Chapter 1
Introduction

1.1 What Is Blockchain and Why Should I Care?

Blockchains are an emerging digital technology that combine cryptography, data management, networking, and incentive mechanisms to support the checking, execution, and recording of transactions between parties. A blockchain ledger is a list ('chain') of groups ('blocks') of transactions. Parties proposing a transaction may add it to a pool of transactions intended to be recorded on the ledger. Processing nodes within the blockchain system take some of those transactions, check their integrity, and record them in new blocks on the ledger. The contents of the blockchain ledger are replicated across many geographically-distributed processing nodes. These processing nodes jointly operate the blockchain system, without the central control of any single trusted third-party. Nonetheless, the blockchain system ensures that all nodes eventually achieve consensus about the integrity and shared contents of the blockchain ledger.

Transactions between parties such as payments, escrow, notarization, voting, registration, and process coordination are key in the operations of government and industry. Traditionally, these transactions are supported by trusted third-parties such as government agencies, banks, legal firms, accounting firms, and service providers in specific industries. Blockchains provide a different way to support these transactions. Instead of trusting third-parties, we would trust the collective jointly operating the blockchain and the correctness of their shared technology platform.

Blockchain technology was originally used for the Bitcoin digital currency, but blockchains are now being implemented in many other platforms and used for many other purposes. Just like a traditional database, a blockchain can in principle be used to represent transactions or information in any kind of application domain. But blockchains are different from traditional databases in important ways. These differences impact the design of systems that use blockchain.

X. Xu et al., *Architecture for Blockchain Applications*,
https://doi.org/10.1007/978-3-030-03035-3_1

The successful operation of a blockchain system relies on several key elements, including:

- Appropriate integrity criteria to be checked for each transaction and block
- The correctness of the system's software and technical protocols
- Strong cryptographic mechanisms to identify parties and check their authority to add new transactions
- A suite of incentive mechanisms to motivate processing nodes to participate in the community and to behave honestly, in their interests

For the software architect and engineer, blockchains are exciting because they can be used as a new foundation for re-imagining systems. They form a neutral infrastructure for processing transactions and executing programs. That is of potential interest for innovation at all touch-points between organizations or individuals. As such, *blockchain applications have the potential to disrupt the fabric of society, industry, and government*. Blockchains can also be used as a technology platform to handle some of the hard issues of data replication and system state synchronization with high integrity.

A Non-technical Explanation of Blockchain by Analogy

Imagine that a group of people, say the population of your community, want to introduce a special community currency. Let's call this currency the Community Dollar C$, which will be a noncash virtual currency. Initially everyone gets C$ 100, and everyone starts a physical ledger book where they note these holdings. The goal is to keep track of C$ ownership in all these ledger books, by ensuring the ledger books of everyone contain the same information.

Say, person *A* wants to pay C$ 50 to person *B*. Therefore, *A* asks everyone in the community to add that transaction to their ledger. Everyone checks if A has the money and signed the transfer order. If so, the transaction is added to the ledger. This results in an updated state where *A* has C$ 50 and *B* owns C$ 150.

Now the Community Dollar is starting to become popular, and many people use it. We start by grouping transactions onto paper pages, and rather than agreeing on each transaction individually, the whole community needs to find agreement which page to include. (Pages correspond to blocks in the blockchain.) Everyone still checks every transaction. To ensure that no one claims a transaction did not happen, we introduce cryptographic hashes that make sure no one can go back on the agreed set of ledger pages and the transactions on them. Assume we also have an incentive mechanism that encourages community members to stay honest and process transactions. That is needed to make such a decentralized system work. All of this can then happen without a trusted third-party, purely operated by members of the community.

1.1.1 *Defining Blockchain*

Before delving further into the details of the technology, we first define the main concepts. Blockchains maintain a ledger and implement a specific kind of distributed ledger technology.

Definition 1 (Distributed Ledger) A *distributed ledger* is an append-only store of transactions which is distributed across many machines.

Being 'append-only' is important: new transactions can be added, but old transactions cannot be deleted or modified. A new transaction might reverse a previous transaction, but both of them remain part of the ledger to allow auditability and ensure long-lasting integrity. We define the *concept* of a blockchain as follows.

Definition 2 (Blockchain) A *blockchain* is a distributed ledger that is structured into a linked list of *blocks*. Each block contains an ordered set of transactions. Typical solutions use cryptographic hashes to secure the link from a block to its predecessor.

A graphical representation of this concept is shown in Fig. 1.1. Cryptographic hashes ensure that a previous block cannot be changed. If the previous block was changed, its new hash would not match the originally recorded hash, so the link between the two blocks would break. We explain this mechanism in more detail in the next chapter, where we discuss specific blockchain platforms.

Some ingredients are necessary for the blockchain concept to work in practice as a system.

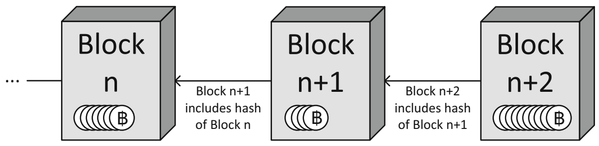
...
Block n
Block n+1 includes hash of Block n
Block n+1
Block n+2 includes hash of Block n+1
Block n+2

Fig. 1.1 Blockchain data structure

Definition 3 (Blockchain System) A *blockchain system* consists of:

(i) a *blockchain network* of machines, also called *nodes*;
(ii) a blockchain data structure, for the ledger that is replicated across the blockchain network. Nodes that hold a full replica of this ledger are referred to as *full nodes*;
(iii) a network *protocol* that defines rights, responsibilities, and means of communication, verification, validation, and consensus across the nodes in the network. This includes ensuring authorization and authentication of new transactions, mechanisms for appending new blocks, incentive mechanisms (if needed), and similar aspects.

