doesn't require any trust though the routing path as the whole transaction amount is collateralized.

C.1 Payment Channel Challenges

A payment channel network allows participants that are not directly connected to transact, however faces several challenges. For instance, every channel requires the transaction volume to be collateralized. A single entity that wants to serve 1000 peers with on average USD 1000, would need to perform 1000 on-chain transactions to initiate those channels, and lock up USD 1M as collateral.

Moreover, peers are supposed to be *continuously* online in a payment channel network, in order to observe the counterparty's behaviour and possibly report misbehaviour. To enable the practical application of payment channels, so called watchguards must therefore be introduced who would be monitoring the transactions to ensure the safety of funds.

The likely most prevalent usability concern of payment channels, however, is the requirement that the recipient is online at the time of payment reception. This requirement complicates, or even renders impractical the real-world usage of channels for many use cases.

Routing is another major complexity obstacle in channel networks, and several projects are working towards alleviating routing complexities. Routing however is also dependent on the available collateral and as such, even if solved, wouldn't likely suffice to make payment channels fully practical. Longer and more complex routing paths, would increase the transaction costs and switching between different paths may expose the users' privacy. Lightning Network for example announced their routed payment channels scheme in 2015[9], but due to a variety of complications, only accomplished the first mainnet-ready implementation three years later. A lot of work proposes to improve current payment channel networks to improve their robustness and scalability properties. REVIVE[10] for example allows the balance of collateral between channels to be redistributed without costly blockchain transactions. Compared with individual collateral management, this solution is more flexible and beneficial to routing optimization.

[8] Detailed mechanism of HTLC will be described in the Hash-locking section of the Interoperability chapter.
[9] Lightning Network – https://lightning.network/ introduced this notion on Bitcoin. Raiden Network – https://raiden.network/ copied the idea on Ethereum.
[10] Khalil, Rami, and Arthur Gervais. "Revive: Rebalancing off-blockchain payment networks." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.

Several other off-chain scaling solutions are currently in the development phase. NOCUST[11], Plasma, Perun are such examples, all with different trust assumptions and different scaling properties. NOCUST[12] for example offers the possibility to operate trustless N-party offchain hubs. End users can register to a particular hub without an on-chain transaction, routing paths amid different hubs are static, long-lived and therefore affordable.

It’s likely that these off-chain solutions become very robust and offer a significant layer on top of existing blockchains which simply improves the scalability while reducing transaction costs. They moreover have the chance to improve the users privacy as transactions are no longer broadcast globally.

What’s *fundamental* is that any off-chain project should properly define and formalize it’s security and trust assumptions. It’s trivial to make a system scalable when compromising on the former, the limitations should therefore formally be described and ideally be proven.

3. Interoperability

Blockchain interoperability generally tackles the ability of sharing states and transacting across different chains[13]. Blockchains can be seen as isolated databases, without proper interfaces for in- or output of data. Blockchain interoperability could enrich use cases for blockchains like portable assets, payment-versus-delivery and cross-chain oracle. Ideally, different blockchains would be abstracted, such that a user can readily manipulate all the functions without accurate understanding of each blockchain. Notary schemes, hash locking, relays and sidechains are the existing primary solutions that will be introduced seriatim.

A. Notary schemes

In notary schemes, transactions highly depend on a third-party notary. When both parties of a transaction across different chains mistrust each other and their information is asymmetric, the simplest method is to seek an intermediary trusted by both sides. The notary can be a centralized entity or a set of entities, who will claim some state or transaction on the targeted chains, either proactively or reactively. Centralized notaries are often criticized for despotism while federated notaries likely mitigate this weakness. For federated notaries, Byzantine-fault-tolerant algorithm and multi-signature verifications are usually utilized for the security purposes.

Technically, notary schemes are simple to implement. The security properties rely on the credits and honesty of the notaries, similar to traditional centralized exchanges and banks. Despite this drawback, notary schemes have been widely used due to their efficiency and flexibility. For example, notary solutions can easily support exchanges between legal

[11] https://liquidity.network/NOCUST_Liquidity_Network_Paper.pdf
[12] https://liquidity.network/, co-founded by Dr. Arthur Gervais and Rami Khalil
[13]XCLAIM: Trustless, Interoperable, Cryptocurrency-Backed Assets, Zamyatin et al, https://eprint.iacr.org/2018/643.pdf, IEEE S&P 2019

tenders, which don't support smart contracts. Similarly, based on a number of third-party connectors, the Interledger Protocol (ILP) provides a network that connects different digital asset ledgers including banks and blockchains[14].

