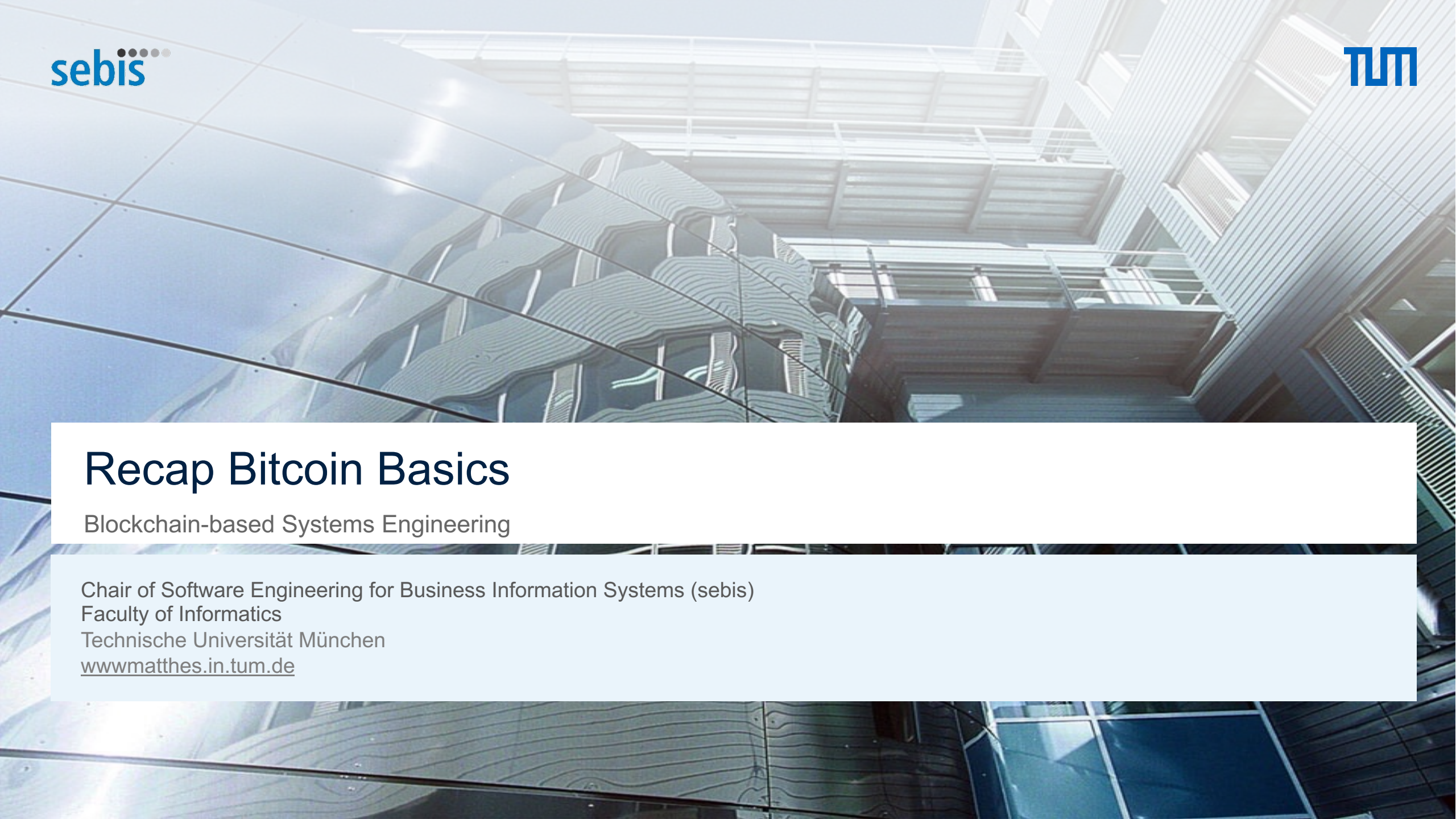
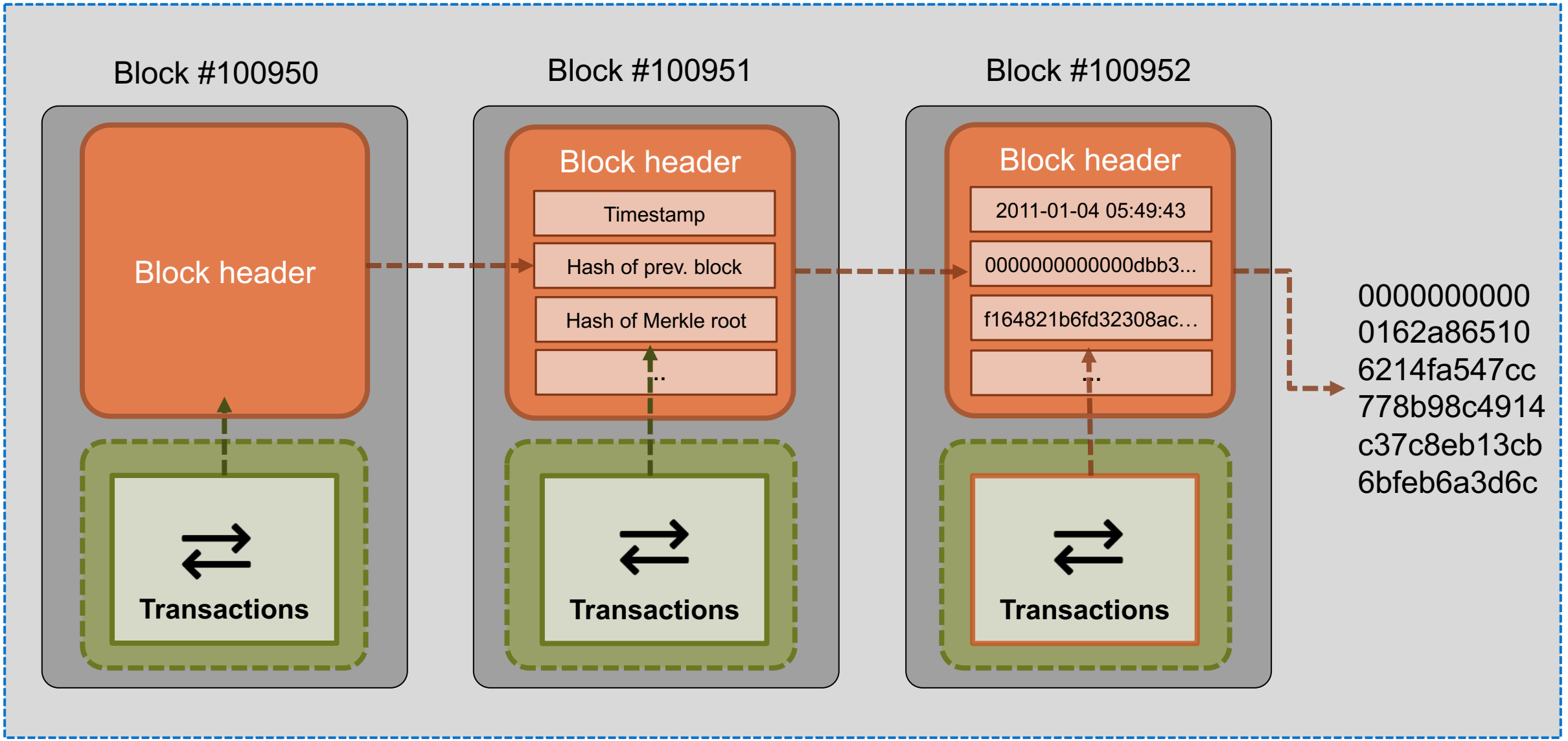