For the verification of transactions, consider the example of Alice spending 2 Bitcoin (BTC), by transferring them to Bob. The system needs to ensure that the party initiating the transaction has Alice's authority and that Alice has the 2 Bitcoin available.

The above definition is still relatively broad and can capture blockchains of various sizes, degrees of openness, for various purposes, etc. The most well-known blockchains are Bitcoin and Ethereum, which are *public blockchains*.

Definition 4 (Public Blockchain) A public blockchain is a blockchain system that has the following characteristics:

(i) it has an *open network* where nodes can join and leave as they please without requiring permission from anyone;
(ii) all full nodes in the network can *verify* each new piece of data added to the data structure, including blocks, transactions, and effects of transactions; and
(iii) its protocol includes an *incentive mechanism* that aims to ensure the correct operation of the blockchain system including that valid transactions are processed and included in the ledger and that invalid transactions are rejected.

Public blockchains are often open leaderless peer-to-peer systems that manage the ownership of assets of value. Examples of such assets on Bitcoin and Ethereum blockchains are Bitcoin (BTC) and Ether (ETH) cryptocurrencies and digital tokens. In a public blockchain, there is not a high degree of trust in information from other nodes. Therefore, all full nodes verify everything, to reduce the risk of integrity violations jeopardizing the value of their own work. While this leads to redundant computation across the network, it is a direct consequence of the community of nodes collectively safeguarding the integrity of the blockchain.

In other settings, for example, within a large enterprise or in a consortium of companies, all blockchain nodes might be known and governed by other organizational or contractual mechanisms. These applications can be served by adopting a more relaxed trust assumption.

Finally, we define the term blockchain platform, which refers to the software used to run a blockchain.

Definition 5 (Blockchain Platform) A *blockchain platform* is the technology needed to operate a blockchain. This comprises the blockchain client software for processing nodes, the local data store for nodes, and any alternative clients to access the blockchain network.

Note that any blockchain platform must have client software with which processing nodes can operate the network, including for transaction propagation and block creation. Light clients may additionally exist, e.g. to enable mobile devices to read and write transactions to the network; these typically do not hold a full copy of the blockchain data structure. Alternative clients, both for processing and light nodes, may exist, particularly if the protocol is specified well.

1.1.2 Smart Contracts and Decentralized Applications

The transactions stored on a blockchain can be more than simple records of the exchange of assets—emerging blockchain systems also allow computer programs to be stored and to execute as part of transactions on the ledger. These are often called 'smart contracts', although the programs are typically not very smart and are often not related to legal contracts.

Definition 6 (Smart Contract) *Smart contracts* are programs deployed as data in the blockchain ledger and executed in transactions on the blockchain. Smart contracts can hold and transfer digital assets managed by the blockchain and can invoke other smart contracts stored on the blockchain. Smart contract code is deterministic and immutable once deployed.

The Bitcoin blockchain allows only very simple forms of smart contracts, but other blockchains such as Ethereum allow computer programs to be written in a 'Turing complete' language, that is, in principle, as expressive as every other general purpose programming language. As a result, blockchains can be more than a simple distributed database—they can be *general computational platforms*—albeit currently with severe practical limitations on computational complexity. This capability significantly expands the power of blockchain systems and increases their range of use and potential for innovation.

Smart contracts can be used to administer the ownership of assets represented by the blockchain cryptocurrency or by digital token implementations using a smart contract—more on that below. Although smart contracts are not always used for legal contracts, they can sometimes be used to automate or monitor the execution of parts of legal contracts. Smart contracts can also implement games, bets, or lotteries. They can also define a protocol of interaction between different parties, like in a collaborative business process across companies, and can support many more use cases. Throughout the book, you will find many applications of blockchain that are only possible due to the smart contract capability.

Applications can be designed to provide their main functionality through smart contracts. Such applications are called *decentralized applications* or *dapps*, and we will discuss them in more detail in Section 2.2.5. *Tokenvote*, the system described in Chapter 13, is an example of a dapp. In this book, we generally talk about *blockchain-based applications*, i.e. applications that make significant use of blockchain. This includes dapps but is not limited to them—significant portions of such applications can be based on traditional systems.

1.1.3 Cryptocurrencies and Tokens

Cryptocurrencies are the base currencies of blockchains. *Ether* is the currency of the public Ethereum blockchain, and Bitcoin is the currency of the public Bitcoin blockchain (thereby highlighting a source of confusion due to overloaded terminology). The respective blockchain keeps track of the ownership of portions of that currency. Say, Alice owns 2 Ether and announces a transaction to transfer 1 Ether to Bob, offering a fee of 0.01 Ether. Once the transaction is included in a block mined by Charly, Alice has 0.99 Ether, Bob has 1, and Charly received the fee of 0.01 Ether. The sum of the money is not changed by these transactions, but the ownership of portions of it is.

Fees for transaction inclusion are paid in the base currency of a blockchain, although the client can choose to offer a fee of 0 (typically reducing the speed and/or likelihood of inclusion). Fees often relate to the *size* of a transaction, not its *value*: more data (including larger smart contracts to be deployed) incur higher fees. Similarly, more complex computations as a result of smart contract invocations incur higher fees. Transfers of 0.01 Ether incur the same fees as transfers of 100 Ether.

Digital *tokens* can be created and exchanged on blockchains. Usually tokens are created using smart contracts. Similar to a cryptocurrency, each token is controlled by an actor on the blockchain. Tokens might represent shares in a company, the right to benefit from future earnings, or perhaps virtual gold in an online game. The use of tokens has become widespread, and tokens can be seen as the first ‘killer app’ of using blockchain for things other than cryptocurrency.