As the typically most trusted entities, governments seem suitable actors as notaries.

B. Hash-locking

HTLC is utilized to make payments routable across multiple payment channels, and hash locking is also suitable for cross-chain atomic swaps. Supposing Alice and Bob want to exchange assets on different chains, the mechanism is as follows:

1. Alice generates a secret key *s*, calculates its hash *h* and sends *h* to Bob.
2. Alice locks her assets into a contract that Bob can redeem the funds with *s* provided within time *y* or the funds will be sent back to Alice. Bod then locks his assets into a contract that Alice can redeem the funds with *s* provided within time *x*, such that *y* is larger than *x*, or the funds will be sent back to Bob.
3. Alice reveals *s* within time *x* to redeem coins from Bob. Bob then learns *s* and can redeem coins from Alice.

The time reserved for Alice and Bod to redeem assets is not definite and varies across different chains. This process is arguably atomic – enough time should be reserved for Bob to redeem funds after Alice's action. Otherwise, Alice could refuse to reveal *s* and all the funds would be sent back. Yet Alice dominates in some scenarios where, for example, exchange rate fluctuations exist. Alice could wait within the time *x* for a favorable exchange rate. To mitigate such a disequilibrium, state channels are frequently exploited to expedite exchanges and thereby dent the advantages of the initiator. In the case of cross-chain operations, hash locking can only support atomic swaps, with difficulties in asset portability and cross-chain oracle.

C. Relays and sidechains

The definition of sidechain is expounded as "a sidechain is a blockchain that validates data from other blockchains"[15]. BTC Relay[16] is considered to be the earliest project introducing the concept of relay and sidechain. Here, a significant technique, simplified payment verification (SPV), which will be mentioned repeatedly later, is worth describing firstly. SPV is a method utilized by many lightweight clients to cryptographically verify whether particular transactions are included in a blockchain without downloading the entire chain. An SPV client holds a chain of blockheaders demonstrating proof of work. Transaction verification can be operated with only a small amount of inputs and calculations on the Merkle tree from the transaction till the root hash, which is included in the block header. Back to the BTC

[14] https://interledger.org/rfcs/0003-interledger-protocol/
[15] https://blockstream.com/sidechains.pdf
[16] http://btcrelay.org/

Relay scheme, a community of relayers continuously submit block headers up to date with Bitcoin blockchain. DApps on Ethereum can then validate Bitcoin transactions through the BTC Relay contract with the build in SPV logic. Hence, states of the Bitcoin blockchain are relayed to Ethereum. Yet BTC Relay is uni-directional, since Bitcoin scripts are not sufficiently elaborate to read Ethereum data. Relays are limited by the consensus time of the underlying blockchain, which in turn slows down cross-chain operations.

The BTC relay is not a fully-fledged sidechain implementation, as it is built for a very specific use case which is to allow Smart Contracts on the Ethereum blockchain to receive proof of payments made on the Bitcoin blockchain. More generically, the term “sidechain” is used to refer to the pegged subordinate chains. Pegged sidechains are able to not only read data from the parent chain, but also support asset portability. Despite the technical feasibility in one-way, the asset transfers are commonly two-way. To implement the two way peg, some assets on the parent chain should be locked in certain method and new coins are hence created on the sidechain with a deterministic exchange rate. Likewise, the parent chain assets can be unlocked only if the created coins are destroyed on the sidechain. Essentially, transfers in the two-way peg scheme are the processes of locking and unlocking assets across different chains.

The ideas from notary schemes can be imitated here. One can send coins to a custody or a multi-signature address controlled by a federation, and, once confirmed, receive the corresponding tokens on the pegged sidechain. Custodies take charge of overseeing the parent chain and governing coins on the sidechain. This is the quickest way, while the user has to trust the custodian.

