Chapter 3
Varieties of Blockchains

Since the advent of Bitcoin in 2008, a diverse range of blockchains has emerged. Blockchain has a complex internal structure and has many configurations and variants. When building applications based on blockchains, we need to systematically consider the features and configurations of blockchains and assess their impact on quality attributes for the overall systems. Since blockchains are still at an early stage, there is little product data or reliable technology evaluation available to compare different blockchains. The lack of product data and reliable technology evaluation resources makes the comparison difficult.

In this chapter, we address the manifold varieties of blockchains by presenting a design taxonomy that defines dimensions and categories for classifying blockchains and ways of using them in systems. Taxonomies have been used in software architecture to understand existing technologies. The compact framework provided by a taxonomy allows architects to explore the conceptual design space and to compare and evaluate design options. Our taxonomy captures major architecturally relevant characteristics of various blockchains and indicates their support for various quality attributes. This includes performance and quality attributes of blockchain-based systems, as well as core concerns of blockchains like decentralization and the data structure used. The taxonomy is informed by existing industrial products, technical forums, academic literature, and our own experience of using blockchains and developing prototypes.

3.1 Fundamental Properties of Blockchain

If data is contained in a committed transaction, it will eventually become in practice *immutable*. The immutable chain of cryptographically signed historical transactions provides *non-repudiation* of the stored data. Cryptographic tools also support data *integrity*, the public access provides data *transparency*, and *equal rights* allows

X. Xu et al., *Architecture for Blockchain Applications*,
https://doi.org/10.1007/978-3-030-03035-3_3

every participant the same ability to access and manipulate the blockchain. These rights can be weighted by the compute power or stake owned by the miner. A distributed consensus mechanism governs addition of new items; it consists of the rules for validating and broadcasting transactions and blocks, resolving conflicts, and the incentive scheme. The consensus ensures all stored transactions are valid and that each valid transaction is added only once.

Trust in the blockchain is achieved from the interactions between nodes within the network. The participants of blockchain network rely on the blockchain network itself rather than relying on trusted third-party organizations to facilitate transactions. These five properties (immutability, non-repudiation, integrity, transparency, and equal rights) are the main properties supported in existing blockchains.

3.2 Decentralization

Decentralization is one of the distinguishing capabilities of blockchain technology, but there are various aspects and varieties of decentralization. Decentralization devolves responsibility and capability from a central location or authority. In a centralized system, all users rely on a central authority to mediate transactions. For example in a bank, customers rely on the bank's systems to correctly adjust their account balances when a bank transfer occurs. A central authority could manipulate the whole system, including by directly updating backend databases or by upgrading the software that implements the system. Thus, a central authority is a single point of failure for a centralized system. In contrast, a fully decentralized currency system like Bitcoin allows people to reach agreement on who owns what without having to trust each other or a separate third-party. Such a system is highly available since every full node in Bitcoin network downloads every block and transaction, checks them against Bitcoin's core consensus rules, and provides functionality to process transactions. There are currently more than 9000 nodes in the Bitcoin network,[1] although not all are full nodes that form the backbone of Bitcoin.

Table 3.1 represents a spectrum of (de)centralization, from full centralization to full decentralization. The column 'fundamental properties' refers to the five properties discussed in Section 3.1. In a system it is possible that some components or functions are decentralized while others are centralized.

There are two types of centralized systems. In the first there is a monopoly service provider, including governments and courts within a jurisdiction, and business monopolies. In the other type, there are competing alternative providers, such as banks, online payments, or cloud computing providers. Any centralized system is a single point of failure for its users. However, where there are alternative providers, the failure of a single service provider only affects its users. Users may switch providers or may be able to use multiple providers.

[1] https://bitnodes.21.co/nodes/.

Table 3.1 (De)centralization with an indication of their relative impact on quality properties (⊕, less favourable; ⊕⊕, neutral; ⊕⊕⊕, more favourable)

Design decision	Option	Impact			
		Fundamental properties	Cost efficiency	Performance	#Failure points
Fully centralized	Services with a single provider (e.g. governments, courts)	⊕	⊕⊕⊕	⊕⊕⊕	1
	Services with alternative providers (e.g. banking, online payments, cloud services)				
Partially centralized and partially decentralized	Permissioned blockchain with permissions for fine-grained operations on the transaction level (e.g. permission to create assets)	⊕⊕	⊕⊕	⊕⊕	*
	Permissioned blockchain with permissioned miners (write), but permission-less normal nodes (read)				
	Off-chain transaction protocols				
Fully decentralized	Permission-less blockchain	⊕⊕⊕	⊕	⊕	Majority (nodes, power, stake)