1.2 Blockchain-Based Applications

Bitcoin has been operational since 2009, and its digital currency had a peak market capitalization of about US$335B in December 2017. The next-largest blockchain, Ethereum, had a market capitalization of US$138B in the same time frame, and there are many other small public blockchains with their own digital currencies. Private blockchains are increasingly deployed inside large enterprises and across industry consortia. The wide array of interest in blockchain technology is underlined by the fast evolution of its ecosystem, including easier deployment through Blockchain-as-a-Service, e.g. from Microsoft Azure[1] and IBM.[2]

Many banks are involved in trials of blockchain technology, including through the R3[3] or Ripple[4] organizations, which are applying blockchain to trade finance and cross-border payments. Financial transactions are the first, but not the only use case being investigated for blockchain technology. A blockchain implements a distributed ledger, which can in general verify and store any kind of transactions. Globally, many financial services companies, enterprises, startups, and governments are exploring its applications in areas as diverse as supply chain, electronic health records, voting, energy supply, ownership management, and protecting critical civil infrastructure. New businesses and business models are expected to arise, but as yet there are not a lot of examples of significant use in production of blockchain systems within industry or government.

Blockchains, particularly public blockchains, offer opportunities for disruptive innovation when implementing decentralized applications. Blockchains provide a new basis for trust in relationships in society, which can allow existing trusted third-party organizations to be disintermediated. In economies where trusted third-parties are not always trustworthy, a significant benefit of blockchain systems may be in the support they can provide for immutability (not changing prior records on the ledger) and non-repudiation (not being able to disown prior actions on the ledger). In developed societies, trusted third-party organizations are usually trustworthy, so the benefits of using blockchain technologies would instead likely arise from enabling faster business model innovation, reducing the cost of establishing business relationships and mitigating risks, and perhaps by reducing the cost or risk of transactions.

For applications of blockchain, there are two *categorically different types*: (1) does the blockchain hold the default source of truth, or (2) does it hold a (possibly incorrect) view of reality? Cryptocurrency is a case of the former: if Bob's account on the blockchain holds 1 Ether, he can control that. By default he is the owner—although a court might determine that he did not fulfil his part of an agreement

[1]https://azure.microsoft.com/en-us/solutions/blockchain/.

[2]http://www.ibm.com/blockchain/.

[3]http://www.r3.com/.

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

and has to pay the 1 Ether back to Alice. In the traditional world, there are some examples of things whose existence and ownership rely on database entries, such as land ownership rights, companies, and patents. These could be ported to a blockchain application of the first type. In contrast, a blockchain record of a physical asset and its state (location, quality, temperature, etc.) is an example of the second category. The view of the asset could be outdated, incorrectly measured, or wrong in some other way. As such, blockchain applications of the first type tend to be more straightforward in their implementation, although they require higher buy-in from the adopters due to their higher degree of reliance on a relatively new piece of technology.

We preview some application areas below and describe some particular use cases in Chapter 4. Three case study chapters in Part IV give detailed accounts from the industry.

1.2.1 Enterprise and Industry

Blockchains were first used for cryptocurrency but are now being used for many other purposes. The full potential of blockchain technology is likely to be realized outside financial services and government. Blockchains are a foundational horizontal platform technology that could be used in any industrial sector including agriculture, utilities, mining, manufacturing, retail, transport, tourism, education, media, healthcare, and the sharing/P2P economy. Generic applications in these sectors include:

Supply Chain When tracking physical assets through changes in ownership and handling, key events and agreements can be recorded and communicated through data stored on a blockchain. This results in provenance information for goods and can provide improved logistics visibility and supply chain quality. Key events within the supply chain could also be linked to automatic payments with the use of smart contracts. Supply chain cases are also captured in the use case chapters on AgriDigital (Chapter 12) and originChain (Chapter 14) as well as Section 4.1.

Internet of Things (IoT) Storage, Compute, and Management Devices connected to the Internet can use the blockchain as a persistent and highly available storage solution. They can also use smart contracts to provide a global distributed-computing capability and can rely on the blockchain as a secure channel for receiving information about software and configuration updates and dynamically-delegated access control. This can include physical access control, for locking and unlocking devices.

Metered Access to Resources and Services Monitoring and payment for usage of utilities or services can be provided by IoT devices and associated smart contracts. An electricity use case is described in Section 4.4.

Digital Rights and IP Management A blockchain can provide a trusted registry of media assets or other intellectual property and can provide the ability to manage, delegate, or transfer access and rights information for those assets. Note that media are not necessarily stored on the blockchain itself. Instead, cryptographic hashes, metadata, and other identifiers stored on the blockchain might be integrated with bulk off-chain storage and communication technologies.

Data Management A blockchain can create a metadata layer for decentralized data sharing and analytics. Although large datasets themselves are unlikely to be stored on it, a blockchain can help to discover and integrate those datasets and data analytics services. Access control mechanisms implemented on a blockchain may allow public data sources to be integrated more easily with private datasets and analysis services. See also Section 4.2 for a use case on open data.

Attestation and Proof of Existence A blockchain can be used to record evidence of the existence of data or documents, by creating a timestamped record of a cryptographic hash of the contents of those documents. This can be combined with records of the attestation or witnessing of corresponding physical documents by trusted third-parties. However, it can be significantly harder to demonstrate the uniqueness or non-existence of such document records, unless there is a widely accepted strict normal form for their contents.

Interdivisional Accounting Multinational companies or large enterprises with separate divisional business units often have jurisdictional or governance needs to control their own internal accounting but also share accounting information with other divisions. A straightforward application of blockchain technologies on a shared private network can create a shared distributed ledger of interdivisional accounts at the interfaces between divisions.

Corporate Affairs (Board and Shareholder Voting and Registrations) The voting authorities of board members or shareholders in companies can be recorded and proxied on a blockchain. Smart contracts on blockchains can use that record to adjudicate votes conducted on the blockchain for specific motions. As blockchain transactions are not necessarily hidden, cryptographic mechanisms may be required to prevent potentially undesirable strategic voting behaviours. The company SecureVote describes their approach and architecture in Chapter 13.

