Bitcoin Basics

Öz, B., Hoops, F., Gallersdörfer, U., & Matthes, F. (2022). “Blockchain-based Systems Engineering”. Lecture Slides. TU Munich.

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

1. Introduction to Bitcoin & Blockchain

2. Bitcoin Architecture
- Blockchain & Blocks
- Block Header & Contents
- Genesis Block

3. Transactions in Bitcoin
- Data Structure
- Transaction Validation

4. Bitcoin Network
- P2P Network
- Types of Nodes

5. Bitcoin Use Cases

6. Storing Bitcoins

This chapter is heavily inspired by and uses examples from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

Technical Paper

Bitcoin:
A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto

October 31, 2008

Abstract

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

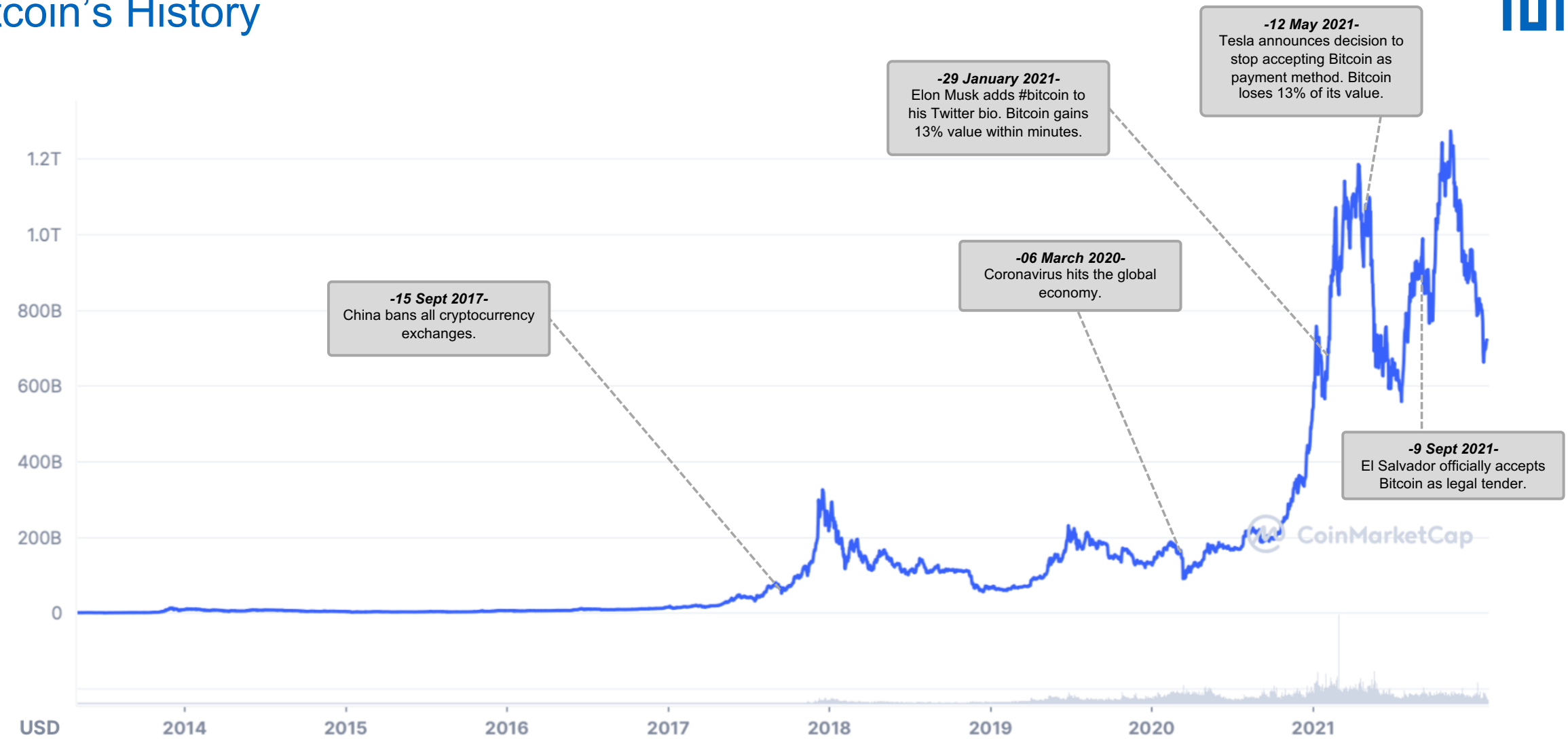
Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions,

- This is a technical paper published online in the year of the financial crisis.
- **Satoshi Nakamoto** wrote a paper in 2008 about *“Bitcoin: A Peer-to-Peer Electronic Cash System”*
- The **real identity** of Satoshi Nakamoto remains **unknown**. We don’t know whether it was a single person or a group.
- In **January 2009** the **first block** of the Bitcoin blockchain was **mined**.
- In contrast to public opinion, Bitcoin was **not invented because of the financial crisis**. Satoshi Nakamoto said he started working on Bitcoin in May 2007 (in contrast the financial crisis started in August 2007).

Nathaniel Popper - Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money

[NAKA2008] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).

Bitcoin's History



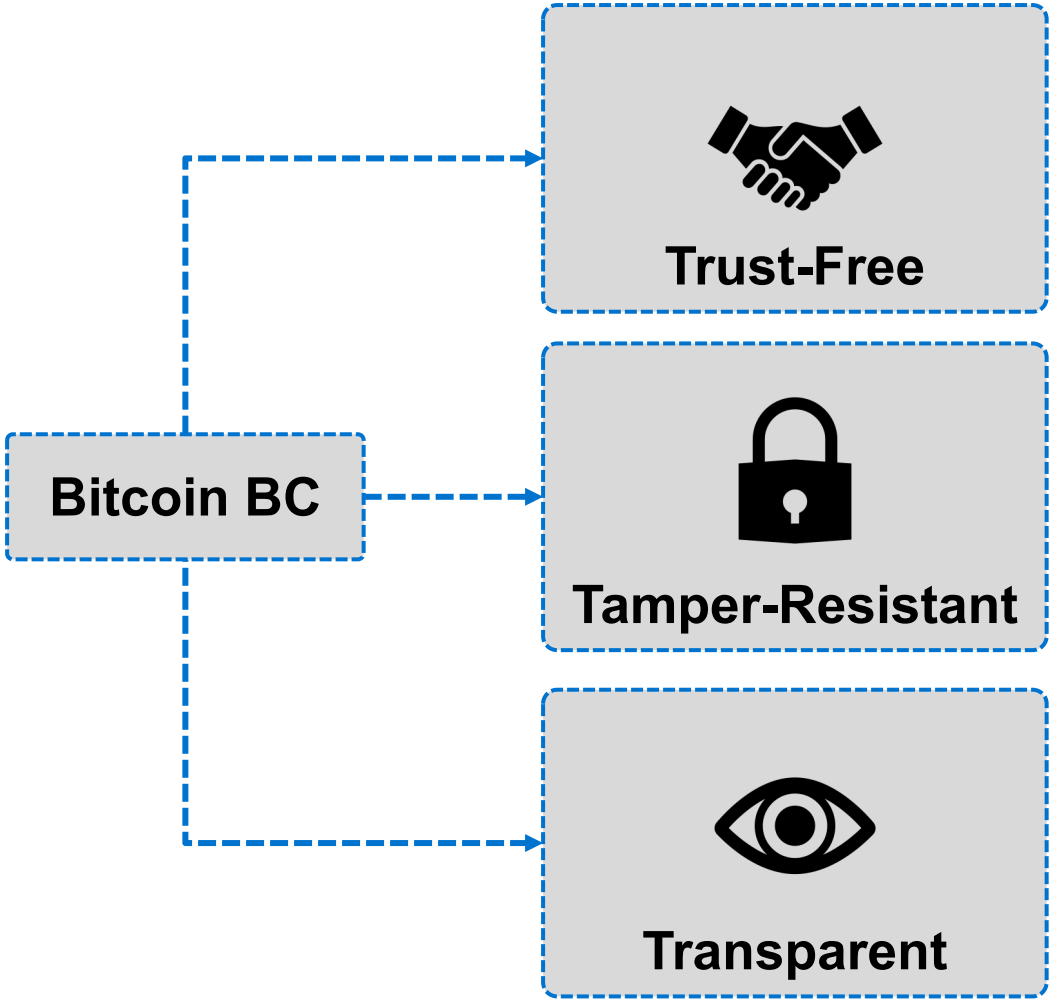
[1]A decentralized organization created to allow people to lend and borrow cryptocurrencies.

