Blockchain, a functional introduction

Marc Sel

PwC
Woluwe Garden – Woluwedal 18 – 1932 St-Stevens-Woluwe

marc.sel @ pwc.com | marc@marcsel.eu

1 Introduction

1.1 Overview

This article provides an introduction to the concepts of what is commonly referred to as "blockchain". The functionality offered by a blockchain is introduced, and its functioning is described. Subsequently blockchain-based solutions are briefly discussed.

The term 'blockchain' is used as a broad catch-all term for a concept based on cryptographic hash trees that has a variety of implementations. The first implementation was the Bitcoin crypto currency. The novelty in Bitcoin was that it used a combination of well-known cryptographic techniques to solve the double spending problem of a virtual currency. This 'double spending' problem refers to the difficulty to represent value in an electronic way, and prevent it from being used multiple times. With paper-based money and payment systems each have their solutions to the double spending problem. However, these do not work for a virtual currency, where a coin is just a series of bits that can be copied.

These well-known cryptographic techniques include linked timestamping for verifiable logs, which goes back to the concept of the Merkle tree[1] and the timestamping concepts from Haber and Stornetta[2]. Regarding digital cash, the seminal work was done by Chaum[3], and fault tolerant

[1] Merkle, R. C. 1980. Protocols for public key cryptosystems. IEEE Symposium on Security and Privacy; http://www.merkle.com/papers/Protocols.pdf.

[2] Haber, S., Stornetta, W. S. 1991. How to timestamp a digital document. Journal of Cryptology 3(2): 99-111; https://link.springer.com/chapter/10.1007/3-540-38424-3_32.

[3] Chaum, D. 1983. Blind signatures for untraceable payments. Advances in Cryptology: 199-203, and Chaum, D., et al. 1988. Untraceable electronic cash. Advances in Cryptology: 319-327; https://dl.acm.org/citation.cfm?id=88969

consensus protocols were proposed by a.o. Lamport[4]. The idea of using public keys as identities is also due to Chaum[5], and smart contracts were proposed by Szabo[6].

After Bitcoin, many variations appeared, aiming to solve other problems, or using a different technical implementation. The most popular one is Ethereum, which comes with its own currency, “Ether”.

2 Blockchain in a nutshell

2.1 Purpose

A blockchain consists of a set of protected information blocks chained sequentially to one-another. Together they form an immurable ledger, distributed over the participating nodes. These nodes are computing platforms that interact with the end users. The terms blockchain and distributed ledger are commonly used as synonyms. The purpose of the ledger is to share information amongst all parties that access it via an application. Access to this ledger in terms of reading and writing may be unrestricted (‘permissionless’), or restricted (‘permissionbased’). The shared information is protected against modification, meaning that any alteration would be easily and immediately detectable. For that reason, once information is recorded on the blockchain, it is considered immutable because it is so strongly protected.

2.2 The blockchain

There is no such thing as ‘the blockchain’. There exist many different blockchains today, some are operated in public, some in private. Without the ambition of being exhaustive, the following are well-known blockhain implementations today: Bitcoin (and a wide range of virtual currencies, referred to as ‘colored coins’), Ethereum, Tendermint, Hydrachain, HyperLedger, Everledger, Lightning Network, Raiden, BigChainDB and Rchain.

2.3 Building blocks

The main building blocks of a blockchain system are its data structure, i.e. the blockchain, and its nodes, where the logic and computations take place. Nodes exist in two types, full function nodes and partial nodes. Each full function node maintains a complete copy of the blockchain, is capable of committing transactions to it, and participates in extending the chain. All full

[4] Lamport, L., et al. 1982. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems 4(3): 382-401; https://dl.acm.org/citation.cfm?id=357176 and Lamport, L. 2001. Paxos made simple; http://lamport.azurewebsites.net/pubs/paxos-simple.pdf.

[5] Chaum, D. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24(2): 84-90; https://dl.acm.org/citation.cfm?id=358563.

[6] Szabo, N. 1994. Smart contracts http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

function nodes are equivalent in terms of functionality, and are connected in a peer-to-peer network. This means there is no hierarchy amongst the nodes, and all nodes are able to communicate with one another. A partial node is equally connected to the network in a peer-to-peer fashion, but does not contain a full copy of the blockchain. It needs the services of a full function node to commit transactions, and it does not participate in extending the chain.