1.2.2 Financial Services

Financial services applications using blockchain technology may include:

Digital Currency New forms of money can be implemented on blockchains, but these can also serve as a foundation for incentive models that support integrity for many blockchain systems. Blockchains allow digital currency to be transferred between parties, often without those transfers being processed or recorded by banks or payment services. With smart contracts, blockchains may be able to support 'programmable money', where automatically enforced policies are attached to specific parcels of currency.

(International) Payments Can be facilitated by blockchain, often via digital currency with local exchanges between the digital currency and fiat currencies. Public blockchain cryptocurrency payments are usually *pseudonymous*. For example, on Bitcoin, transacting agents (which are not necessarily persons) are only identified with a cryptographic key. Therefore international exchange of the Bitcoin digital currency can be performed without establishing real-world identity, and we may not know which actual person is behind which account. Nonetheless, international payments usually have regulatory requirements to establish the identity of participants, as part of Anti-Money Laundering (AML) and Counter-Terrorism Financing (CTF) regulations, and AML/CTF requirements are not obviated by the use of a blockchain. Still, transacting parties can choose to establish their real-world identities to each other and to local exchanges, and this is typically how regulation of blockchain-based international payments is enforced. This topic is also covered in Section 4.3.

Reconciliation for Correspondent Banking Reciprocal nostro/vostro accounts can be replaced by a single shared ledger. Rather than conducting laborious end-of-day reconciliation as a batch task, the two banks can create a single shared view of truth between their accounts, maintained in real time. To limit the distribution of this commercially sensitive information, usually the shared ledger would be restricted to just the two correspondent banks concerned.

Securities Settlement The joint exchange of payment and security holdings is enacted as a single transaction on a blockchain. The exchanged assets are typically represented by tokens implemented on the blockchain, either using smart contracts or other asset representation capabilities provided by the underlying blockchain platform. Payments too are sometimes made using such tokens standing for conventional fiat currency or can sometimes be made using the native cryptocurrency on the blockchain.

Markets Smart contracts on blockchains can provide a platform for making and accepting offers to trade assets or services. The blockchain will record the status of these trade offers. Individual smart contracts could themselves carry the digital currency required to be paid on fulfilment of these offers. This functions as a kind of escrow, without the need for a trusted third-party organization. However, today's blockchain systems are not suitable for high-frequency (low latency) market trading. Also, for public blockchains, pending transactions are visible across the network which can allow a kind of 'front-running', where participants (here, usually the nodes operating the blockchain) might unfairly take advantage of information in these as-yet unexecuted instructions.

Trade Finance The blockchain can be used to evidence trade-related documents in order to reduce lending risk and improve access to finance for industry. Smart contracts could control inter-organizational process execution (see also Chapter 8) and transparently automate delayed or instalment payments. This can improve assurance about previous trading history and about current commitments by counterparties, which can reduce risk to trade finance providers, thus allowing more widespread and economical trade finance offerings into the market. AgriDigital's case study chapter, Chapter 12, discusses these issues.

1.2.3 Government Services

Blockchains could target improved government service delivery, and private blockchains could be used to facilitate information sharing and process coordination across agencies within government. Application areas being explored in governments globally include:

Registries and Identity Including the identities and attributes of persons, companies, or devices, licensing, qualifications, and certifications. Storing registry entries or cryptographic certification of registry entries on a blockchain can facilitate access to and validation against the register. Blockchains could be used to share authenticated identifiers for individuals and companies, and these identifiers could in turn also enable many other blockchain applications. Blockchains can support federated management of multiple related registries, by allowing different agencies to retain authoritative control over the contents of their registers, but still provide a shared view of truth about how their registers are interrelated—see also Chapter 8. The contents of some government registers are public, but in general there are often complex considerations about privacy and confidentiality.

Grants and Social Security Smart contracts could automate the process coordination to apply for, decide on, and distribute payments for grants and social security. With a sufficiently sophisticated payment environment, a smart contract could automatically limit payments to approved suppliers or categories of expenses. One early use of blockchain in this way was to account for allowances and payments by refugees in a UN refugee camp. Other experiments have been carried out in the context of disability support grants.

Quota Management Government-granted quotas, allocations, and rights to physical resources could be awarded and tracked through tokens established on a blockchain. Examples include water access licences providing rights to take a certain volume of water from specific sources during specific time frames or CO_2 emission credits. Where policy allows, blockchain could support an independent secondary market for these rights. The blockchain creates an ongoing immutable audit log of these rights and their use.

Taxation Possible applications range from automated collection of tax using smart contracts to improved compliance by authoritative publication of taxation regulation and calculation tools as smart contracts on blockchain.

1.3 Blockchain Functionality

Software architects need to understand functional and non-functional characteristics of blockchains. In this section, we sketch the functionality of blockchain as a data store and as a computational infrastructure. Figure 1.2 gives an overview of the functionality a blockchain can offer. Blockchains are complex, network-based software components, which can provide data storage, computation services, and communication services. Blockchain features can include cryptographically secure payment, mining, transaction validation, incentive mechanisms, and permission management. What is called an *oracle* supplies information about the external world to the blockchain, usually by adding that information to the blockchain as data in a transaction. Below we expand on the two major functional capabilities of blockchain, for data storage and for computation.