Coinmarketcap, Bitcoin Charts, https://coinmarketcap.com/currencies/bitcoin/, retrieved 30th January 2022

Key Properties of the Bitcoin Blockchain

It has **three key properties**:

- **Trust-Free**: *The system does not require a third party which controls or maintains the system.*
- **Tamper-Resistant**: *The system is resistant to manipulation. The history of events cannot be changed.*
- **Transparent**: *Every participant of the system can read and validate all information and the current state.*



Outline

1. Introduction to Bitcoin & Blockchain

2. Bitcoin Architecture
   - Blockchain & Blocks
   - Block Header & Contents
   - Genesis Block

3. Transactions in Bitcoin
   - Data Structure
   - Transaction Validation

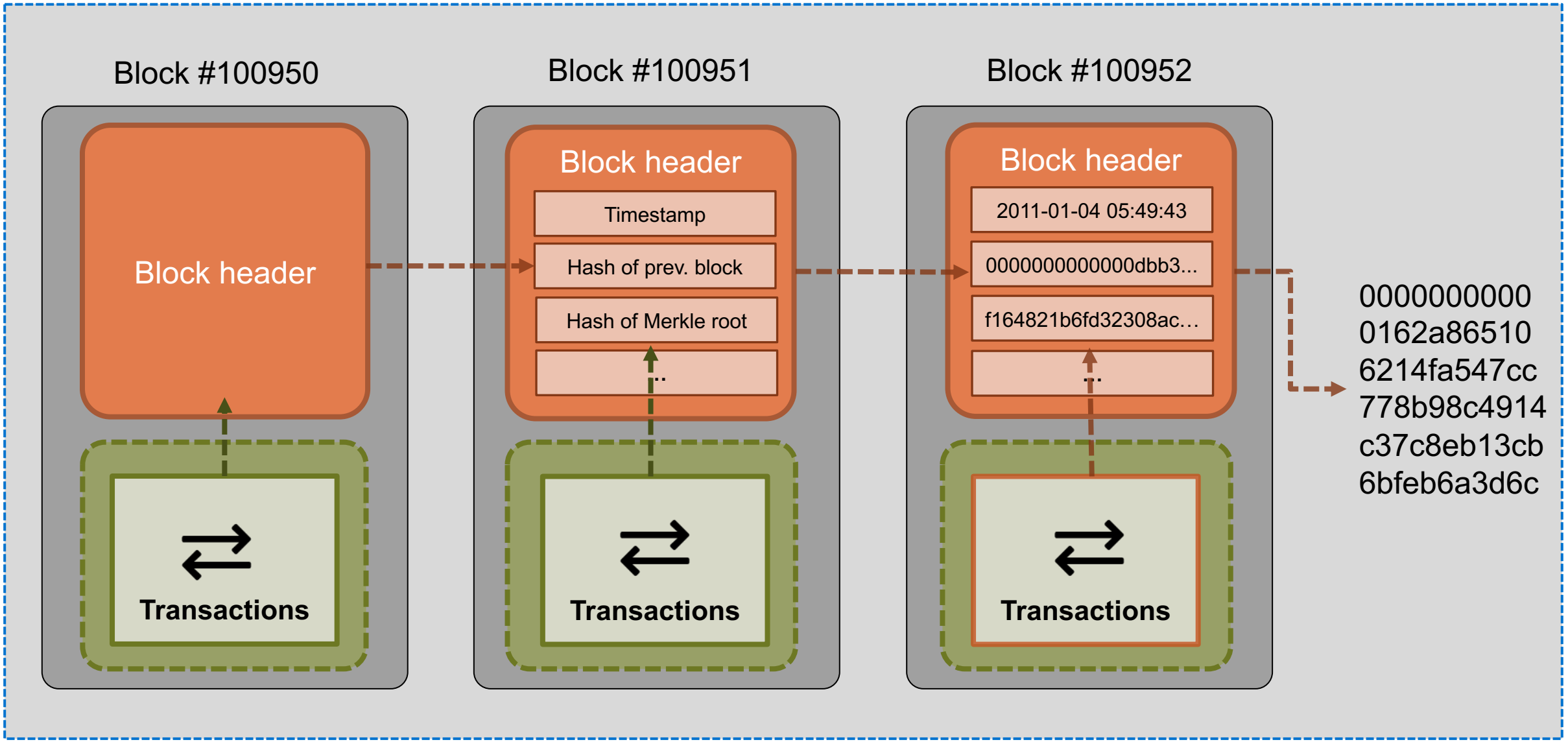
4. Bitcoin Network
   - P2P Network
   - Types of Nodes

5. Bitcoin Use Cases

6. Storing Bitcoins

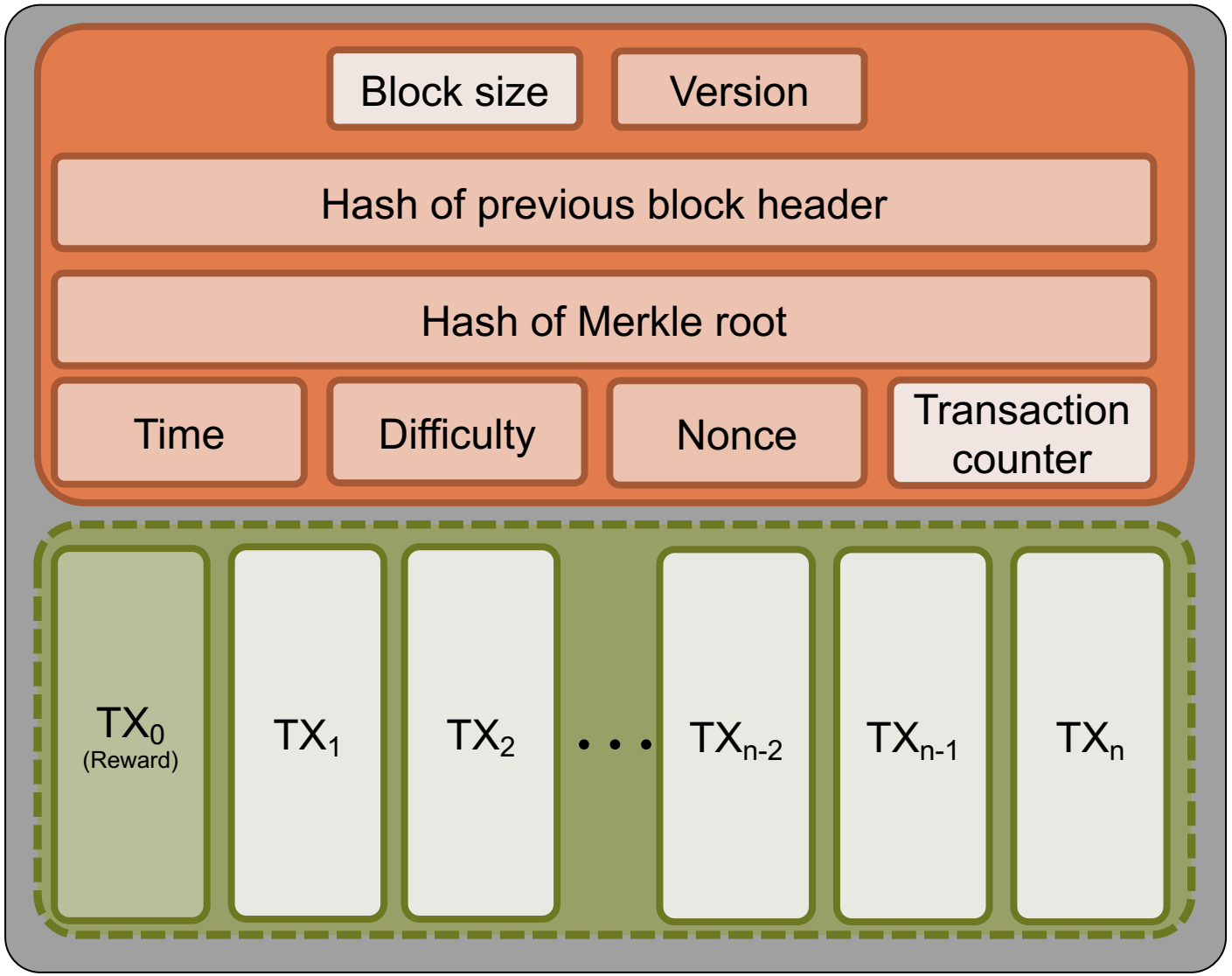
This chapter is heavily inspired by and uses examples from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

Bitcoin Blockchain



Block Details

- The *hash of the previous* block creates the chaining.
- The hash of the Merkle root node of a Merkle tree structure with all transactions (as explained in Chapter 2).
- The *nonce* is required for the consensus mechanism in the network.
- The block‘s hash used for chaining is calculated from the *version* until the *nonce* field.
- The height of the block is stored in the coinbase transaction. (TX_0)



Genesis Block

Blocks have to reference their predecessor ⟶ **How does a blockchain start?**

Bitcoin‘s genesis block[1]:

- mined at 2009-01-03 18:15:05
- references a previous block with hash `0`
- contains only the mining reward transaction → first 50 BTC which can never be spent

The fact that it cannot be spent is based on the source code of the current `bitcoin-core` client[2]. The client searches through all blocks in `ConnectBlock` and processes all transactions but skips the genesis block.