A blockchain starts from its genesis block, and new blocks are appended periodically. Each block records executed transactions. The nodes collaborate to connect the blocks into a blockchain, creating a ledger that cannot be changed backwardly without redoing a proof-of-work.

2.4 Functioning and security functions

Each block contains two types of information. The first type is application-specific information ('payload') that records transactions or smart contracts. These consist of a combination of data and code executable by the nodes. The second type is internal information that secures the block and specifies how it is chained to another. Blocks get automatically propagated across the network, verified and linked via hash[7] values.

The main protection mechanisms are the following. The first protection mechanism is linking each block with its predecessor in a way that is computationally hard to undo. This is achieved by the combination of two techniques. The first technique is the use of a hash tree. This means that a hash is calculated for each block, which includes the hash value of the previous block. This is done for each new block created, with the exception of the first block (the 'genesis' block), which has no predecessor. The second technique is the inclusion of a special number in each block, the block's 'nonce'. Insertion of the right nonce allows to calculate a specific hash value over the entire block. Such a nonce is computationally hard to calculate, therefore it is referred to as a 'proof-of-work'. When the correct nonce is inserted in the location reserved, calculating the hash function over the block will yield a specific hash value, i.e. one that starts with a specified number of zeroes. Since the nonce is hard to calculate, replacing a block by another one would mean redoing the nonce computations of all blocks that were subsequently linked to it. With the current state of algorithms and computing power, it is generally believed to be infeasible after extending the chain with approximately six blocks.

The second protection is the peer-to-peer built-in consensus mechanism. A majority of nodes need to agree about the next block that extends the chain. There is no central point of control that can be compromised. A blockchain system functions without a central trusted entity, in a peer-to-peer mode, where all nodes are equal. There is no trust between the nodes, so they need to rely on a consensus mechanism to confirm the transactions. The consensus mechanism is based on a verification by every node that the received information complies with a set of rules, and by a verification of the nonce (the *'Proof of Work'*). The rules verify that the proposed transaction complies with the application functionality. This is application-specific. For example in the case of a virtual currency it is first verified that the transaction adheres to the required structure, and that the payer has ownership over the coins he wants to spend. Such ownership is demonstrated by a signature using the private key of a Public Key Infrastructure (PKI) key pair. The signature will be need to be successfully verifiable for the transaction to

[7] A hash function is a mathematical one-way function that converts an input string of arbitrary length in an output string of fixed length, e.g. 128 or 160 bits. One-way means given the output, it is mathematically infeasible to derive the input. Other requirements imposed on hash functions include the impossibility for collisions (different inputs that convert to the same output) and the impossibility to find a second pre-image (given the output, it is mathematically infeasible to find a second input that would convert to the same output)

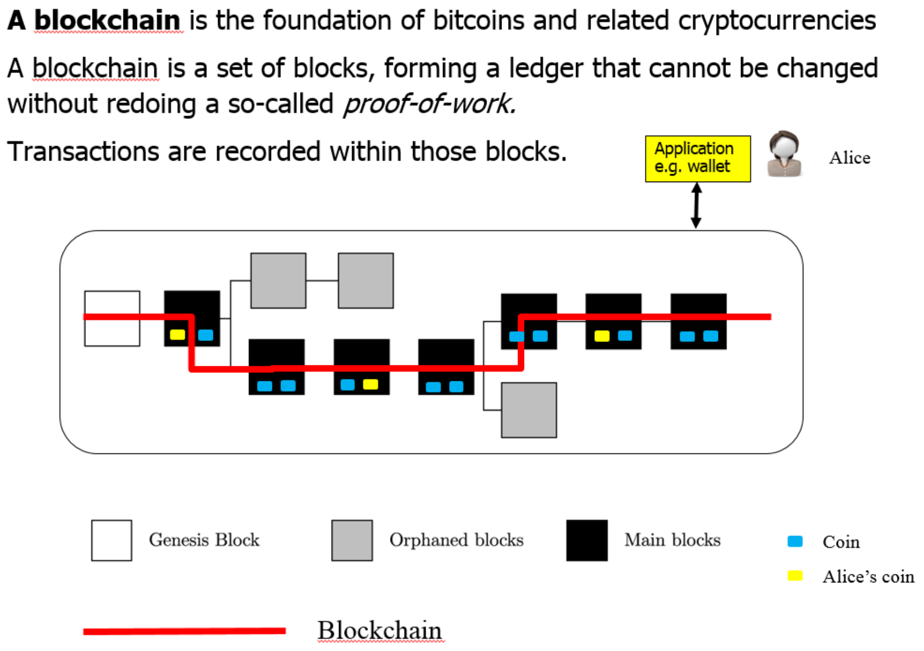
be eligible. Subsequently the verification of the *'Proof of Work'* has to demonstrate that a node invested the required computational power to participate in the extension of the chain.