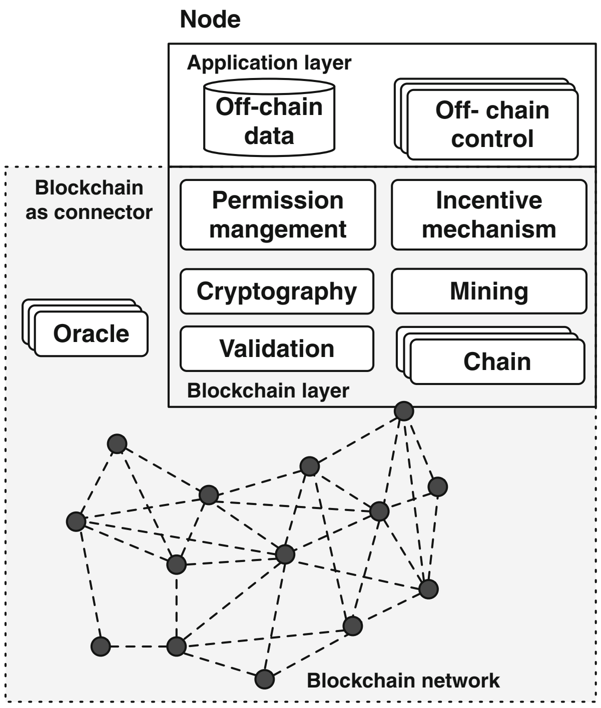

Fig. 1.2 Overview of the functionality that blockchain can offer as an architecture element. © 2016 IEEE. Reprinted, with permission, from Xu et al. (2016)

1.3.1 *Blockchain as Data Storage*

As a data structure, a blockchain is an ordered list of blocks, where each block contains a small (possibly empty) list of transactions. Each block in a blockchain is 'chained' back to the previous block, by containing a hash of the representation of the previous block. Thus historical transactions in the blockchain may not be deleted or altered without invalidating the chain of hashes. Combined with computational constraints and incentive schemes on the creation of blocks, this can in practice prevent tampering and revision of information stored in the blockchain. As a data storage facility, information in a blockchain is recorded within the transactions and within blocks. Important categories of information are transactions about cryptocurrency and transactions involving tokens for other kinds of assets.

Transactions

Transactions update the state recorded on a blockchain. For cryptocurrency transactions, the state information is about the transfer of holdings of cryptocurrency between accounts (addresses). Sometimes additional data can be recorded with a transaction which might have meaning for participants or systems outside of the blockchain. On blockchains such as Ethereum, transactions can record code, variables, and the results of function calls. Public key cryptography and digital signatures are normally used to identify accounts and to ensure integrity and authorization of transactions initiated on a blockchain.

A simplified life cycle of transactions is shown in Fig. 1.3. Once created, the transaction is signed with the signature of the transaction's initiator, which provides the authorization to spend the money, create a contract, or pass the data parameters associated with the transactions. A signed transaction should contain all the information needed to be executed.

A proposed transaction is sent to a node connected to the blockchain network, which checks the validity of the transaction. Invalid transactions are discarded. Valid transactions that are previously unknown to the node are propagated to other connected nodes. These will in turn further validate the transactions and send them to their peers, until the transactions reach every node in the network.

In a global network, this flooding approach means that a valid transaction will usually reach the whole network within a few seconds. To ensure that the transaction propagates, senders do not need to trust any individual node they send the transaction to, as long as they send it to enough other nodes. Recipients do not need to trust senders, because all transactions are signed and can be independently validated by any node.

When a transaction reaches a 'mining' node, it is verified and may be included in a block. *Mining* is the process of appending new blocks to the blockchain data structure. A blockchain network relies on *miners* to aggregate valid transactions into blocks and append them to the blockchain. New blocks are broadcast across the whole network, so that each full node holds a replica of the whole ledger.

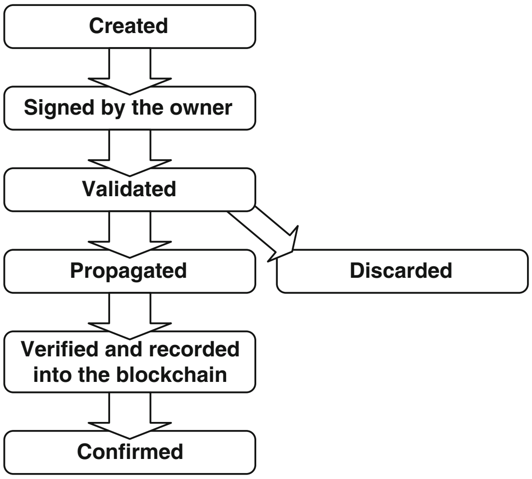

Fig. 1.3 Simplified transaction life cycle

The network aims to reach a consensus about the latest block to be included into the blockchain. There are different consensus mechanisms, e.g. ‘proof of work’ or ‘proof of stake’, which we will describe in more detail later.

However, there is no certainty about whether a particular transaction will eventually be committed or whether it will be outdated. An outdated transaction will be considered an invalid transaction forever, e.g. due to an alternative transaction getting committed. Also, it is often impossible to know whether a transaction that is invalid in some state of the system could ever become valid in a later state. For some consensus mechanisms, the inclusion of the transaction in a newly mined block on some branch of the chain is not sufficient to guarantee that a transaction is permanently added to the blockchain: if the blockchain forks (i.e. conflicting versions of new blocks are proposed simultaneously), then the block comprising the transaction may simply be discarded, in which case it could be re-included later. Each of these inclusions takes time as they require the system to solve computationally hard cryptographic puzzles. If during that time a conflicting transaction Tx' is included first, then the original transaction Tx may simply become invalid, e.g. until a possible third transaction Tx'' compensates for the effect of transaction Tx'.

Depending on the consensus mechanism and the required guarantees, different blockchain platforms and applications can have different notions of when a transaction is *committed* or *confirmed* and thus be immutable. For example, in Bitcoin users often wait five subsequent blocks to be appended to the block containing a transaction before viewing the transaction as committed. However, this is a probabilistic commitment, and so the number of blocks one should wait can depend on the value at risk in the transaction, and the likelihood of it unexpectedly failing. In practice, waiting long enough will make that transaction an immutable part of the earlier history of the Bitcoin blockchain. In contrast, in many private blockchains, committed transactions are more like normal database transactions and so, when accepted under the blockchain’s consensus protocol, will immediately be a permanent part of the ledger.