[1] https://blockchain.info/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f (accessed 24.10.2017)
[2] https://github.com/bitcoin/bitcoin/blob/9546a977d354b2ec6cd8455538e68fe4ba343a44/src/main.cpp#L1668 (accessed 24.10.2017)

Data Contained in the Genesis Block

THE TIMES

Saturday January 3 2009 timesonline.co.uk No 69523 £1.50

Eat Out from £5

More than 900 great restaurants, including four **Gordon Ramsay** favourites from £15

Start collecting tokens today Pullout inside

Israel prepares to send tanks and troops into Gaza

Israel allowed foreigners to flee the Gaza Strip as it prepared for a ground offensive. At least 430 Palestinians were killed in a week of airstrikes News, page 3

Michael Sheen Frost, Nixon and me

Magazine

Working mums So that's how she does it

Body&Soul

Detox in style The best spas on the planet

Travel

Salman Rushdie I won't marry again

Pages 22, 23

Giant killing? Guide to the FA Cup third round

Sport

Chancellor on brink of second bailout for banks

Billions may be needed as lending squeeze tightens

Francis Elliott Deputy Political Editor
Gary Duncan Economics Editor

99p

Pub chain cuts the price of a pint from £1.69 to 1989 levels

Business, page 47

```
00000000   01 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000010   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000020   00 00 00 00 3B A3 ED FD   7A 7B 12 B2 7A C7 2C 3E   ....;£íýz{.²zÇ,>
00000030   67 76 8F 61 7F C8 1B C3   88 8A 51 32 3A 9F B8 AA   gv.a.È.Ã^ŠQ2:Ÿ¸ª
00000040   4B 1E 5E 4A 29 AB 5F 49   FF FF 00 1D 1D AC 2B 7C   K.^J)«_Iÿÿ...¬+|
00000050   01 01 00 00 00 01 00 00   00 00 00 00 00 00 00 00   ................
00000060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000070   00 00 00 00 00 00 FF FF   FF FF 4D 04 FF FF 00 1D   ......ÿÿÿÿM.ÿÿ..
00000080   01 04 45 54 68 65 20 54   69 6D 65 73 20 30 33 2F   ..EThe Times 03/
00000090   4A 61 6E 2F 32 30 30 39   20 43 68 61 6E 63 65 6C   Jan/2009 Chancel
000000A0   6C 6F 72 20 6F 6E 20 62   72 69 6E 6B 20 6F 66 20   lor on brink of
000000B0   73 65 63 6F 6E 64 20 62   61 69 6C 6F 75 74 20 66   second bailout f
000000C0   6F 72 20 62 61 6E 6B 73   FF FF FF FF 01 00 F2 05   or banksÿÿÿÿ..ò.
000000D0   2A 01 00 00 00 43 41 04   67 8A FD B0 FE 55 48 27   *....CA.gŠý°þUH'
000000E0   19 67 F1 A6 71 30 B7 10   5C D6 A8 28 E0 39 09 A6   .gñ¦q0·.\Ö¨(à9.¦
000000F0   79 62 E0 EA 1F 61 DE B6   49 F6 BC 3F 4C EF 38 C4   ybàê.aÞ¶Iö¼?Lï8Ä
00000100   F3 55 04 E5 1E C1 12 DE   5C 38 4D F7 BA 0B 8D 57   óU.å.Á.Þ\8M÷°..W
00000110   8A 4C 70 2B 6B F1 1D 5F   AC 00 00 00 00            ŠLp+kñ._¬....
```

The data highlighted is stored in the *scriptSig* field of the first transaction (=coinbase transaction).

A possible motivation to put this headline into the genesis block is to prove that no “pre-mining” has happened.

Outline

1. Introduction to Bitcoin & Blockchain

2. Bitcoin Architecture
- Blockchain & Blocks
- Block Header & Contents
- Genesis Block

3. Transactions in Bitcoin
- Data Structure
- Transaction Validation

4. Bitcoin Network
- P2P Network
- Types of Nodes

5. Bitcoin Use Cases

6. Storing Bitcoins

This chapter is heavily inspired by and uses examples from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

Account-based Ledger

Transactions
Create 25 coins and credit to Alice signed by miners
Transfer 17 coins from Alice to Bob signed by Alice
Transfer 8 coins from Bob to Carol signed by Bob
Transfer 5 coins from Carol to Alice signed by Carol
Transfer 15 coins from Alice to Bob signed by Alice

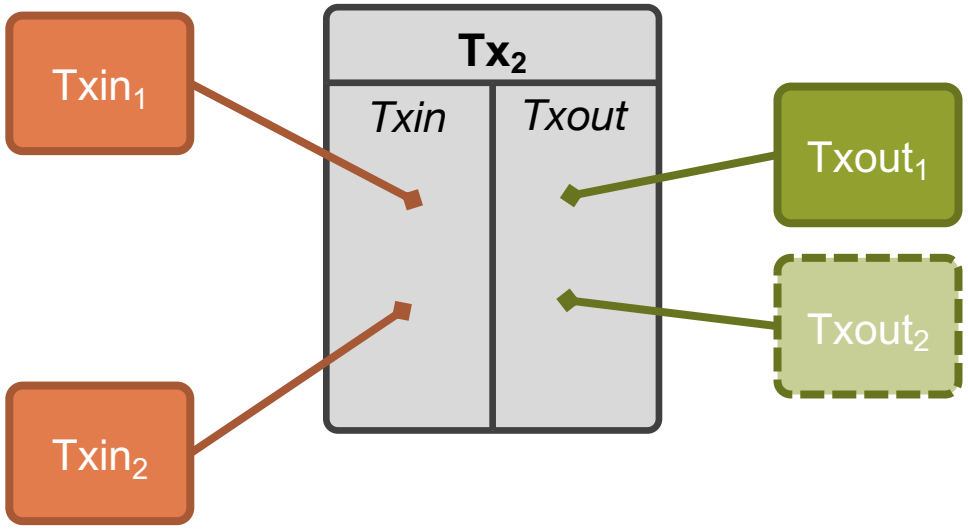
Alice	Bob	Carol
25	0	0
8	17	0
8	9	8
13	9	3
-2	24	3

World State

- *Intuitively*: We consider Bitcoin to use an account-based ledger. However, an account-based approach takes a lot of effort to track the balances of every account.
- In reality, Bitcoin keeps track of the transactions an account has received and does not add up account balances (on the chain).
- Transactions lead to a “world-state” of accounts and allow to calculate account balances off-chain (e.g., in a wallet).
- By using a transaction-based ledger, Bitcoin enables wallet owners to define conditional transactions using Bitcoin Script[1].

[1] *Bitcoin Script is a stack-based scripting language used in Bitcoin transactions. It will be covered further in the upcoming exercises..*