If two nodes would broadcast different versions of the next block at the same time, some nodes may receive one or the other first. Each node would work on the first block received, but save the other branch in case it becomes longer. The tie will be broken when the next nonce is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

While these two protection mechanisms (linking of blocks and consensus) are inherent to each blockchain, the third protection mechanism is optional. It stems from the fact that blockchains come in two different flavours: permission-less and permissioned. The public, bitcoin-like systems, where every node can participate (read, add entries or extend the blockchain by finalising a candidate block with the correct nonce) are denoted as permission-less. On the other hand, permissioned blockchains allow only a limited set of known and accepted nodes to process the transactions and extend the chain. As this type of chain is typically set by know and consenting organizations with assumed level of trust, the consensus mechanism can be based on a less intensive computational process. Such permissioned blockchain function is based on the self-interest of the participants and they do not need to prove each other they invested sufficient amount of computational power in confirming the transactions.

2.5 Basic applications – virtual coins

Virtual coins are a popular family of applications build on blockchain. A coin consists of the combination of data (representing value) and code (rules on how to spend the value). Figure 1 illustrates the main components of a coin system such as bitcoin (a virtual currency) or namecoin (a repository where DNS-names and their corresponding IP address are stored).



A blockchain is the foundation of bitcoins and related cryptocurrencies

A blockchain is a set of blocks, forming a ledger that cannot be changed without redoing a so-called *proof-of-work.*

Transactions are recorded within those blocks.

Figure 1: Coins on a blockchain

2.5.1 Making a payment

An end user installs a wallet application and generates an account and an address to interact with the blockchain. He initially pays using a traditional payment method to receive his first coins at that address. Once these are received, he can create his own payment transactions from the wallet. Such a transaction contains data and code. The data identifies payer, payee and amount. The code defines in a script language how to unlock the value the payer wants to transfer to the payee, and how to lock the value subsequently to the payee. Performing the transaction requires interaction with a full function node to execute the script code. Upon successful execution, the transaction output is broadcast to peer nodes, which relay the output to further peers.

2.5.2 Mining

Mining is bitcoin's protection against users that try to ‚double-spend'. Upon reception, nodes insert the transaction output they received in the payload of their new candidate block. In the payload there is room for this output, and there are two reserved locations. One location is reserved to be filled in by the nonce, the other one can be filled by a value that represents the creation and allocation of a benefit. All full function nodes insert the benefit value of their choice (typically a transaction that makes a payment to them self) and start 'mining', i.e. searching the nonce that when combined with the rest of the information, yields a valid hash value. This searching is also referred to as the *Proof of Work* (POW). Alternatives for mining exist, the most common one is referred to as *Proof of Stake* (POS).

With POW, the first node to find a hash value that meets the specified condition broadcasts the newly completed block to all other nodes, to verify it. This new block contains the benefit value for the miner that was the first to successfully find the required nonce. If this new block is successfully verified by the network, the originating miner sees his efforts rewarded by the benefit and the results included in the payload of the new block are available in all full function nodes. The successful miner created value for itself, which can be used in future transactions. A competing miner may broadcast his block just after the first miner, and also link his block to the blockchain. However, the nodes will notice the time difference and his block will become an orphan block. This means the block will no longer participate in the active chain.

Partial nodes do not mine, and may store the entire blockchain, or only parts thereof, i.e. those blocks that contain transactions relevant to them. Partial nodes can interact with end users, but they are dependent upon full function nodes to commit transactions to the blockchain. A wallet can be implemented on a mobile devices as partial node, maintaining only information about the coins its owner can spend. The mobile device would not have to store the full blockchain, but would still be able to offer wallet functionality to its user. Making or receiving payments would however require the wallet on the partial node to interact with a full function node.

For more information about cryptocoins, the seminal article by Nakamoto[8] is suggested. Today there are a significant number of competing coins available, and on-line reporting[9] is available via different channels.