At the other end of the spectrum, fully decentralized systems include permission-less public blockchains, such as Bitcoin and Ethereum. Permission-less public blockchains are completely open: new users can at any time join the network, validate transactions, and mine blocks. Decentralized systems using anonymous validators need to protect against *Sybil attacks*, where attackers create many hostile anonymous nodes. Bitcoin partly guards against this through its proof-of-work mechanism, so that it is not the total number of nodes that is important for integrity but rather the total amount of computational power. While it is easy for an attacker to create anonymous nodes, it is not easy for them to amass large amounts of computational power. Any system can be defeated if an attacker controls a majority of authority (nodes, computational power, or stakeholding). Game-theoretic attacks can change this threshold, requiring a higher (e.g. 66%) majority to maintain integrity. There is a spectrum of possibilities between centralization and decentralization. There are two dimensions to classify a blockchain, including *permission* and the type of *deployment*. These two dimensions are discussed in the next two subsections.

Another hybrid approach is the use of off-chain transaction protocols to progress transactions between parties and then later to reconcile the effects of those protocol executions on-chain. The Bitcoin *Lightning network*[2] moves some transactions off-chain by establishing a multi-signature transaction between two participants as a micropayment channel to transfer value off-chain. Once both sides wish to close the micropayment channel and finalize the value transfer, a transaction is submitted to the global Bitcoin blockchain. Such bidirectional channels can be connected to establish a payment network leveraging Bitcoin. The intermediate transactions occurring in the payment channel are not included in the blockchain. Raiden[3] is a similar project on Ethereum, using its smart contract facilities.

3.2.1 Permission

Instead of anonymous public participation, a blockchain may be permissioned in requiring that one or more authorities act as a gate for participation. This may include permission to join the network (and thus read information from the blockchain), permission to initiate transactions, or permission to mine. Some permissioned blockchains, e.g. MultiChain,[4] allow more fine-grained permissions, such as the permission to create assets. Permissioned blockchain networks include Ripple[5] and Eris.[6] The code for public blockchains can also be deployed on private

[2] https://lightning.network/.

[3] https://github.com/raiden-network/raiden.

[4] http://www.multichain.com/.

[5] https://ripple.com/.

[6] https://monax.io.

networks to create a kind of permissioned blockchain using network access controls. Permission information can be stored either on-chain or off-chain.

Permissioned blockchains may be especially suitable in regulated industries. For example, banks are required to establish the real-world identity of transacting parties to satisfy Know Your Customer (KYC) regulation. In contrast, a transaction on a permission-less blockchain across jurisdictional boundaries can circumvent this and undermine regulatory controls. Permissioned blockchains may be able to better control access to off-chain information about real-world assets.

There are often trade-offs between permissioned and permission-less blockchains including transaction processing rate, cost, censorship resistance, reversibility, finality, and the flexibility in changing and optimizing the network rules. The suitability of a permissioned blockchain may also depend on the size of the network. Nonetheless, the permission management mechanism may itself become a potential single point of failure, not just operationally but also from a business perspective.

3.2.2 Deployment

When using a blockchain, there are different types of deployments, including public blockchain, consortium/community blockchain, or private blockchain. An overview is given in Table 3.2.

Most digital currencies use public blockchains, which can be accessed by anyone on the Internet. Using a public blockchain results in better information transparency and auditability but sacrifices performance and has a different cost model. In a public blockchain, data privacy relies on encryption or cryptographic hashes.

A consortium blockchain is typically used across multiple organizations. The consensus process in a consortium blockchain is controlled by pre-authorized nodes. The right to read the blockchain may be public or may be restricted to specific participants. In a private blockchain network, write permissions are often kept within one organization, although this may include multiple divisions of a single organization.[7]

Whether using a consortium blockchain, private blockchain, or permissioned public blockchain,[8] a permission management component will be required to authorize participants within the network. Private blockchains are the most flexible for configuration because the network is governed and hosted by a single organization. Many blockchain platforms support deployment as consortium blockchains or private blockchains, e.g. MultiChain and Eris.

[7] There is a grey area between consortium blockchains and private blockchains, and the differences may be more administrative than technical. Nonetheless we distinguish them here because at their extremes they have architectural differences.