Transaction-based Ledger

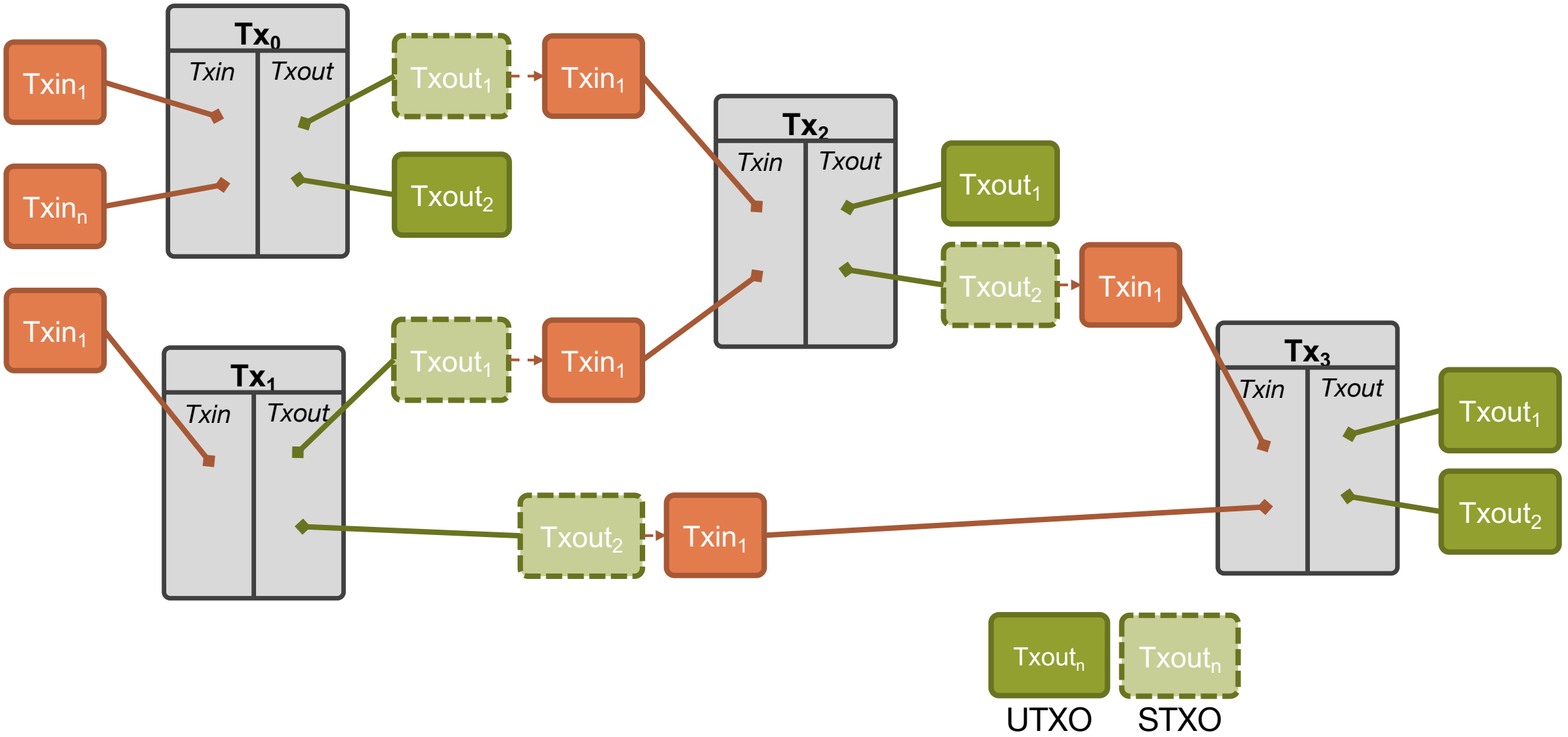


- Transactions (**Tx**) have a number of inputs and a number of outputs.
  - **Inputs (Txin)**: Former outputs, that are being consumed
  - **Outputs (Txout)**: Creation of new coins and transfer of coin ownership

- In transactions where **new coins** are created, **no Txin** is used (no coins are consumed)

- Each transaction has a unique identifier (**TxID**). Each output has a unique identifier within a transaction. We refer to them (in this example) as *#TX[#txout]*, e.g., 1[1], which is the second Txout of the second transaction.

Transactions Connected by Inputs and Outputs



Transaction-based Ledger

0	Txin: ∅ Txout: 25.0 → Alice
1	Txin: 0[0] Txout: 17.0 → Bob, 8.0 → Alice signed by Alice
2	Txin: 1[0] Txout: 8.0 → Carol, 9.0 → Bob signed by Bob
3	Txin: 1[1] Txout: 6.0 → David, 2.0 → Alice signed by Alice

Example:
0. No input required, as coins are created.
1. The Tx is used as an Txin. Two Txout are created, one to Bob and one to Alice. (1[0] and 1[1]) The Tx is signed by Alice.
2. Uses first Txout of Tx1. Creates two Txout to Carol and Bob, signed by Bob.
3. Uses second Txout of Tx1. Creates two Txout to David and Alice, signed by Alice.

Further remarks

Change Address:
Why does Alice have to send money back to herself? In Bitcoin, either all or none of the coins have to be consumed by another transaction. The address the money is sent back to is called a *change address*. This enables an efficient verification, as one only has to keep a list of **unspent transaction outputs** (**UTXO**).

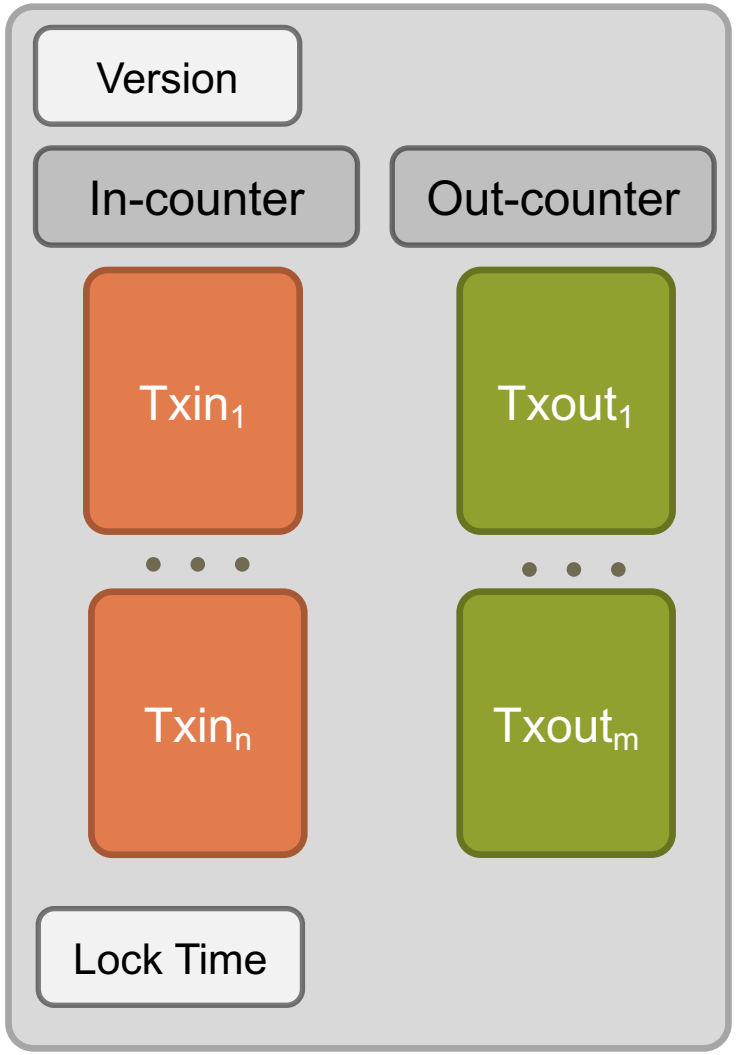
Consolidating funds:
Instead of having many unspent transaction outputs, a user can create a transaction that uses all UTXO she has and creates a single UTXO with all the coins in it.

Joint payments:
Two or more parties can combine their inputs and create one output. Of course, it requires signatures from all involved parties.

An Advanced Look at Transactions

As previously stated, transactions consist of inputs and outputs following these principles:

- All inputs reference an existing unspent output or a coinbase transaction.
- Inputs and outputs **contain scripts** (scriptSig, scriptPubKey) **for verification**.
- Output scripts (scriptPubKey) **specify the conditions to redeem their value**.
- Input scripts (scriptSig) **provide a signature** to redeem the referenced output.
- Only **outputs store** the **BTC value** and the **receiver's address**.
- **All coins have a history (inputs/outputs) up to the original coinbase transaction that created them.**



Input format

Txin

- previous transaction hash
- previous `Txout`-index
- script length
- *scriptSig*

Output format

Txout

- value in *Satoshi (=10^{-8} BTC)*
- script length
- *scriptPubKey*

Logical Data Structure of Transactions


```
"hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
"ver":1,
"vin_sz":2,
"vout_sz":1,
"lock_time":0,
"size":404,
"in":[
 {
  "prev_out":{
   "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
   "n":0
  },
    "scriptSig":"30440..."
 },
 {
  "prev_out":{
   "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
   "n":0
  },
  "scriptSig":"3f3a4ce81...."
 }
],
"out":[
 {
  "value":"10.12287097",
  "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
 }
]
```


An example transaction with two Txin and one Txout.

Metadata
It contains the **size of the transaction**, the **number** of **inputs** and **outputs**, the version and a lock-time. The hash is also contained, which can be referenced.

Inputs
An **array** of **all inputs**. Each input contains the previous transaction and the index of Txout. Also, a signature *script* is provided.

Outputs
An **array** of **all outputs**. One output has two fields: the amount of the transferred coins[1] and the *scriptPubKey*.

[1]Note, that the sum of all output amounts has to be the same or smaller than the sum of all inputs. The difference is the transaction fee, which is covered in the chapter “Bitcoin Consensus”.

Example taken from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan

Validation of Transactions

Whenever a transaction is validated, it must pass a long list of checks. Simplified, they look like this:

1. Check syntactic correctness
2. Make sure neither in or out lists are empty
3. Check that size in bytes <= MAX_BLOCK_SIZE
4. Each output value, as well as the total, must be in legal money range
5. No input may be a coinbase transaction input
6. Check that each input …
   a) references an old output
   b) references an old output not yet referenced in another transaction (in block or in memory pool)
   c) contains the correct authorization to spend that old output
   d) … satisfies the conditions defined by the script
7. Check that the sum of output values <= sum of input values
8. Add to transaction pool
9. Relay transaction to peers

The Bitcoin wiki provides an overview about Bitcoin protocol rules for clients.