There also exists a trustless mechanism that utilizes SPV to peg one chain to another. The parent assets to be transferred are sent to a special output locked by a SPV proof from the sidechain. Next, the SPV proof of this transaction is provided to the sidechain and the corresponding tokens are consequently released, that can be traded freely on the sidechain. Operations in the reverse direction are symmetric – providing SPV proof to the parent chain that those coins are destroyed on the sidechain to unlock the parent assets. Unfortunately, the latency for a SPV based transfer is high since users have to wait a confirmation period before sending a SPV proof to resist denial of service attacks and a contest period before trading transferred assets to prevent double-spending. In the contest period, the previous SPV proof will be invalidated if there is a chain with more aggregate work not containing the locking transaction, which is called a reorganisation proof. To avoid such a delay, users who want to move assets from one chain to the other can swap with those who happen to have the requests in the opposite direction, with the help of the hash locking mechanisms. This is much faster than that they transfer separately using SPV proofs. If participants of the sidechain are full validators of the parent chain, transfers to the sidechain will not require SPV proofs, while still require in reverse, since the parent chain knows nothing about the sidechain. Such an asymmetric two-way peg mechanism will improve security drastically– even a 51% attack on the sidechain can hardly transfer coins from the parent chain mistakenly. Transfers between two asynchronous systems without an intermediary to harmonize are complicated to cope with. Whether to wait for a reorganisation proof in the contest period, or to detect misbehaviors on the parent chain for the validators, additional overhead will be incurred. More troublesome are the responses to the scenario that the transfer is invalidated but the transferred coins are traded, which needs a reasonable design.

Sidechains extend functions of the parent chain, for instance, transferring Bitcoins to a sidechain that supports smart contract so that Bitcoins can be consumed as gas for contracts. They can also hold their own applications taking the load off the parent chain. In a certain sense, sidechains scale up the parent chain. Theoretically, a sidechain can peg different parent chains. In that case, the transferred coins from the parent chains should be marked with some identity to prevent malicious users from stealing valuable coins with the worthless. For different public chains, it is practical to build sidechains and bridge the sidechain. In this way, the messy interactions are totally constructed upon sidechains, meaning that public chains are impacted minimally.

D. Interoperability with the outside

This section doesn’t discuss cross-chain interoperability, but the ways how the outside world interacts with blockchains. Prior to blockchain, mass adopted web products and services have been developed. To avoid reinventing the wheel, leveraging existing standards to connect blockchains and extensive Internet services is necessary.

To illustrate this clearly, communicating with an Ethereum node over JSON-RPC is a good example. JSON (JavaScript Object Notation) is a lightweight text format, which is human-readable, easy to generate and parse[17]. JSON is independent of any specific programming language, making it an ideal data-interchange format. JSON-RPC is a stateless, light-weight and apparently JSON-based remote procedure call (RPC) protocol[18]. By defining the data structures explicitly, different types of nodes, cpp-ethereum, go-ethereum, py-ethereum, parity, can provide standardized methods[19]. Those JSON-RPC methods can be called over various transports like sockets and http. Web3.js is an Ethereum compatible JavaScript library that implements the JSON-RPC spec above[20]. With this library, a webpage can easily operate with Ethereum, like querying the states and sending transactions. There are also some other implementations in different languages. Not only Ethereum, almost all the blockchain platforms provide similar interoperating methods, so that blockchains are linked with the outside world effectively.

A trustless option to make smart contracts aware of real world data is to let the data provider cryptographically sign the data of interest. TLS-N[21] is such a solution that is compatible with TLS 1.3.

[17] http://json.org/

[18] https://www.jsonrpc.org/specification

[19] https://github.com/ethereum/wiki/wiki/JSON-RPC#json-rpc-api

[20] https://github.com/ethereum/web3.js/

[21] TLS-N: Non-repudiation over TLS Enabling Ubiquitous Content Signing for Disintermediation, Ritzdorf et al., NDSS 2018, https://eprint.iacr.org/2017/578.pdf

4. Sustainability

Sustainability is a notion introduced in the domain of environment[22]. It has been extended to almost every field. Albeit the technical means in the previous sections are unquestionably important to the development of blockchains, this topic goes far beyond pure technical realm. The balance and growth of an industry is always governed by a number of factors. In this chapter, opportunities and challenges concerning the sustainability of blockchains will be demonstrated from some other angles.