2.6 Smart contracts

A smart contract is essentially a computer protocol to digitally agree, verify, or enforce the negotiation or performance of terms between parties, without third parties. These transactions

[8] https://bitcoin.org/bitcoin.pdf

[9] http://coinmarketcap.com/

are trackable and irreversible. As example, consider a smart contract between two parties, Alice and Bob, about the price of a publicly quoted stock S. Our imaginary contract specifies that Alice pays Bob a certain amount if on an agreed date, a condition holds. This condition can e.g. be that the price of S is equal to or above 100 €. Otherwise, Bob pays Alice the same amount. This contract can be encoded in a smart contract programming language such as Solidity, which can then be activated on a blockchain. When the time arrives, the contract will use an oracle to fetch the value of the stock S, and the payment will be made according the condition. Obviously, the oracle must be trustworthy.

Smart contracts are based on the mechanism explained in the preceding section. The underlying idea is to make a breach of a contract expensive (e.g. vending machine dispatches a drink in exchange for cash, 'breaking' the machine is more expensive than supplying the cash).

Smart contracts define rules and consequences, as traditional legal documents do. Furthermore they take information as in input and perform the specified actions as a result. They contain a combination of data and code. Rather than being coded in a dedicated cryptocurrency script language, smart contracts are written in a richer programming language such as Solidity[10]. A contract layout consists of:

contract contractname

Variables (the data part, where 'public' variables maintain the state)

[Events] (optionally, a list of events the contract listens for)

Functions (the code part)

Constructor (the part of the code that creates the contract on the blockchain)

Other functions (other application logic)

Contracts are created by a function called the constructor. Upon execution of the contract's constructor it gets inserted into the blockchain. When the relevant event happens, a blockchain transaction is sent to that address and the smart contract is executed. The execution typically consumes some cryptocurrency value.

Today the most popular implementation of smart contracts is probably Ethereum[11], a public blockchain-based platform. Each node runs the Ethereum Virtual Machine (EVM), which can execute peer-to-peer contracts using a cryptocurrency called ether. It was proposed in 2013 by Vitalik Buterin, and its development was funded by an online crowd sale during July–August 2014. The Ethereum platform was officially launched at July 30, 2015 and is now a significant development ground for smart contract applications.

2.7 Situating Bitcoin

Bitcoin can be seen as the original blockchain. This blockchain was used to implement a cryptocurrency to create the first purely peer-to-peer version of electronic cash without central authority. Bitcoin was created by an unknown (group of) person(s) who invented the blockchain. This unknown (group of) person(s) named itself Satoshi Nakamoto and the Bitcoin development is driven by a core group of Open Source developers.

[10] While the programming languages used to program Bitcoin are limited scripting languages, the languages used to program smart contracts are so-called Turing-complete languages. This means they can simulate any Turing machine. In layman's terms: they are universal programming languages.

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

The trust within the blockchain and thus in the public ledger it represents, is created thanks to collective agreement of the nodes within the network on a set of updates to the state of the Bitcoin ledger. This is referred to as the consensus. This blockchain is the most mature among all public blockchains, but also suffered from the most attacks over the last years.

The rise and fall of the Mt. Gox bitcoin exchange between 2010 and 2014 is illustrative. Mt. Gox exchanged bitcoins for real money and vice versa. They were based in Tokyo, and handled over 70 percent of all the bitcoin transactions worldwide at their peak. They were established in 2010, and filed for bankruptcy in Japan and in the US in 2014. For many customers, they kept the customer's bitcoins in escrow, on a Mt. Gox system. Access was provided to customers through a so-called 'hot wallet'. Such a wallet did not hold the bitcoins, but had access to the bitcoin node where these were held.

As from 2011 onwards they suffered various attacks, as well a regulatory problems. In 2014 Mt. Gox had to announce that approximately 850 000 bitcoins belonging to customers and the company were likely stolen by hackers. These 850 000 bitcoins were evaluated at more than 450 million $ at the time. A detailed technical explanation was never given, and public domain analysis points in the direction of weaknesses in their hot wallet escrow system. But despites Mt. Gox's failure, today there is still a thriving Bitcoin ecosystem.

2.8 Ethereum, the second generation

Ethereum is the first of the second-generation blockchains, which focus on smart contracts. These smart contracts are applications that run exactly as programmed without possibility of downtime, censorship, fraud or third party interference. To do so the Ethereum's creator, Vitalik Buterin, enhanced the Bitcoin's virtual machine scripting mechanism to give Ethereum contracts a state and Turing-completeness. Ethereum contracts encode arbitrary state transitions making it possible to write systems by simply writing the logic in a few lines of code. Ethereum has its own cryptocurrency, called Ether.