Digital Assets

One of blockchain's most distinctive capabilities is allowing the creation and secure transfer of *digital assets*. Normally when you give information or digital files to someone, then you both end up with a copy. However, the fundamental characteristic of *property* is that of exclusion: when I have property, no one else has it; and when I transfer that property to you, then I no longer have it. Blockchain transactions and the globally visible blockchain ledger allow everyone to recognize and check the transfer of control or ownership of digital assets registered on the blockchain. This is how blockchain technology supports digital assets. The two most important kinds of digital assets have been discussed earlier: cryptocurrencies and tokens.

Cryptocurrencies are normally 'baked in' to the core platform of public blockchains. They have a kind of symbiotic relationship: the blockchain enables exclusive ownership and secure transfer of the cryptocurrency, and the cryptocurrency enables the incentive mechanism for the operation of the blockchain. Cryptocurrency uses cryptography to control the issuance of money (i.e. minting new coins) and to secure its transfer. Transfers are performed and recorded as financial transactions on a ledger. This virtual money can be transferred directly between users without using a trusted authority such as a bank. The first cryptocurrency, Bitcoin, created in 2009, is still the dominant one in terms of total market value at the time of writing in 2018. There are many cryptocurrencies, most of which are managed through the basic platform capabilities of specific blockchains, such as Ethereum's Ether, Ripple's XRP, and Nxt's NXT. Platforms such as Ripple and Nxt also provide native capabilities to define new cryptocurrencies.

In contrast to cryptocurrencies, tokens are usually not implemented directly in the core platform of a blockchain. Instead, they are implemented on top of blockchain platforms, using transaction data or smart contract features provided by the blockchain. Bitcoin allows developers to add 40 bytes of arbitrary data to a transaction, which is then permanently recorded on the blockchain. Thus, Bitcoin has been used for purposes such as representing digital assets (like document notarization) or physical assets (like diamonds). Bitcoin can also represent tokens using 'overlay networks', for example, using so-called colored coins, where a portion of Bitcoins is tainted to represent and manage real-world assets. Other overlay networks define a completely new transaction syntax, such as Omni and Counterparty. In Ethereum, tokens are usually implemented using smart contracts that maintain a register or table of ownership of tokens. Regardless, as a digital asset, there is much more variation among tokens than there is among cryptocurrencies. Tokens might represent fungible (interchangeable) commodities or might represent unique or serialized assets. Tokens might represent rights to use a service or might represent shares or voting rights in a company. Tokens are often implemented with features allowing their independent transfer or sale, but it is also possible to implement tokens that are not transferable or have other limitations on their transfer.

1.3.2 Blockchain as a Computational Infrastructure

Software components are the fundamental building blocks for software architecture, and blockchain can be a software component offering computational capabilities. As discussed earlier in Section 1.1.2, smart contracts allow us to execute small programs on the blockchain.

Ethereum views smart contracts as a first-class element. Smart contracts on Ethereum can express triggers, conditions, and business logic, to enable complex programmable behaviours. Smart contracts are used by components connected to a blockchain to reach agreements and solve common problems with minimal trust. A common simple example of a smart contract-enabled service is *escrow*, which can hold funds until the obligations defined in the smart contract have been fulfilled. As escrow holder, a smart contract's code has control over the assets held. Smart contracts can also be used to enable machine-to-machine communication for Internet of Things (IoT) applications.

One of the main kinds of architectural decisions is about which pieces of functionality should be allocated to which components. *For blockchain-based systems, this includes the key decisions about which parts of the data and computation should be placed on-chain or kept off-chain.* Parts of an application can be implemented inside the blockchain component using the blockchain ledger and smart contracts. However the amount of computational power, data storage space, and control of read accesses on a blockchain can be limited. So, parts of an application implemented outside the blockchain component might host off-chain data and application logic. Blockchain transactions and their effects sit at the interface between on-chain and off-chain functions.

A common practice is to store hashed data, metadata, and some small-sized public data on-chain and to keep large or private data off-chain. Due to the limited size of the data store provided by the blockchain, an off-chain data store is necessary for some applications. There are existing platforms providing a data layer on top of the blockchains, such as Factom,[5] which stores only the hash of the private data and small amounts of public data in their own blockchain. Distributed data storage, like IPFS,[6] or systems using DHTs (distributed hash tables) are also sometimes used in combination with blockchains to build decentralized applications.

Blockchain computation has a closed-world assumption; smart contracts can usually only examine state that is stored on the blockchain ledger. So in order to interact with the external world, *oracles* are invoked to bring external state into the blockchain. There are various sorts of oracles: some are like normal users of the blockchain and merely record facts about the world as normal transactions on the ledger; while others are components or nodes within the blockchain platform that can invoke smart contracts privileged to them. In either case, oracles typically become a trusted party for the respective data about the external world.

[5]https://www.factom.com/.

[6]InterPlanetary File System (IPFS)—https://ipfs.io/.

1.4 Blockchain Non-functional Properties

Besides blockchain's main functionality described above, software architects need to understand the non-functional properties of a system. We next give an overview of these for blockchain and touch on the implications for systems built on blockchain. As we will discuss in Section 1.5, understanding these issues is of central importance in the design of blockchain-based systems.

1.4.1 Non-functional Properties and Requirements

When specifying a system, software engineers often distinguish functional requirements from non-functional requirements. For a computer system, simple functional requirements characterize the relationship between observable inputs and outputs. Non-functional requirements (NFRs) are needs expressed for non-functional properties (NFPs), which are also known as 'qualities', or 'ilities'. These include characteristics such as cost, security (confidentiality, integrity, availability, privacy, non-repudiation), performance (latency, throughput), modifiability, and usability.