A. Energy Consumption of Proof of Work

Bitcoin is a typical representative to explain the "energy crisis" of PoW blockchains. As is well known, Bitcoin gives miners an incentive to maintain the network zealously by allowing the creator of a block to mint certain number of new coins owned by itself. Currently, every 10 minutes, a block creator wins a coinbase reward of 12.5 bitcoins, which is worth about $80,000[23], and a small part of transaction fees. Miners are racing to invest heavily in mining equipment to improve chances of profiting. Such a severe computing power competition leads to the huge consumption of electricity. It is estimated that the current electricity consumption of Bitcoin amounts to 22 terawatt-hours (TWh) per year, comparable with Ireland and more than 4 times what Google expended worldwide in 2015[24]. The number is still growing. It is unreasonable to say the energy is wasted. After all, the security of all the PoW blockchains is maintained in that way. But in the context that energy conservation is advocated globally, the mining of Bitcoin is highly controversial.

Regarding mining, there are some interesting phenomenon and viewpoints worth sketching. Most miners support the Bitcoin network to pursue economic benefits. It only makes sense when the Bitcoin price reaches a certain level that miner's lucre exceeds their investments in the real world. Since the fall in Bitcoin's price from the end of 2017, a number of mines were unable to afford operating expenses and closed down. Rational miners would flee at any time, in case they suffered a deficit in a long period. This is a huge peril that losing maintainers would totally ruin Bitcoin. To avoid inflation, Satoshi Nakamoto designed a mechanism when creating Bitcoin that the coinbase reward would be gradually reduced remaining the gross of Bitcoins at 21,000,000. In the future, miners will only relies on transaction fees to profit. Provided that bitcoin price couldn't meet their demands, they would increase the fees constantly, then bringing on reduction in trading volume. It would be a vicious circle that Bitcoin may finally shutdown. Another is the centralization issue. Due to high variance of block creation, miners tend to compose a mining pool for stable revenue.

[22] https://en.wikipedia.org/wiki/Sustainability
[23] Real time price of Bitcoin can be inquired in https://www.blockchain.com/explorer
[24] https://www.economist.com/the-economist-explains/2018/07/09/why-bitcoin-uses-so-much-energy

Several noted mining pools control a large share of the computing power[25], which runs against where Bitcoin starts from.

B. Governance

Governing a blockchain is a substantial obstacle. Even if a blockchain is permissionless, i.e. anyone can join and leave at any point in time, the core developers are making daily decisions on behalf of the other blockchain users. It's therefore appropriate to understand whether Bitcoin is actually decentralized[26].

5. Conclusion

Scalability of permissioned blockchains is relatively approachable, as they compromise the freedom of trust for scalability. Scalability of permissionless blockchains requires novel concepts, and the current progress looks very promising from a practical perspective. What's ultimately important is to properly lay out the security and trust assumptions of any given solution, coupled with appropriate security proofs. Lastly, we should understand why blockchain is required and a useful solution to a given problem[27].