Recap Bitcoin Basics

Blockchain-based Systems Engineering

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

Bitcoin Blockchain



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

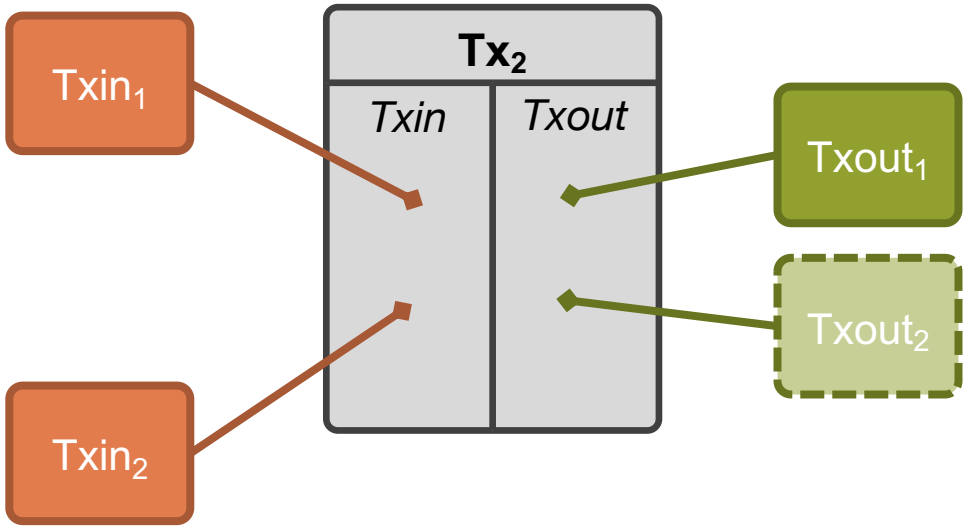
Alice	Bob	Carol
25	0	0
8	17	0
8	9	8
13	9	3
-2	24	3