Ethereum is the most mature and robust smart contract platform. Ethereum's development is overseen by the Ethereum Foundation that is well funded through the Ether it kept during the Ethereum Initial Currency Offering (ICO[12]). Ethereum was the basis for various other smart contract blockchain projects.

Deploying applications on emerging technology such as Ethereum is not free of risks, as illustrated by the rise and fall of the Distributed Autonomous Organisation (DAO) in the course of 2016. Started by a small blockchain company called Slock.it, a community formed the concept of establishing a decentralised autonomous organisation. This community referred to itself as the DAOhub. The DAO was launched on April 30, 2016, and broke all existing crowdfunding records by amassing the equivalent of about 250 million USD in funding, in Ethereum tokens (Ethers). On May 28, 2016, the DAO went live, and the first project commissioned was the ceation of a smart lock system to enable sharing economy members (such as AirBnB homeowners) to grant access to their homes. As from June 17, 2016, an unknown hacker exploited a bug and in the end managed to drain approximately 30 % of the total funds. Eventually, a 'hard fork' of the Ethereum blockchain was created, with a special contract to move all tokens to. A majority of miners implemented this, and the DAO was erased. The minority of miners that did not implement the change did split from the mainline blockchain, and this split is referred to as 'Ethereum Classic'. Much has been written and said about the rise and fall of the DAO, but

[12] An ICO is the initial period when a cryptocurrency is made available to anybody willing to buy. It can be compared to an Initial Purchase Offering (IPO) when a security is to be traded on a stock exchange

essentially it was an experiment in organisational governance through cryptocurrency and smart contracts. The experiment failed but much was learned along the way. And Ethereum continues to be a popular blockchain technology.

2.9 Other blockchain technologies at a glance

Today’s most active blockchain technologies include the pioneers such as Bitcoin, Ethereum and Ripple (a clearing and settlement technology), as well as Ethereum-based follow-ups that apply new approaches for scalability, such as Tendermint, Hydrachain, and Hyperledger (Burrow, Fabric, Iroha and Sawtooth). There is also a broad category of more scalable designs such as the Lightning Network, Raiden, BigchainDB, RChain, and Aeternity. There are superchains, which connect multiple blockchains together. These include Interledger and Cosmos. And there are others, such as Corda, which come with separated ‘fact’ databases, where the data is kept consistent but not everyone has a copy of everything.