[8] Ripple can arguably be seen as a permissioned public blockchain.

Table 3.2 Blockchain deployment (⊕, less favourable; ⊕⊕, neutral; ⊕⊕⊕, more favourable)

	Impact			
Deployment option	Fundamental properties	Cost efficiency	Performance	Flexibility
Public blockchain	⊕⊕⊕	⊕	⊕	⊕
Consortium/community blockchain	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Private blockchain	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

Table 3.3 Ledger structure (⊕, less favourable; ⊕⊕, neutral; ⊕⊕⊕, more favourable)

	Impact			
Option	Fundamental properties	Cost efficiency	Performance	Flexibility
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Restricted shared ledgers (Corda)	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

3.3 Ledger Structure

The ledger can be structured in different ways; Table 3.3 provides an overview. In Bitcoin, the history of all transactions is captured in the blockchain structure. This is a single global list (chain) of lists (blocks) of transactions, as discussed in Chapters 1 and 2. Bitcoin nodes actually record the blockchain as a tree of blocks, where shorter branches attached to the main chain represent alternative competing histories. However, the tree data structure is relevant mainly for the nodes operating the blockchain and determining consensus; under the logical view from a user's perspective, the blockchain is a list of blocks. This is similar for Ethereum.

Other blockchain and distributed ledger systems have different data structures. For example, the logical view of transactions recorded in Hashgraph[9] is based on a directed acyclic graph (DAG) of blocks, rather than a list. Somewhat similarly, IOTA[10] also uses a DAG but of individual transactions rather than blocks of transactions.

These systems all maintain a single global transaction history. Other distributed ledger systems such as Hyperledger Fabric and Corda have been proposed where there are essentially many small ledgers, shared only between parties of interest

[9] https://www.hederahashgraph.com/.

[10] https://www.iota.org/.

who are authorized to view the transactions recorded in those ledgers. For the Corda distributed ledger, the abstract logical view of transaction history is of a global graph of transactions. However, transactions are only distributed to parties of interest; special agents (notaries) can be used to further limit the distribution of transactions while attesting to the integrity of unseen parts of the transaction graph. So although there is notionally a global graph of transactions, the view that most parties see is a collection of small ledgers, each shared with their related business contacts. Hyperledger Fabric is somewhat similar, because parties also see a collection of small ledgers shared with related business contacts (via 'channels'). However, Fabric has a more rigid transaction distribution policy, isolating transactions within the channels.

3.4 Consensus Protocol

The choice of *consensus protocol* impacts security and scalability. An overview is given in Table 3.4. Once a new block is generated by a miner, the miner propagates the block to its connected peers in the blockchain network. However, miners may encounter different competing new blocks and resolve this using the blockchain's consensus mechanisms. Usually the approach is fixed for a particular blockchain; but Hyperledger Fabric deviates from this norm, as a framework with a modular architecture that caters for pluggable implementations of various consensus protocols.

The typical overall approach is called Nakamoto consensus, as introduced in Section 2.1.5. This relies on participants selecting as authoritative the longest chain of blocks they have observed at every point in time. In Bitcoin, new blocks are generated through a *proof-of-work* mechanism. Proof-of-work uses a cryptographic puzzle which is easy to verify, but solving it is difficult and takes effectively random

Table 3.4 Consensus protocol (⊕, less favourable; ⊕⊕, neutral; ⊕⊕⊕, more favourable)

Option		Impact			
		Fundamental properties	Cost efficiency	Performance	Flexibility
Security-wise	Proof-of-work	⊕⊕⊕	⊕	⊕	⊕
	Proof-of-retrievability	⊕⊕⊕	⊕	⊕	⊕
	Proof-of-stake	⊕⊕	⊕⊕	⊕⊕	⊕⊕⊕
	Practical Byzantine Fault Tolerance (PBFT)	⊕	⊕⊕⊕	⊕⊕⊕	⊕
Scalability-wise	Bitcoin-NG	⊕⊕⊕	⊕	⊕	⊕
	RBBC	⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕

time. Bitcoin miners compete to solve such a puzzle for each block, using large amounts of computer power (and hence electricity) to increase their chances of winning the competition for the block. The investment required by miners for this acts to align their incentives with the good operation of the overall system. There are various proof-of-work mechanisms, such as Ethash[11] used by Ethereum and Hashcash[12] used by Bitcoin. The work done in proof-of-work systems can sometimes be put to good use. For example, the mechanism in Primecoin[13] generates prime number chains which are of interest to mathematical research. Permacoin uses 'proof-of-retrievability' to repurpose Bitcoin's mining resources to distributed storage of archival data.