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

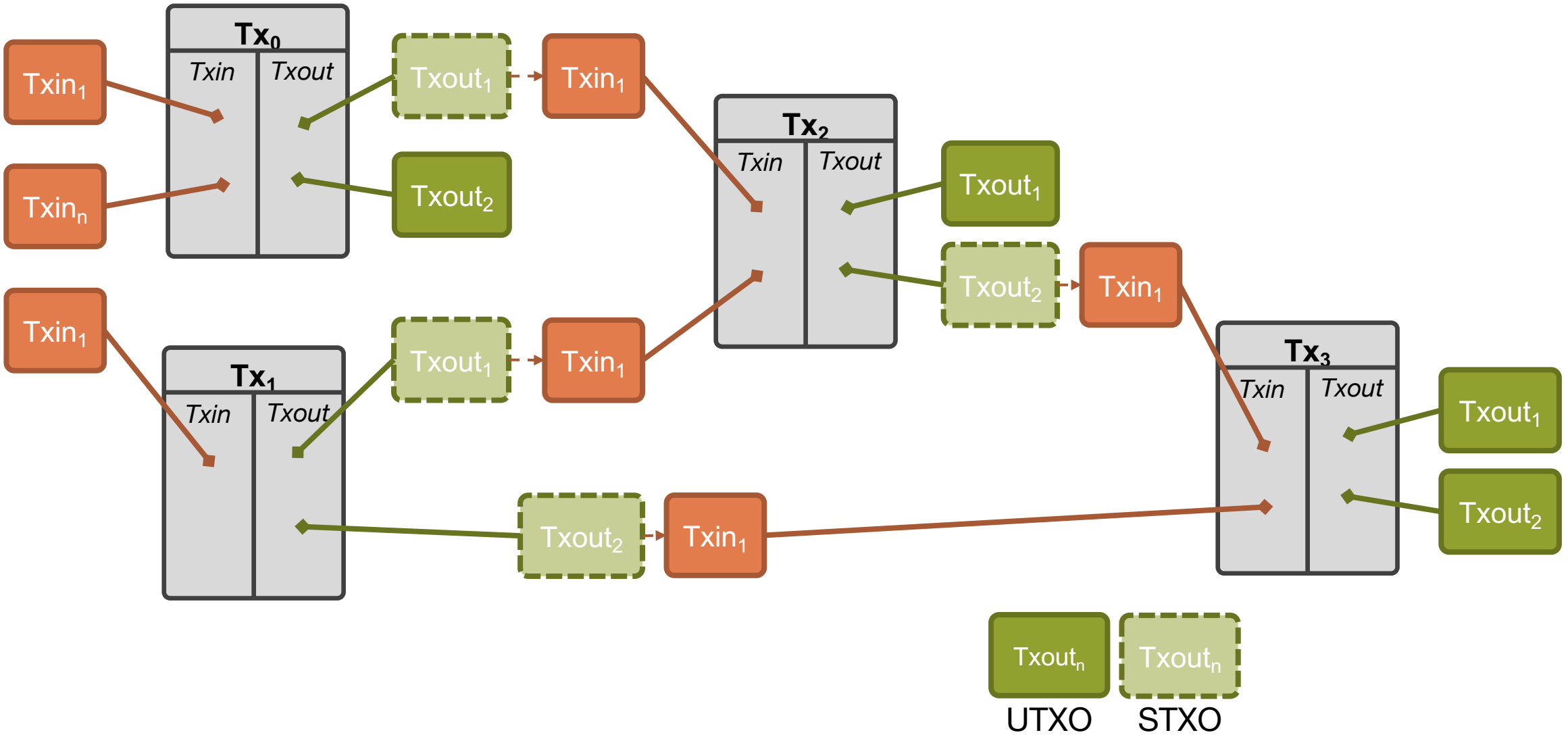


- Transactions (**Tx**) have a number of inputs and a number of outputs.
  - **Inputs (Txin)**: Former outputs, that are being consumed
  - **Outputs (Txout)**: Creation of new coins and transfer of coin ownership