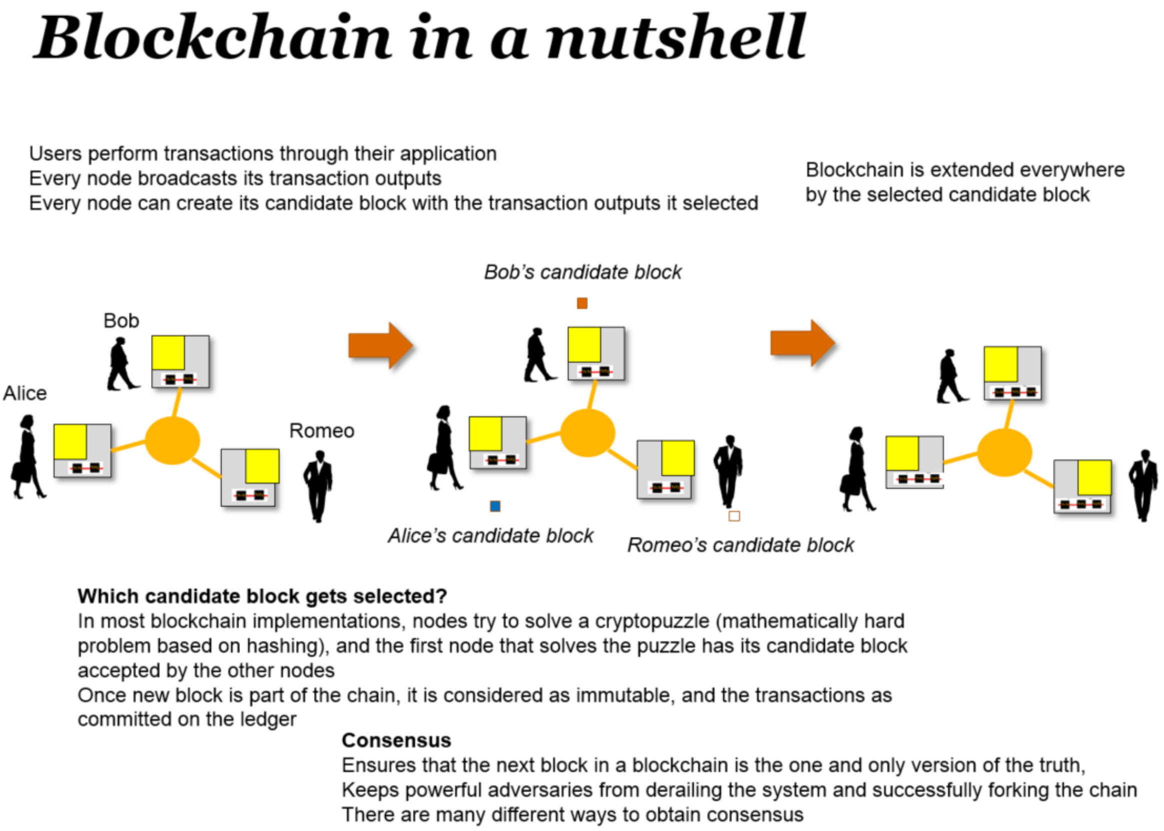
2.10 Functional summary

Blockchain functionality comes in two main types, on-chain and off-chain. This refers to the distinction whether the assets are digital and reside on the blockchain , or the assets are real world assets such as a piece of land, and what resides on the blockchain is a mere representation. The first case is referred to as ‚on-chain‘, and is typical for cryptocurrencies. The second case is referred to as ‚off-chain‘, and is typical for ‚digital doppelgängers‘. Regardless whether the chosen solution is on-chain or off-chain, the following explanation remains valid. A summary is proposed in figure 2.

Assuming a blockchain has been put in place, users perform transactions through their application, every node broadcasts its transaction outputs, and every node can create its candidate block with the transaction outputs it selected. Every node then tries to satisfy the conditions that would allow its candidate block to be promoted to as the next block in the shared chain. The nodes jointly agree on which candidate block is promoted through what is referred to as the consensus.

There are many approaches that define this consensus and its conditions. The most well-known ones are *Proof Of Work* (POW) and *Proof Of Stake* (POS). POW originated as a solution to fight spam emails, by enforcing that senders demonstrate they performed some calculations prior to accenting their emails for sending onward. These calculations consist of solving a problem which is moderately hard but feasible to execute, and easy to check. A popular problem is a partial hash inversion, i.e. finding the input to a hash function that satisfies conditions on the output such as containing a number of consecutive zeros. The latter is moderately hard (depending on the hash chosen) because a good hash’s output will be random, so a minimal amount of calculations will have to be done to find a hash that contains the specified number of zeros. In Bitcoin, POW is used to prevent the „double spending“ of coins. In POS, the next valid block in the blockchain is selected on the basis of account holders‘ stakes. Many schemes exist for POS, including those based on the concept of "coin age", a number derived from the product of the number of coins times the number of days the coins have been held.

One might wonder if one’s candidate block might never make it to inclusion, if the node where this candidate block is formed is e.g. not powerful enough to successfully compete in a POW scheme. First, it should be realised that transactions are send out by nodes for inclusion in multiple candidate blocks. So a transaction will normally over time make it on the chain, via one candidate block or another. Second, techniques can be used that offer an incentive for a node to include a transaction in a candidate block, e.g. by paying a small transaction fee.



Blockchain in a nutshell

Users perform transactions through their application
Every node broadcasts its transaction outputs
Every node can create its candidate block with the transaction outputs it selected

Blockchain is extended everywhere
by the selected candidate block

Which candidate block gets selected?
In most blockchain implementations, nodes try to solve a cryptopuzzle (mathematically hard problem based on hashing), and the first node that solves the puzzle has its candidate block accepted by the other nodes
Once new block is part of the chain, it is considered as immutable, and the transactions as committed on the ledger

Consensus
Ensures that the next block in a blockchain is the one and only version of the truth,
Keeps powerful adversaries from derailing the system and successfully forking the chain
There are many different ways to obtain consensus

Figure 2 Extension of the blockchain from 2 to 3 blocks