Proof-of-stake is an alternative mechanism for Nakamoto consensus, which selects the next mining node based on the control of the native digital currency of the blockchain network. For example, the miners in Peercoin[14] need to prove the ownership of a certain amount of Peercoin currency to mine blocks. Thus, proof-of-stake naturally aligns the incentives of digital currency holders in the blockchain with the good operation of the blockchain. There are various proof-of-stake protocols, e.g. Tendermint[15] used in Eris and Casper[16] for Ethereum. These have different design goals, favouring some non-functional properties over others. Proof-of-stake does not necessarily select the next miner based on largest stakeholding, e.g. Nxt[17] also uses a random factor, and Peercoin combines randomization and coin age. BitShares[18] uses *delegated proof-of-stake*, where the accounts may delegate their stake to other accounts, rather than participating in the process of validating transactions directly. The representatives take turns in a round-robin manner, signing blocks. Compared with proof-of-work, proof-of-stake is more cost-efficient because much less computational power is used in mining and latency is also shorter. However, passive holding of assets may become harder.

The *Practical Byzantine Fault Tolerance (PBFT)* protocol has been applied for consensus in permissioned blockchains, e.g. in Stellar.[19] PBFT ensures consensus despite arbitrary behaviour from some fraction of participants. Compared to Nakamoto consensus, it is a more conventional approach within distributed systems. Roughly speaking, PBFT-based blockchains offer a much stronger consistency guarantee and lower latency but for a smaller number of participants. The core of Tendermint is also a PBFT protocol but uses a proof-of-stake mechanism to prevent Sybil attacks. PBFT requires that all participants must agree on the list of

[11] https://github.com/ethereum/wiki/wiki/Ethash.

[12] https://en.bitcoin.it/wiki/Hashcash.

[13] http://primecoin.io/.

[14] http://peercoin.net/.

[15] http://tendermint.com/.

[16] https://github.com/ethereum/casper/.

[17] https://nxt.org/.

[18] https://bitshares.org/.

[19] https://www.stellar.org/.

participants in the network. Thus, the protocol is normally only used in permissioned blockchains.

Some new protocols have been proposed to improve *scalability*. Bitcoin-NG decouples Bitcoin's operation into two planes: leader election and transaction serialization. Once a leader is selected, it is entitled to serialize transactions until the next leader is selected. Thus, the leader election in Bitcoin-NG is forward-looking and ensures that the system is able to continually process transactions. Another new protocol is used in the Red Belly Blockchain (RBBC). This algorithm is a kind of democratic Byzantine consensus approach in not requiring leader nodes. The approach starts with submitted transactions being collected by a set of proposers. These nodes collectively decide on a proposed set of transaction to send to a verifier nodes, who enforce consensus using hashes exchanged for the proposed sets of transactions.

3.5 Block Configuration

Block configuration concerns options for the size (number/complexity of transactions) allowed in blocks and the frequency by which blocks are generated. These choices can impact *scalability* in terms of transaction processing rate. An overview is given in Table 3.5.

One configuration change would be to adjust mining difficulty to shorten the time required to generate a block, thus reducing latency and increasing throughput. However, a shorter inter-block time would lead to an increased frequency of forks. Ethereum has a much shorter inter-block time (10–20 s) than Bitcoin, while still using Nakamoto consensus and proof-of-work. The increased frequency of forks ('uncle blocks' in Ethereum's terminology) leads to users waiting for more confirmation blocks than in Bitcoin, though still achieving overall lower transaction latency.