- 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.

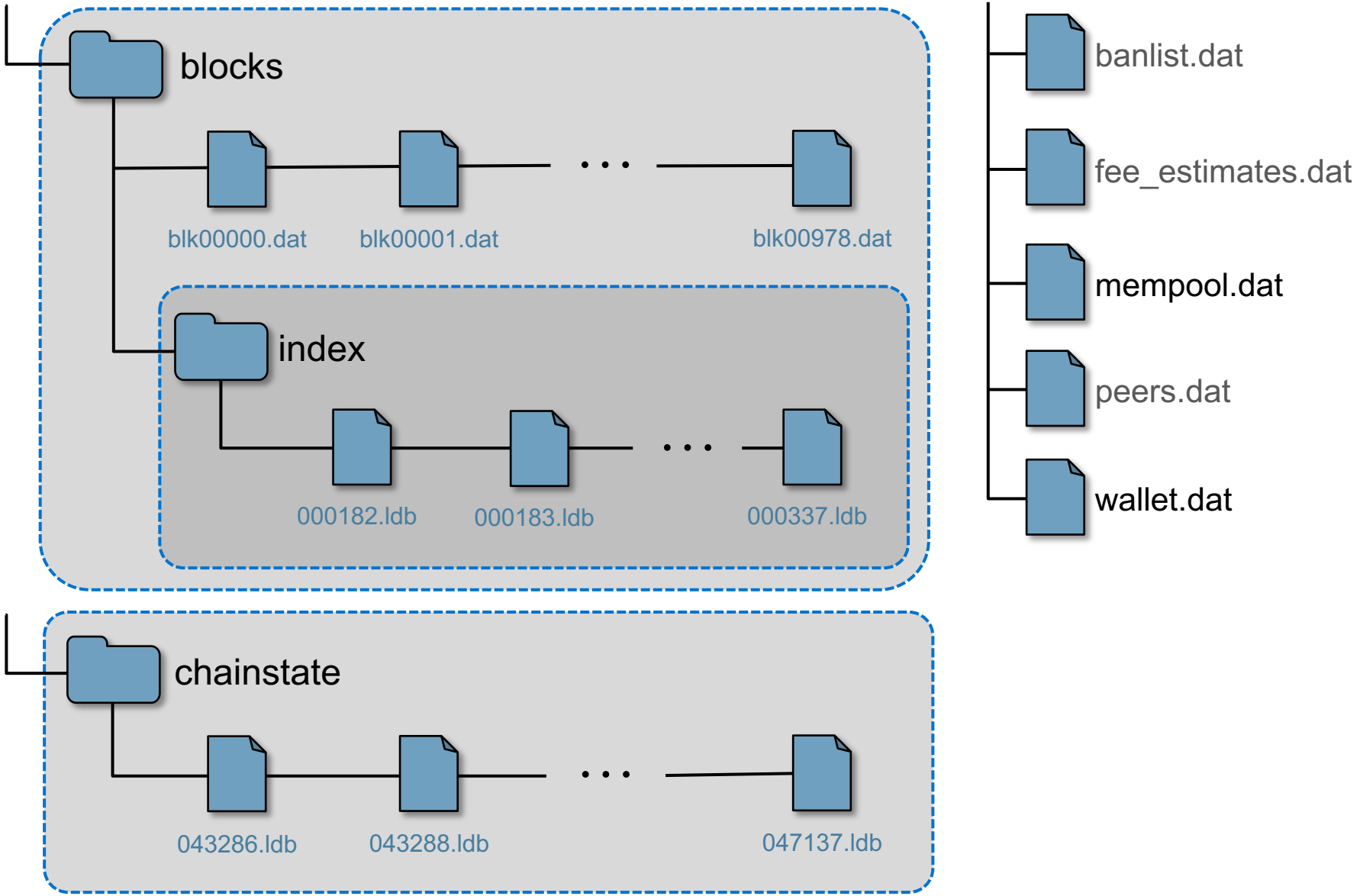
Joint payments:
Two or more parties can combine their inputs and create one output. Of course, it requires signatures from all involved parties.