Outline

1. Introduction to Bitcoin & Blockchain

2. Bitcoin Architecture
- Blockchain & Blocks
- Block Header & Contents
- Genesis Block

3. Transactions in Bitcoin
- Data Structure
- Transaction Validation

4. Bitcoin Network
- P2P Network
- Types of Nodes

5. Bitcoin Use Cases

6. Storing Bitcoins

This chapter is heavily inspired by and uses examples from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

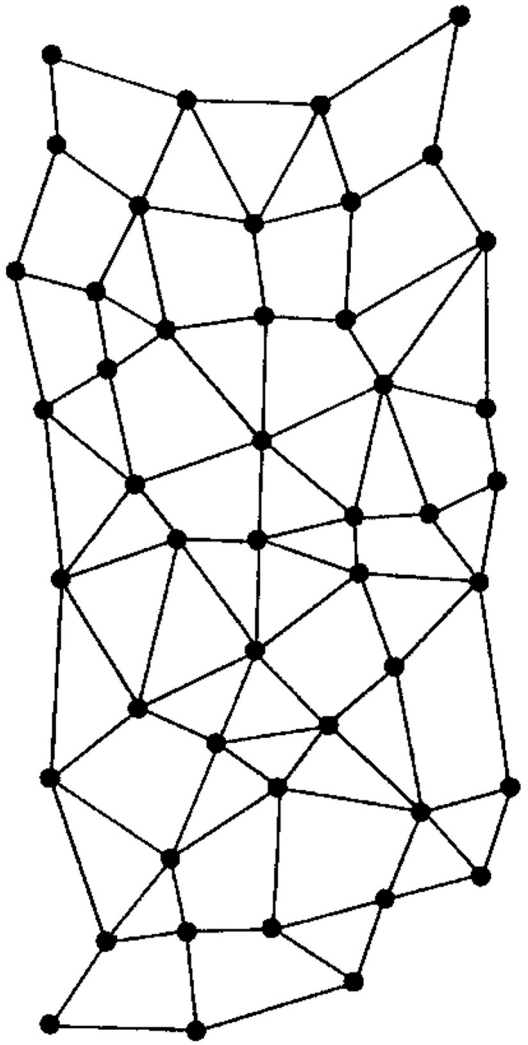
Gossip Protocol

- Bitcoin itself consists of different users and nodes. We distinguish between **wallet owners**, **light nodes**, **full nodes** and **miners**. (More on that topic later)

- They communicate in a **decentralized** fashion, meaning that no single entity or node is superior.

- To communicate, they need to have clear rules
  - How to **find** other nodes (bootstrapping)
  - How the **sync** the blockchain
  - How to **send** and **receive** transactions
  - How to **send** and **receive** blocks

- The basic network uses a peer-to-peer **gossip** protocol. Messages about new blocks or transactions are **validated** and then **broadcasted**. To prevent a second broadcast, the node keeps track of the transactions and blocks sent by itself.



An advanced look into the verification of messages containing transactions or blocks provides The Bitcoin Wiki: Protocol Rules.

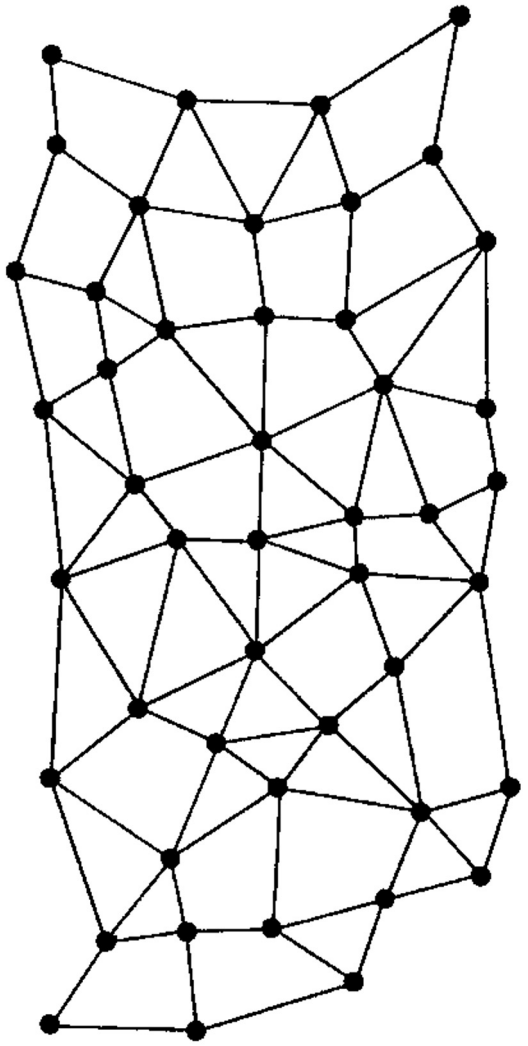
Bootstrapping of Nodes

Client Node Discovery

How are new nodes introduced into the Bitcoin network?

There are several ways:

- (deprecated) Get to know new clients via **IRC-channels**
- Hard-coded **DNS-services** which offer IP-addresses of nodes
- Hard-coded **seed addresses** (last resort)
- Addresses stored in a **database** maintained locally (to be loaded after a restart)
- **Command-line** provided addresses
- **Text-file** provided addresses



Roles in the Bitcoin Network

Full nodes are almost always a miner.

Wallet Owner (User)

- The wallet owner owns different **private keys** to unspent transaction outputs (UTXOs).
- He is the **owner** of all stored currencies on these addresses.
- He sends money by **signing** and **publishing** new transactions to a connected light node, full node or miner.

Full Node (Software)

- The full node **maintains** the **complete blockchain**. Its record of the chain is complete, it contains every single transaction and block until the genesis block.
- Is **connected to** other **full nodes** and exchanges information. Namely:
  - **Validates** every transaction and block it receives
  - **Relays** all new transactions and blocks

Miner (Software)

- The miner **needs** the **same record** as a **full node** in order to work properly. He also is connected to other nodes and maintains the network.
- Additionally, the miner is **responsible** for **creating** new **blocks** by trying to solve the **mining puzzle**.
- The miner gets rewarded by creating new blocks.

Light Node (Software)

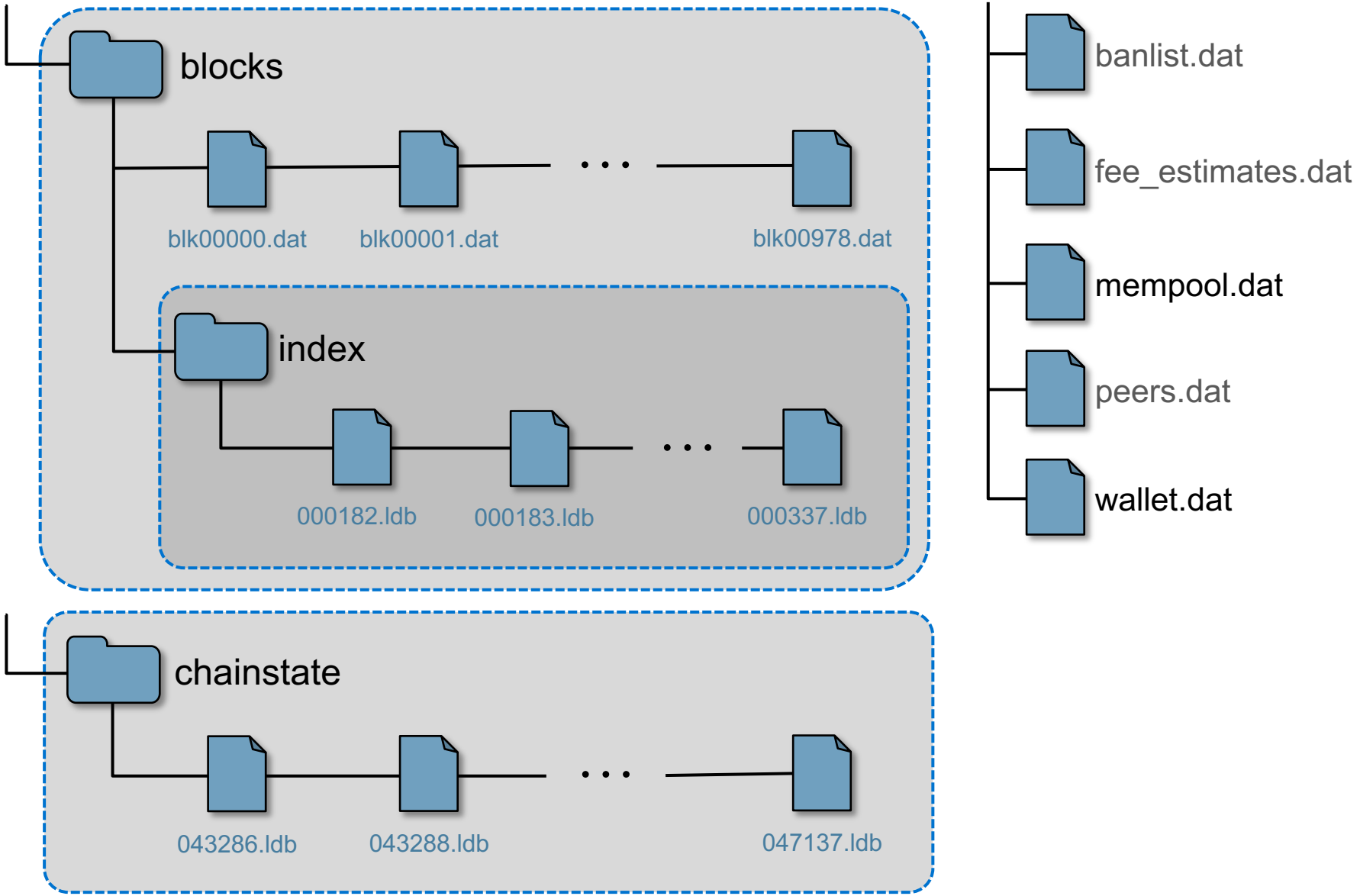
- The light node can **act** as a **relay** for transactions of one **wallet owner.**
- It **validates** whether a **single transaction** of the wallet owner was executed correctly.
- The light node also **requires a full node** to connect to the network.
- **Almost no relevance** in practice today. Today, centralized services are used to create transactions.