Another important block configuration parameter concerns block size. Depending on the blockchain used, this is specified differently, e.g. as block size limit in Bitcoin (data size in MB) or as block gas limit in Ethereum (limiting the complexity of the contained transactions). For example, there are some proposals for Bitcoin to increase its block size from 1 to 8 MB, to include more transactions into a block and

Table 3.5 Block configuration (⊕, less favourable; ⊕⊕, more favourable)

	Impact			
Option	Fundamental properties	Cost efficiency	Performance	Flexibility
Original block size and frequency	⊕⊕	n/a	⊕	n/a
Increase block size/decrease mining time	⊕	n/a	⊕⊕	n/a

thus increase maximum throughput. The decision on the size of blocks is subject to a trade-off between speed of replication, inter-block time, and throughput and works as follows. When a new block has been proposed, processing nodes need to select a set of transactions from the transaction pool/mempool and validate and execute those. This cannot be done before observing the latest block, because the state changed as a result of the new block and may render some transactions invalid or alter their effects. Once that is complete, the block can be formed, and, in the case of proof-of-work consensus, mining can start. On the one hand, if the block can be too big or too complex, transaction processing may take too much time. Take the extreme example of having no limit; then, the system could be subject to a DoS attack by flooding it with transactions, such that the inter-block time would rise to unacceptable levels. Very big blocks also take longer to replicate among the full nodes. On the other hand, high limits can result in higher throughput. For these reasons, block limits should be set with care in private and permissioned networks. On the public Bitcoin blockchain, the long-time limit of 1 MB sparked significant controversy[20] and led to an effective increase to 2–4 MB. Public Ethereum's block gas limit has changed a number of times (see also Section 11.6.2) and is about eight million gas at the time of writing.[21] On public proof-of-work blockchains, high block limits also increase the risk of empty blocks. Consider the case where miner *A* tries to include many transactions and miner *B* tries to mine empty blocks. While *A* is processing transactions, *B* is already working on its proof-of-work, thereby increasing its relative chances to find a new block first. If block limits and block mining rewards are high, it might actually be economical to mine as many empty blocks as possible. Unfortunately, that also deteriorates the value of the network, because now it does not process new transactions anymore.

3.6 Auxiliary Blockchains

When building and deploying a new blockchain, it might be combined with or built on an existing blockchain, thus forming an *auxiliary blockchain*. Different strategies can be used to achieve security and scalability. An overview is given in Table 3.6.

For *security*, the new blockchain can be aligned with public blockchains, utilizing existing infrastructure, resources, and trust. The first option is *merged mining*, which reuses the mining power of an existing public blockchain to mine and secure the new blockchain. In this case, a proof-of-work found by a miner of the public blockchain is used by both blockchains. First, the miner produces a transaction set for both blockchains. The hash of the block produced for the new blockchain is added to the public blockchain. Then, once the miner finds a proof-of-work solution at the difficulty level of either blockchain, the proof-of-work is combined with the

[20] https://en.bitcoin.it/wiki/Block_size_limit_controversy.

[21] https://etherscan.io/chart/gaslimit.

Table 3.6 Auxiliary blockchains (⊕, less favourable; ⊕⊕, neutral; ⊕⊕⊕, more favourable)

Option		Impact			
		Fundamental properties	Cost efficiency	Performance	Flexibility
Security-wise	Merged mining	⊕⊕⊕	⊕⊕	⊕	⊕
	Hook into popular blockchain at transaction level	⊕⊕	⊕	⊕⊕	⊕⊕⊕
	Proof-of-burn	⊕	⊕	⊕⊕⊕	⊕⊕
Scalability-wise	Sidechains	⊕⊕⊕	⊕	⊕	⊕
	Multiple private blockchains	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕
	Mini-blockchain	⊕⊕	⊕⊕	⊕	⊕⊕

transaction set and submitted to the corresponding blockchain. Namecoin is the first blockchain that uses merged mining with the Bitcoin blockchain. Merged mining reuses an established blockchain network. It might be difficult initially to persuade the miners of an existing blockchain to join a new blockchain network.

A more loosely coupled way is to *hook* the new blockchain into a public blockchain, by periodically adding hashes of the new blockchain to transactions of the public blockchain. For instance, Factom[22] anchors into the Bitcoin blockchain by submitting a transaction to the Bitcoin blockchain every 10 min, with the current hash of the Factom blockchain.

The third option is *proof-of-burn*. The purpose of proof-of-burn is to verifiably destroy tokens on the existing chain rather than minting new tokens on the new chain. To 'transfer' tokens from a public blockchain to the new blockchain, the participants need to provide proof that their tokens were sent to a verifiably unspendable address. The burnt tokens, originally mined by proof-of-work, represent the corresponding computational power. Proof-of-burn can be used for bootstrapping a new cryptocurrency, e.g. Counterparty,[23] as it ensures serious commitment.