NFRs are expressed separately from functional requirements because they are often 'cross-cutting concerns' that span many system functions. For example, a requirement for the scalability of system performance might constrain the resources that can be used to respond to a given level of concurrent demand in a timely manner, up to some limit on that demand. The demand in this requirement would typically be a mix of many different kinds of system functions in normal usage.

Different use cases carry different NFRs. For example, in safety-critical industries such as medical devices or aerospace systems, NFRs for safety are paramount. In enterprise software systems, regulatory requirements often constrain NFPs such as privacy and data integrity. In regulated industries, legislation or regulation can provide constraints on minimum standards for critical NFPs within the industry. These constraints may be mandated to provide consumer protections or to manage systemic risks or negative economic externalities within the industry. NFPs are also important in understanding innovation: NFPs are quality or performance dimensions for technology, and technological progress pushes out the frontiers of performance on these various dimensions. Orders of magnitude improvements in performance on NFP dimensions open up possibilities for new markets and new business models using that technology innovation.

1.4.2 Non-functional Properties of Blockchain

Compared to conventional centralized databases and computational platforms (on-premise or cloud), blockchains can reduce some counterparty and operational risks by providing neutral territory between organizations. Blockchain technologies may provide advantages for immutability, non-repudiation, integrity, transparency, and equal rights. 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 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.

Trust in the blockchain is achieved from the interactions between nodes within the network. The participants of a 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.

Data privacy and *scalability* are two points of criticism of public blockchains. As discussed earlier, in this setting privacy is limited: there are no privileged users, and every participant can join the network to access all the information on blockchain and validate new transactions. Often, applications need to find an acceptable trade-off between data privacy/confidentiality and transparency.

Current public blockchains have scalability limits on:

1. the size of the data on blockchain, due to the global replication of all data across all full nodes.
2. the transaction processing rate. For example, mainstream public blockchains can only handle on average 3–20 transactions per second,[7] whereas mainstream payment services, like VISA, handle an average of 1700 transactions per second.[8]
3. the latency of data transmission. Because nodes can have a local copy of the blockchain, read latency can be good, but because updates must be propagated across a global network, write latency is typically not good. The number of transactions included in each block is also limited by the bandwidth of nodes participating in leader election (for Bitcoin the current bandwidth per block is 1 MB). Latency between submission and confirmation that a transaction has been included on a blockchain is affected by the consensus protocol. This is around 1 h (10-min block interval with 6-block confirmation) on Bitcoin and around 3 min (14-s block interval with 12-block confirmation) on Ethereum.

[7]https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain/.

[8]https://usa.visa.com/run-your-business/small-business-tools/retail.html.

1.5 Blockchain Architecture Design

Understanding the main functional and non-functional properties of blockchain described above is vital for good architectural design for systems using Blockchain, which we introduce below.

1.5.1 Software Architecture: Design and Analysis

The software architecture of a software-based system is the high-level structure of relationships between software elements (components and connectors) in the system. In the creation of a software architecture, there are many possible options for these structures, and the choices between these options are important design decisions. A key realization in the discipline of software architecture is that these design decisions have a critical impact on a system's ability to meet NFRs. Given a design candidate for a software system, software engineers may use qualitative, analytical, or simulation-based tools to evaluate the design for its predicted ability to achieve an NFR.

To achieve an NFR, the right design decisions must be made, and each design decision will impact a number of NFPs, either positively or negatively. Often this will lead to conflicts between NFPs, so it is important to manage trade-offs between these when designing a system. An important part of software architecture as a practice is to document the design for a system, including the rationale for why specific design options were chosen.

1.5.2 Designing Blockchain-Based Applications

Blockchain-based systems can be different from traditional systems in various ways, as outlined below. Chapter 3 provides a more comprehensive taxonomy.

Admittance of Processing Nodes In a public blockchain system, such as Bitcoin, anyone may become a processing node (or 'miner'). In private blockchain systems, the admittance of processing nodes is controlled by its governing bodies. Public blockchains provide very low barriers to entry for new participants, which can facilitate competition, innovation, and productivity. However, public blockchains typically do not mandate authentication of those participants, which creates challenges regarding AML/CTF and tax avoidance. Private blockchains can impose more controls on authentication and access, which can partly address those regulatory concerns.

Consensus Mechanism Most public blockchains use *Nakamoto consensus*, where processing nodes by convention treat the longest history of blocks as the authoritative history. The rate at which blocks can be created is limited, often by using a proof-of-work mechanism, whereby a processing node can only add a new block by demonstrating that a difficult task has been completed. Proof-of-work is widely used, but the auxiliary effort required to complete the difficult task can be economically inefficient. In a proof-of-stake system, the processing node that can add a new block in the next round is determined by the size of its stakeholding in the global blockchain and/or in that round. Proof-of-stake can be more efficient than proof-of-work but to date has been less widely used. Proof-of-work implementations have demonstrated operational stability over years. Other consensus mechanisms have been proposed. On private blockchains where there are a smaller number of more trustworthy processing nodes, conventional replication algorithms such as Practical Byzantine Fault Tolerance (PBFT) can be used instead of Nakamoto consensus.

Representation of Transactions A distributed ledger may record financial transactions, such as in Bitcoin. However as a shared database, a distributed ledger might allow other kinds of data to be recorded. In particular, the data recorded for a transaction may be the text of a computer program, and the integrity check for that transaction may involve executing that program. This allows participants to create smart contracts, which allow transactions to represent behaviour as well as data.