Anatomy of the Bitcoin Blockchain

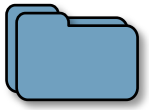
Raw data is on a disk.

Miners and full nodes organize their data in a certain way. (Bitcoin core)

As of February 2022, the total data size of the Bitcoin blockchain is 388 GB.



Anatomy of the Bitcoin Blockchain



blocks and **blocks/index**

Contains .blk files that contain the actual blockchain in raw network format; **index** contains a database which stores the location of each block on the disk keyed with its hash.

chainstate

A LevelDB (leveldb.org) database with all currently unspent transaction outputs in the system (UTXO). This is used when operating a Bitcoin node in favor of the raw blockchain data.

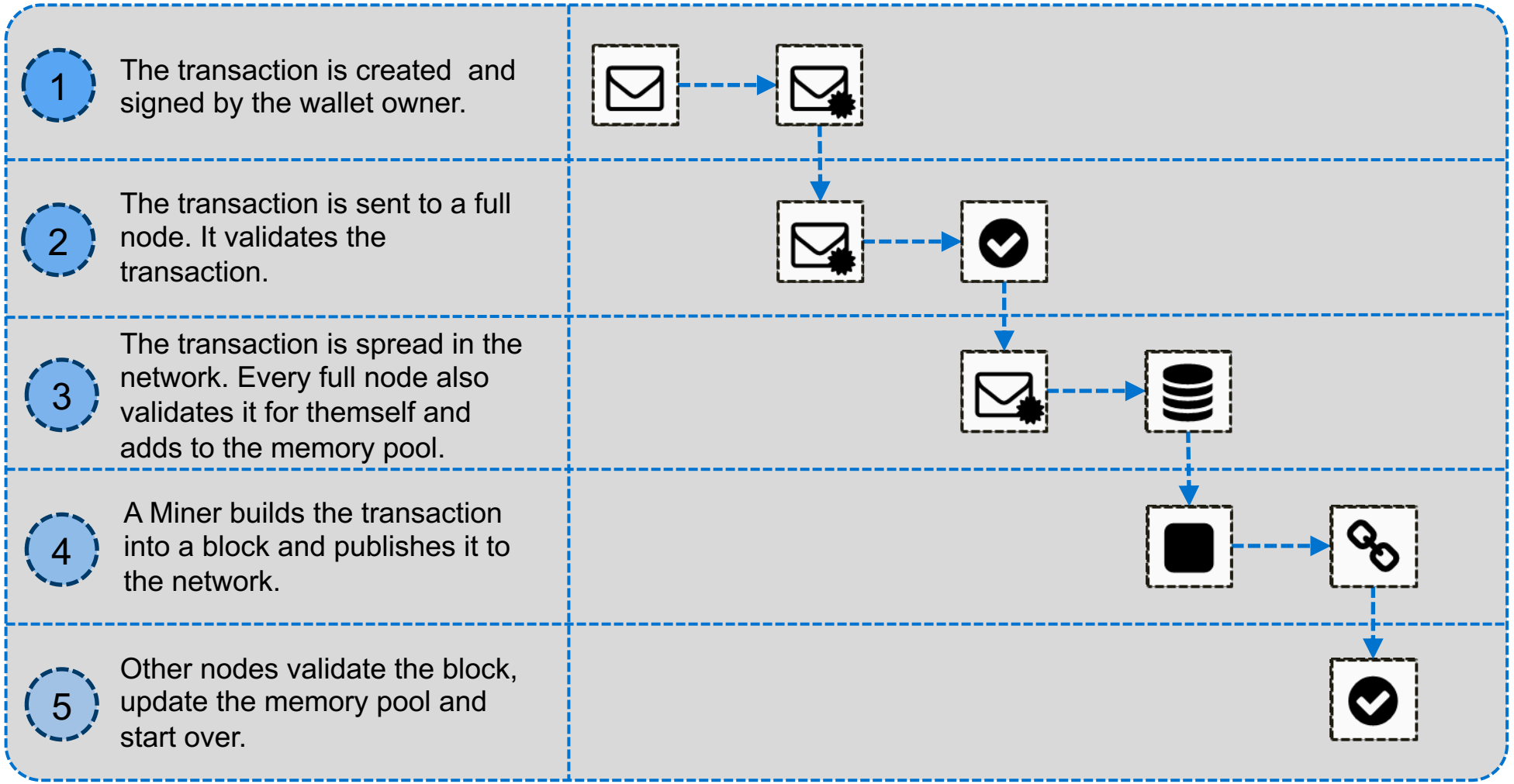
mempool.dat

A list of unconfirmed transactions to be part of a future block.

wallet.dat

Data regarding the user‘s (owner of the node) personal wallet.

A Newly Created Transaction's Way into a Block



A high-level representation of how transactions are included in blocks.

Outline

1. Introduction to Bitcoin & Blockchain

2. Bitcoin Architecture
- Blockchain & Blocks
- Block Header & Contents
- Genesis Block

3. Transactions in Bitcoin
- Data Structure
- Transaction Validation

4. Bitcoin Network
- P2P Network
- Types of Nodes

5. Bitcoin Use Cases

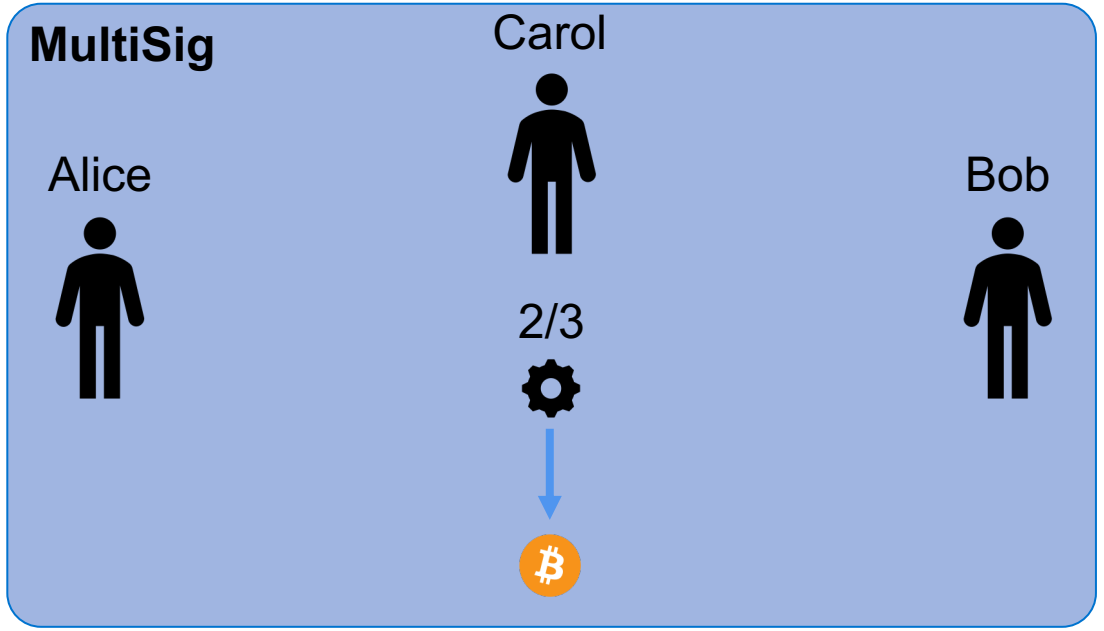
6. Storing Bitcoins

This chapter is heavily inspired by and uses examples from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan

MultiSig Address

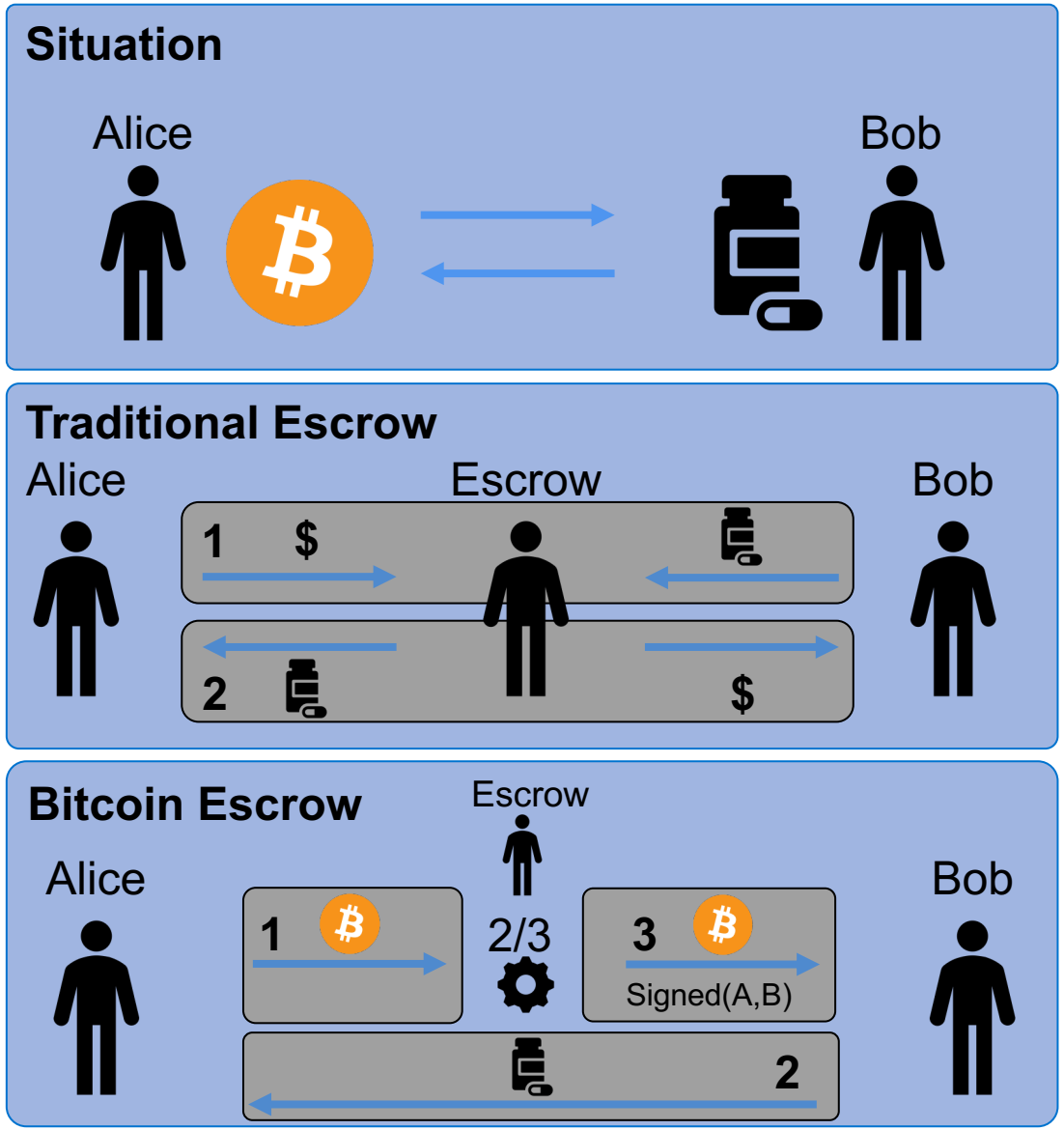
- A N/M multisig address is an address which has following properties
  - It requires more than one signature
  - N defines the number of required signatures
  - M defines the number of maximum signatures

- A 2/3 multisig address requires two arbitrary signatures from a pool of three predefined signatures.



Escrow

- Alice and Bob want to do business together. Alice wants to buy some real-world product from Bob with Bitcoin. However, Alice does not want to send the Bitcoin before having the physical good, but Bob does not want to send the physical good before having the Bitcoin in his wallet.
- What to do? In traditional systems, an escrow would enable the transaction. The escrow would receive both the good and the money and exchange them.
- With Bitcoin (or other cryptocurrencies), this process can be enhanced. Alice creates a 2/3 multisig-address and transfers the money to it. Bob can now safely send his good.
- When the good arrives, both Alice and Bob can sign the transaction and Bob receives his money.
  → If the exchange is correctly executed, the escrow is not involved.



Micropayments

Alice wants to continuously use a service from Bob and pay a certain small amount of Bitcoin for each usage.

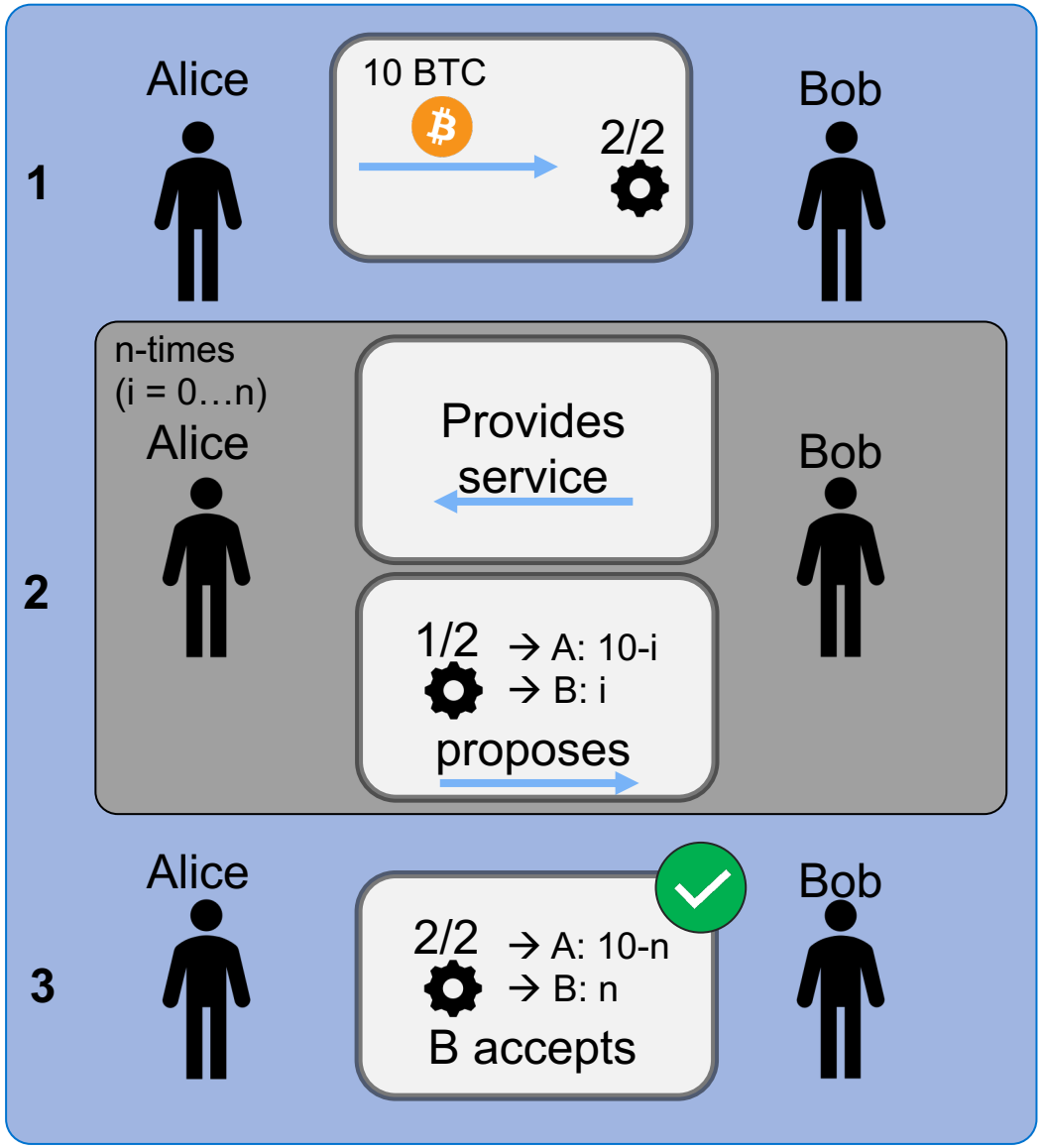
Creating new transactions every time won't work as:

- Too many transactions get created
- The fees for the transactions would be too high

An approach would be following:

- At the beginning, Alice send her maximum spendable amount to a 2/2 multisig address.
- Every time she consumes the service by Bob she signs a transaction locally for the multisig address, sending the accumulated amount to Bob and the change to herself. A message containing this incomplete transaction[1] is sent to Bob.
- If she is finished using the service, Bob will sign the last transaction he received and publish it on the network.