Anatomy of the Bitcoin Blockchain

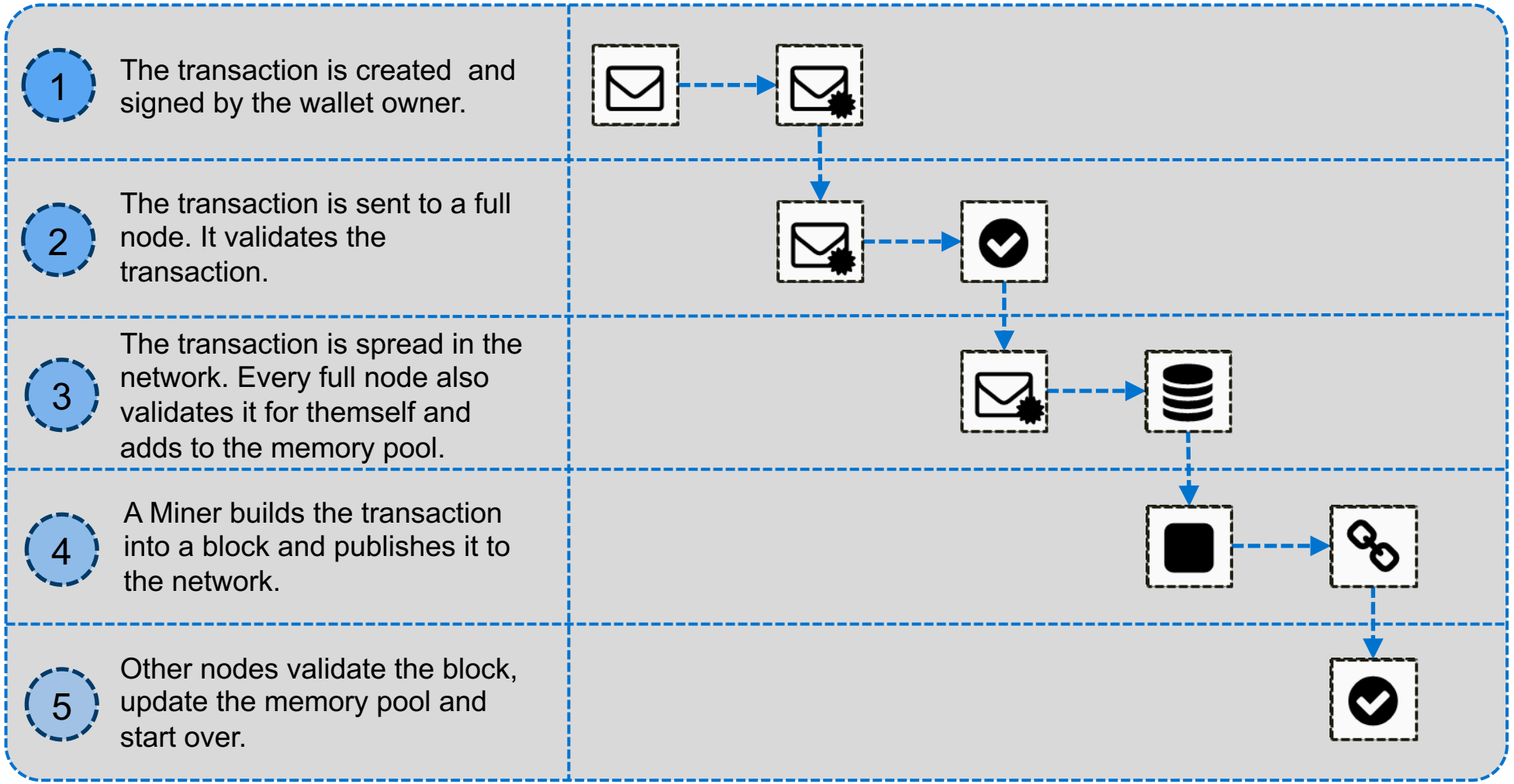
Raw data is on a disk.