Auxiliary blockchains can also be used to improve *scalability*. Rather than using a unique chain to record all types of transactions, multiple blockchains can be used to isolate information of separate concerns and with different characteristics and therefore improve scalability. Different mechanisms have been proposed to support interaction across multiple blockchains. One of the mechanisms is to use an off-chain hash lock. In the Bitcoin ecosystem, using a hash lock with *contracts* can enable *atomic cross-chain trading*,[24] which allows one cryptocurrency (e.g. the Bitcoin cryptocurrency, BTC) to be traded for another cryptocurrency (e.g.

[22] http://factom.org/.

[23] http://counterparty.io/.

[24] https://en.bitcoin.it/wiki/Atomic_cross-chain_trading.

tokens on a Bitcoin sidechain). This mechanism is also applicable in the Ethereum ecosystem.[25]

The first option for scalability is to use sidechains. Sidechaining is a mechanism that allows tokens of one blockchain to be securely transferred and used in another blockchain; eventually, they can be moved back to the original chain securely. The original chain is called *main chain*, and the one that accepts the tokens from the original chain is called *sidechain*. The second option is to have multiple private chains, where each of the private chains could link with a public blockchain. With sidechains, there is a layer of separation between two blockchains, which means that the main chain can be protected from issues or damages on the sidechains. Sidechains can help to build a blockchain ecosystem based on a popular main blockchain, without significantly increasing the load on the main chain. However, the clients of sidechains may become complex, because they typically need to be able to process transactions from the main chain and the sidechain.

There are two ways of sidechaining: unilaterally pegged sidechain and bilaterally pegged sidechain. For a unilateral (or one-way) peg, the interaction is only from the main chain to the sidechain, e.g. through proof-of-burn. For a bilateral peg, the interaction is bidirectional. One mechanism to secure bilateral pegged sidechains is essentially a voting system, where a group of custodians cast votes on when to lock and unlock tokens on one blockchain and where to send tokens on the other blockchain. The first option is to have an exchange holding the locked tokens from one blockchain and the unlocked equivalent tokens from the other blockchain. The exchange would locally enforce the promise of locking the tokens from one blockchain before unlocking the tokens of the other blockchain. This design introduces a central trusted third-party to control the exchange. A better option is to have a group of notaries control a multi-signature wallet, where a majority has to approve unlocking tokens. This is more decentralized than the first option but still centralizes control to a degree. To achieve better decentralization, the notaries could be from different jurisdictions and geographies with good reputation and good security.

The full nodes of most blockchain networks need to keep all historical transactions and the state of blockchain network, which requires sizeable storage space. For example, Bitcoin and Ethereum require more than 200 GB[26] and 600 GB[27] of storage space, respectively, at the time of writing, and these sizes keep growing. To reduce the storage burden of blockchain participants and address other scalability concerns, applying the concept of *sharding* to blockchain has been proposed. Sharding means to divide the state of blockchain into pieces. The participating blockchain nodes only hold data of some shards instead of the complete blockchain data structure. There are two types of sharding, including transaction sharding and

[25] https://dappsforbeginners.wordpress.com/tutorials/two-party-contracts/.

[26] https://bitnodes.earn.com/dashboard/bitcoind/.

[27] https://bitinfocharts.com/ethereum/.

state sharding. Elastico and Zilliqa[28] support transaction sharding. Ethereum 2.0[29] plans to improve scalability of its public blockchain through sharding based on structuring the network into two layers.

Instead of keeping all transaction information, a *mini-blockchain* scheme proposed by Cryptonite[30] periodically forgets old transaction history. The Cryptonite network maintains an account tree that holds the balance of all addresses and a separate proof chain that stores all the historical block headers. The account tree is updated according to the transactions, and after a period of time, the transactions are forgotten by the network. Neither off-chain transactions nor the mini-blockchain stores all the transactions on the blockchain. Thus, both sacrifice the fundamental properties of blockchain. The mini-blockchain saves space by forgetting historical transactions, but its performance is not necessarily better because the consensus mechanism is still the same.

3.7 Anonymity

Although the Bitcoin blockchain is perceived to be anonymous, research has shown that Bitcoin transactions can be linked to compromise the anonymity of Bitcoin users. Different techniques have been proposed to preserve anonymity on blockchain. Zcash,[31] also called Zerocash or Zerocoin, encrypts the payment information in the transactions and uses a cryptographic method to verify the validity of the encrypted transactions. A zero-knowledge proof construction is used to allow the blockchain network to maintain a secure ledger and enable private payment without disclosing the parties or amounts involved.