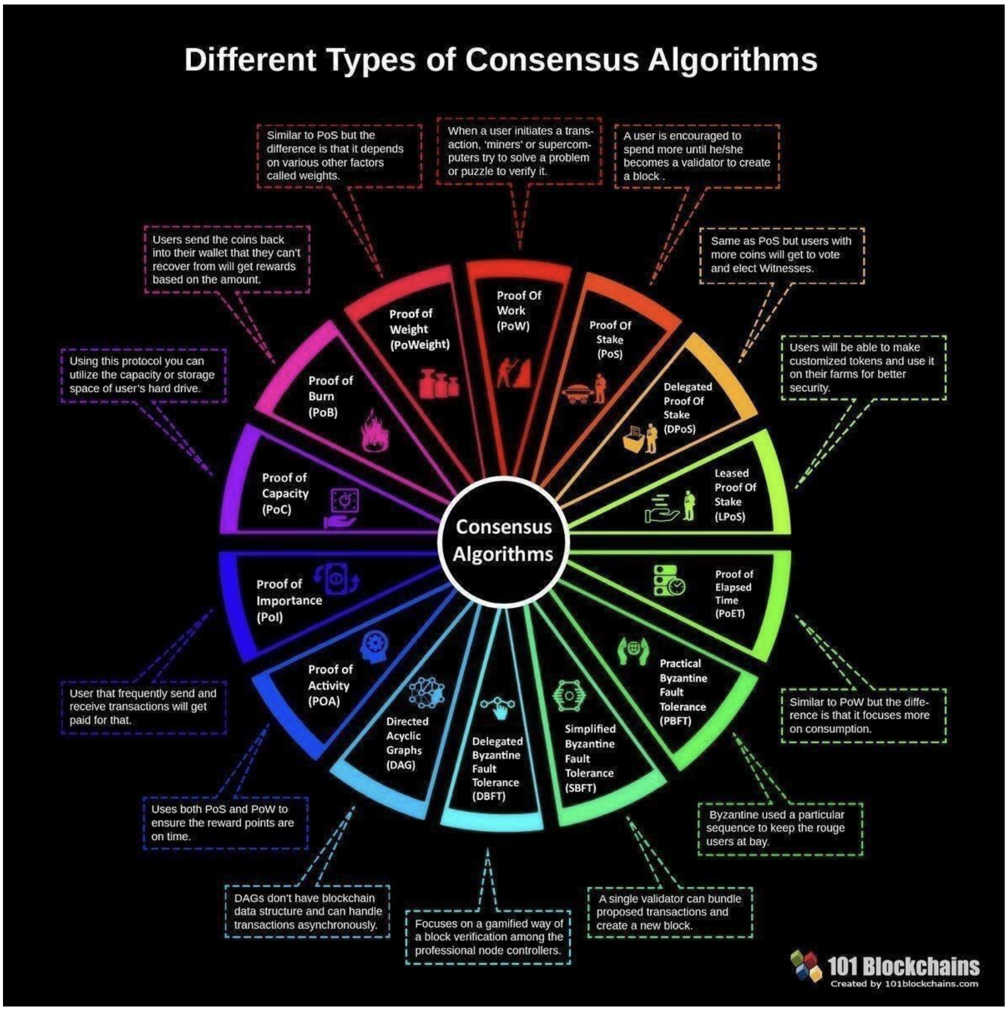
[25] https://www.blockchain.com/pools

[26] Is Bitcoin a Decentralized Currency?, Gervais et al., IEEE S&P Magazine 2014, https://eprint.iacr.org/2013/829.pdf

[27] Do you need a Blockchain?, Wüst and Gervais, Crypto Valley Conference 2018, https://eprint.iacr.org/2017/375.pdf

A. Appendix

Although Figure 3 lists different consensus algorithms, it is inappropriate to compare those directly as the different consensus algorithms have fundamentally different trust assumptions (e.g. permissioned vs. permissionless), might not be secure or not even practical.


Different Types of Consensus Algorithms
Consensus Algorithms
Proof Of Work (PoW)
Proof Of Stake (PoS)
Delegated Proof Of Stake (DPoS)
Leased Proof Of Stake (LPoS)
Proof of Elapsed Time (PoET)
Practical Byzantine Fault Tolerance (PBFT)
Simplified Byzantine Fault Tolerance (SBFT)
Delegated Byzantine Fault Tolerance (DBFT)
Directed Acyclic Graphs (DAG)
Proof of Activity (POA)
Proof of Importance (PoI)
Proof of Capacity (PoC)
Proof of Burn (PoB)
Proof of Weight (PoWeight)
101 Blockchains
Created by 101blockchains.com

When a user initiates a transaction, 'miners' or supercomputers try to solve a problem or puzzle to verify it.

A user is encouraged to spend more until he/she becomes a validator to create a block .

Same as PoS but users with more coins will get to vote and elect Witnesses.

Users will be able to make customized tokens and use it on their farms for better security.

Similar to PoW but the difference is that it focuses more on consumption.

Byzantine used a particular sequence to keep the rouge users at bay.

A single validator can bundle proposed transactions and create a new block.

Focuses on a gamified way of a block verification among the professional node controllers.

DAGs don't have blockchain data structure and can handle transactions asynchronously.

Uses both PoS and PoW to ensure the reward points are on time.

User that frequently send and receive transactions will get paid for that.

Using this protocol you can utilize the capacity or storage space of user's hard drive.

Users send the coins back into their wallet that they can't recover from will get rewards based on the amount.

Similar to PoS but the difference is that it depends on various other factors called weights.