There are several kinds of blockchains, and to provide more general insights we take a broad view. For example, the Bitcoin system is a public blockchain, which allows unfettered public participation in both its operation and use. Other well-known systems, such as the Ethereum[9] blockchain, are similar in this regard. It is possible to use a separate instantiation of the Bitcoin or Ethereum computer programs to operate a blockchain within a private context, for example, on a virtual private network. These would then be one kind of 'private blockchain'. Note that operators of such networks would not normally use proof-of-work consensus in private networks, because of limitations with that kind of consensus and because other controls or assumptions can be used to address integrity. The access controls possible for private networks and private computer systems allow for greater administrative control over such blockchains. However, the software for public blockchains is not always the best technical solution to use in a private setting. Many industry consortia, such as Hyperledger,[10] R3,[11] and Ripple,[12] are actively developing specialized private blockchain solutions. These typically support a smaller number of processing nodes than public blockchain solutions, but can provide confidentiality and increased performance.

Recently, proof-of-authority (PoA) has gained popularity as a consensus mechanism for private or permissioned blockchain systems, and implementations in

[9]https://www.ethereum.org/.

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

[11]https://www.r3.com/.

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

	Permission-less	Permissioned
Public	**Consensus**: Proof-of-X **Permission management** • Blockchain layer • Application layer (optional) **Incentive**: Blockchain layer	**Consensus** • Proof-of-X • PBFT, Federated consensus, Round Robin *etc.* **Permission management** • Blockchain layer • Application layer (optional) **Incentive**: • Blockchain layer • Governance around permissions
Private	**Consensus** • Proof-of-X • PBFT, Federated consensus, Round Robin *etc.* **Permission management:** • Blockchain layer • Network layer • Application layer (optional) **Incentive**: Governance around access	**Consensus** • Proof-of-X • PBFT , Federated consensus, Round Robin *etc.* **Permission management:** • Blockchain layer • Network layer • Application layer (optional) **Incentive**: Governance around access permissions

Fig. 1.4 Core components of different types of blockchain

	Permission-less			Permissioned	
Public	Immutability	+++	(#Nodes, Consensus, Topology)	Immutability	++
	Integrity	+++	(#Nodes, Consensus, Topology)	Integrity	++
	Transparency	++	(Access control)	Transparency	++
	Availability	+++	(#Nodes, Topology)	Availability	++
	Performance	+	(Consensus, latency)	Performance	++
	Cost Efficiency	+		Cost Efficiency	++
Private	Immutability	+		Immutability	+
	Integrity	+		Integrity	+
	Transparency	+		Transparency	+
	Availability	+		Availability	+
	Performance	+++		Performance	+++
	Cost Efficiency	+++		Cost Efficiency	+++

Fig. 1.5 Non-functional properties of different types of blockchain

Ethereum client software are gaining adoption. PoA assigns the right to mine new blocks to a set of authorities (blockchain accounts, i.e. key pairs) that produce new blocks.

Figures 1.4 and 1.5 show the core components of different types of blockchains and the corresponding quality impacts. In practice, the lack of standard and reliable technology evaluation criteria makes a precise comparison difficult. 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. For example, a blockchain transaction is not appropriate for all data: because it is replicated globally, transactions should not contain very large data nor plain-text data which must be kept confidential. Similarly, for competitors within an industry consortium, private blockchains may not be private enough to provide normal levels of commercial confidentiality for business operations, competitive position, and customer relationships. Consequently there are choices about what data should be stored *on-chain* inside transactions and what should be stored *off chain*, in external systems. Although a specific blockchain platform may have significant limitations, if it can be combined in a design with other components in an effective way, then many kinds of business challenges can be targeted by blockchain-based systems.

1.6 Summary

Blockchains and distributed ledgers are currently very hot topics in computing. In this chapter, we introduced what they are and why there is wide interest in them within various application areas. To provide clarity, we have defined the most important terms used in this book.

Then we discussed, at a high level, the most important aspects for the software architect and engineer aiming to develop a blockchain-based application: what does blockchain offer in terms of functional and non-functional properties and how to approach designing blockchain-based applications?

In the next chapter, we will present an in-depth view of some existing blockchain platforms. Chapter 3 then discusses the conceptual differences between various blockchain technologies and their implications for architectural design. Concrete use cases are discussed in Chapter 4.

1.7 Further Reading

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

The original conception of blockchain was first discussed in the Bitcoin paper (Nakamoto 2008). A more complete introduction of blockchain and Bitcoin can be found in Swan (2015) and Antonopoulos (2015). The original conception of smart contracts predated blockchain technology (Szabo 1997). Smart contracts were originally a way of realizing legal contracts in physical computing systems. However, in the blockchain context, smart contracts are not necessarily related to legal contracts.

Comprehensive surveys on the state of the art of existing cryptocurrencies include Morisse (2015), Bonneau et al. (2015), and Tschorsch and Scheuermann (2016). The market value of cryptocurrencies can be found on http://coinmarketcap.com.

Some government reports discuss potential applications of blockchain in various scenarios, for example, Walport (2016) and Staples et al. (2017). The case of the UN refugee camp's use of blockchain has been described by Juskalian (2018). The experiments on blockchain for programmable money to automate disability support grants are described by Royal et al. (2018).

For an interesting historical account of a community dollar, read *The Island of Stone Money* by Friedman (1991). It describes the island Yap in Micronesia, where currency was held by ownership designation on big stones.

The software architecture of a software-based system is the high-level structure of relationships between software elements (components and connectors) in the system (Clements et al. 2003; Bass et al. 2012). The design of software architecture needs to consider non-functional requirements, which are needs expressed for non-functional properties. These include characteristics such as cost, security and dependability (Anderson 2008; Avizienis et al. 2014) (confidentiality, integrity, availability, maintainability, safety, reliability, privacy, non-repudiation), performance (latency, throughput), modifiability, and usability.