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.



A Newly Created Transactions Way into a Block



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

Storing Bitcoins

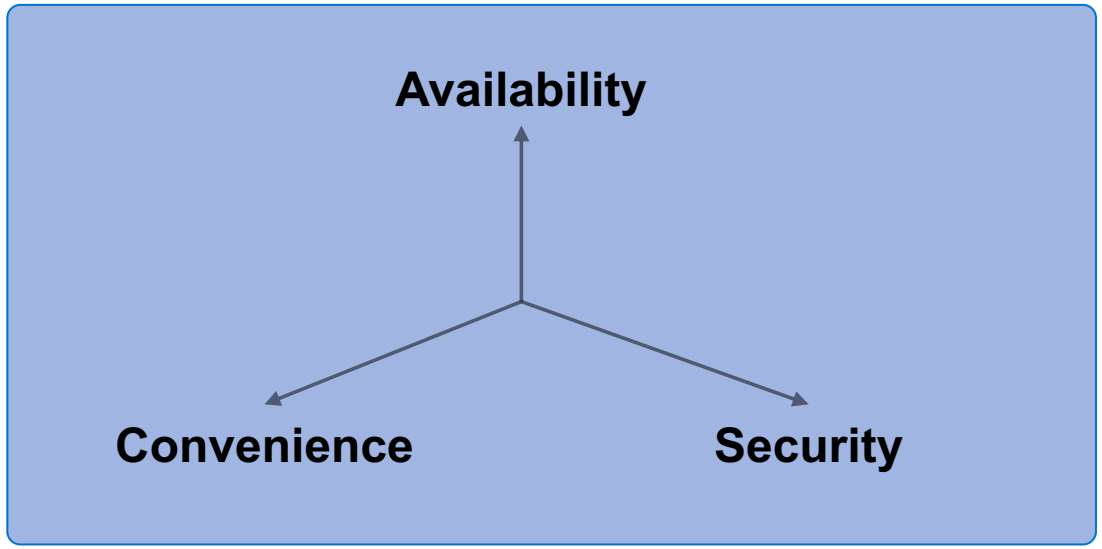
Storing Bitcoins is all about storing and managing secret keys.[1]

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** at the cost of availability and security
- Example: Storage on your pc / mobile

Useful Resources

- mempool.space
  - A block and mempool explorer for Bitcoin
  - Read this article to better understand how to use the tool

- Jochoe's Bitcoin mempool
  - Another mempool explorer with useful insights

- Blockchair, Blockchain.com
  - Some other blockchain explorers