→ What is the problem?



[1] The transaction at this moment is rejected by the network, because the second signature is missing.

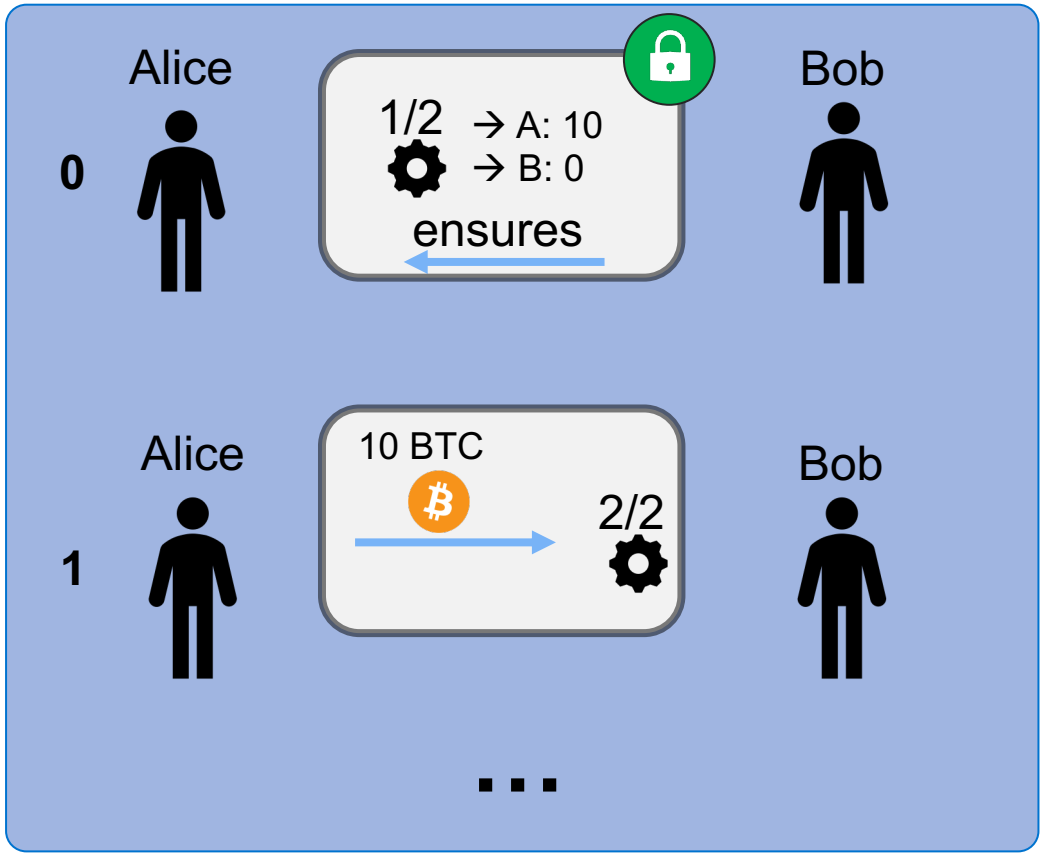
Lock Time

The problem is that Bob could decide **not to sign** the last transaction (or any transaction at all), leaving Alice's money stuck inside the multisig address.

However, there is a way around it:

In order to ensure that Alice receives her money back, she can demand from Bob to create a transaction from the multisig-address, transferring the complete funds back to Alice before creating it. This transaction contains the lock time[1] as the transaction should only be spendable after a certain amount of time *t* and if the output is not spent otherwise.

With this, it is safe to create a multisig address without having to worry if one receives the money back.



[1]As there is no notion of time in Bitcoin, the lock time is given as the block height. Until this block height is reached, the transaction is invalid.

Outline

1. Introduction to Bitcoin & Blockchain

2. Bitcoin Architecture
- Blockchain & Blocks
- Block Header & Contents
- Genesis Block

3. Transactions in Bitcoin
- Data Structure
- Transaction Validation

4. Bitcoin Network
- P2P Network
- Types of Nodes

5. Bitcoin Use Cases

6. Storing Bitcoins

This chapter is heavily inspired by and uses examples from „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

Storing Bitcoins

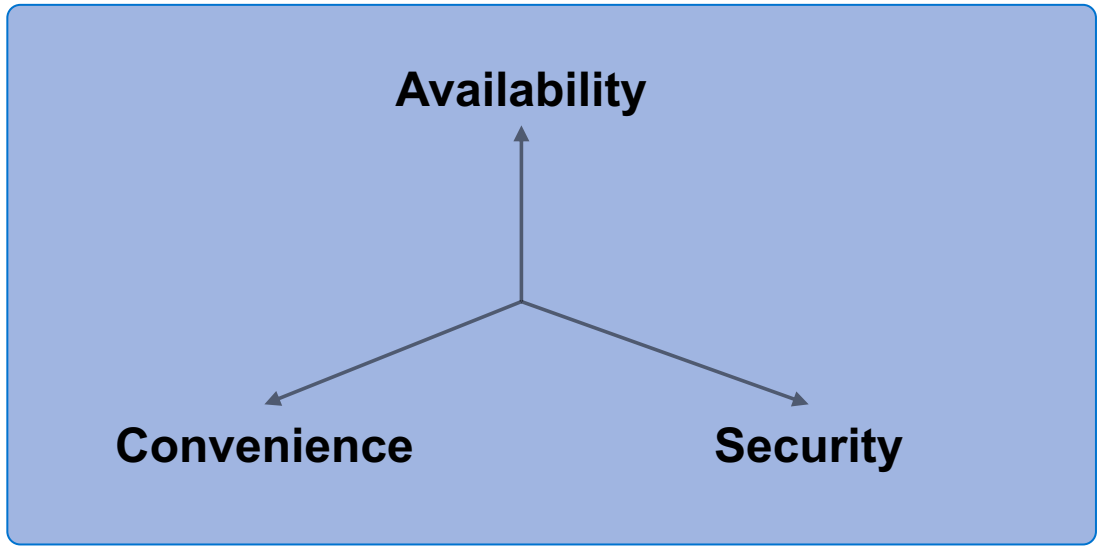
Storing Bitcoins is all about storing and managing secret keys.[1]

Different approaches for storing and managing secret keys lead to different trade-offs between **availability**, **security** and **convenience**.

Availability: being able to access the keys when one wants to

Security: restricting access to the keys

Convenience: easy use



Of course: The simplest approach is to store the secret key on one's hard drive. What could *possibly* happen?

[1]This is not only important for Bitcoin, but for every blockchain technology in this lecture.

Cryptocurrency Wallets

Cold Storage

- Takes some time to “activate”
- **Enhances security** at the cost of convenience and availability
- Advantage: Cold Storage does not have to be online to receive coins

Hot Storage

- Is **immediately available**
- Enables **convenience** and **availability** at the cost of security
- Example: Storage on your pc / mobile

Wallet Types

Brain Wallet

- A brain wallet stores Bitcoins with nothing but a secret passphrase.
- There is no need for hard drives, paper or else to store information.

- Idea: a deterministic function to generate a private key out of a passphrase.

- However:
  - Source of randomness determines the security → offline guessing / password cracking
  - If passphrase is forgotten, bitcoins are lost forever

- Example:
  „witch collapse practice feed shame open despair creek road again ice least"

Paper Wallet

- **Key material** is **printed** to paper. Paper can be placed in secure places like safes or vaults.

- However, keep in mind:
  - **Source** of **randomness**
  - Side-Channel attacks
    - Infected computer / malware
    - Malicious paper-wallet generator
    - Monitored printer
  - Durability of paper
  - Durability of ink
  - Secure place: dark, 16-19°C, low humidity

Bitcoin Paper Wallet

Wallet Types (cont.)

Hardware Wallet

- **Key material** is stored on the **hardware device**.
- Additional: Generation of a 24-word passphrase. If device gets destroyed, the passphrase allows for a recovery.
- Device is designed to keep your private key private. Key is **securely stored** within the device.
- **The display shows the amount and target of a transaction.** The **keys** on the device allow for a **confirmation or rejection** of the transactions.
- Requires trust in manufacturer and intermediaries. (Never buy used hardware wallets!)

Hardware Wallet - Ledger Nano S

Online Wallet

- Let other people/ companies store your Bitcoins/ cryptocurrencies for you (custodial wallet).
- No access to the private key, coins can only used through a certain interface / website.
- Very common within most exchanges. The money is sent to the exchange, the account on the platform has now a new balance which can be traded or paid out.
- However: Very dangerous! Many exchanges got hacked, users lost their funds. Be careful!

MT.GOX coinbase BINANCE