Mixing services offer an alternative method for anonymization. A mixing service groups several transactions together so that a payment contains multiple input addresses and multiple output addresses. Anonymity is preserved because it is hard to track which output address is paid by which input address. To further improve the way that mixing service operates, a series of mixing services can be linked sequentially. If the mixed transactions are uniform in value, the traceability between input and output addresses is minimized. Uniform values can be achieved by using standardized denominations, similar to bank notes and coins in traditional cash. A centralized mixing service requires a third-party to operate, e.g. CoinJoin[32] and Blindcoin. Distributed mixing services, on the other hand, do not rely on a single third-party, e.g. CoinSwap.[33] Some blockchains have a kind of native, built-in

[28] https://zilliqa.com/.

[29] https://github.com/ethereum/wiki/wiki/Sharding-FAQs.

[30] http://cryptonite.info/.

[31] https://z.cash/.

[32] https://bitcointalk.org/index.php?topic=279249.0.

[33] https://bitcointalk.org/index.php?topic=321228.0.

mixing service, including Dash and Monero. Dash pre-anonymizes funds of users through mixing rounds, so that the funds can later be spent without delay.[34] In contrast, Monero uses ring signatures, such that the sender of a transaction cannot be identified among a group of possible senders.

3.8 Incentives

Blockchains and their applications (especially on public blockchains) introduce financial incentives in the cryptocurrencies of the respective networks. Incentives are paid to make miners to join the network, validate transactions, generate blocks, and (where applicable) execute smart contract functions correctly. For example, in Bitcoin, miners have two incentives: the reward for generating new blocks and the fees associated with transactions. Miners in Ethereum also charge a fee to execute smart contracts. Enigma[35] has a fixed price for storage, data retrieval, and computation within the network. Enigma also requires a security deposit for nodes to join the network. If a node is found to lie, its deposit will be split among the honest nodes.

3.9 Summary

Blockchain platforms can have various configurations and design options. Using blockchain in different scenarios requires the comparison of blockchain options and products with different implementations and configurations. In this chapter, we discussed a taxonomy of blockchain systems. The taxonomy can be used when comparing blockchains and assist in the design and evaluation of software architectures using blockchain technology. Our taxonomy captures major architectural characteristics of blockchains and the impact of different decision decisions. This taxonomy is intended to help with important architectural considerations about the performance and quality attributes (e.g. availability, security, and performance) of blockchain-based systems.

3.10 Further Reading

This chapter is partly based on our earlier works (Xu et al. 2017).

Taxonomies have long been used in the software architecture community to understand existing technologies (see, e.g. Mehta et al. 2000; Gorton et al. 2015).

[34] https://docs.dash.org/en/latest/introduction/features.html#privatesend.

[35] https://www.media.mit.edu/projects/enigma/overview/.

From a software architecture perspective, blockchain can also be characterized as a software connector (Xu et al. 2016), which has a complex internal structure and many configurations and variants. Blockchain is a decentralized system that can be defeated unless there is a majority of honest or favourable authority (computational power, stakeholding, etc., depending on the consensus mechanism). Eyal and Sirer (2018) show that game-theoretic attacks can change this threshold for proof-of-work, requiring a higher (e.g. 66%) majority to maintain integrity and prevent double-spending attacks. More definitions of different types of blockchain and discussion on the trade-offs between them can be found in Swanson (2015) and Buterin (2015).

Nakamoto consensus provides probabilistic immutability. There is always a chance that the most recent few blocks get replaced by a competing chain fork. The impact of inter-block time on the frequency of forks is discussed in Decker and Wattenhofer (2013). A detailed comparison between proof-of-work and proof-of-stake can be found in Gervais et al. (2016). Permacoin's 'proof-of-retrievability' is discussed in Miller et al. (2014). Discussion on PBFT-based blockchains can be found in Vukolić (2015). The Red Belly Blockchain (Crain et al. 2017) uses a new kind of democratic Byzantine consensus protocol. Some protocols have been proposed to improve scalability, for example, Bitcoin-NG (Eyal et al. 2016) and the Bitcoin Lightning network (Poon and Dryja 2016).

More information on sidechaining can be found in Back et al. (2014). Blockchains that apply sharding technology are discussed in Luu et al. (2016) and Danezis and Meiklegohn (2016).

Detail of Blindcoin can be found in Valenta and Rowan (2015).