

Introduction to Blockchains

33c3

Niklaus 'vimja' Hofer
niklaus@mykolab.ch

January 1, 2017

```
4946 454c 743d 6865 6c70 7461 8a65 4458
5646 4549 4557 2832 3a3a 2428 732d 6568
6c6c 7728 6968 6863 6a20 7975 6864 8a29
618a 6c6c 283a 2824 4946 454c 2a23 6470
8a66 2a8a 4858 4ee1 3a55 6320 656e 6e61
638a 656c 6e61 8a3a 5c89 6972 2a29 612e
7975 2a29 6c2e 676f 2a28 6a2e 7861 2a29
617e 7475 2a29 732e 686a 2a28 742a 636f
8a6a 582e 4f48 594e 283a 6964 7473 6c63
6165 8a6e 6964 7473 6c63 6165 3a6e 096a
75c 286d 2824 4946 4558 2a29 6470 6165
283a 6976 7765 768a 6569 3a77 2829 2a28
4828 4c49 2945 782e 6864 696a 245c 5828
```

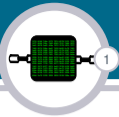
```
2f74 6f66 746e 2f73 7974 6570 2f31 7275
2f77 6568 786c 7465 6369 752f 7668 3872
2e61 6e70 3a62 4f8a 7475 7570 2074 7277
7469 6574 296e 6e6f 7420 6865 6c70 7461
2e65 6470 2966 3128 2832 6170 6567 2c73
3328 3235 3838 2932 7962 6574 2973 8a2e
4458 2946 7473 7461 7369 6974 7363 8a3a
3228 3131 5820 4644 6f20 6a62 6365 7374
6f20 7475 6f20 2966 3831 3838 2828 616d
2e78 3820 3833 3638 3738 6a29 3128 3837
6328 6d6f 7270 7365 6573 2864 626f 656e
736e 6f20 7475 6f20 2966 3831 3838 2828
616d 2e78 3528 3838 3838 2938 286e 3431
```

```
7265 7348 6365 6974 6e6f 6170 6567 2073
3570 707d 7d38 7d7d 5c8a 7740 6972 6574
6966 656c 6e7b 7661 7b70 685c 6165 6364
6d6f 616d 646e 7b20 625c 6165 656d 4072
7573 7362 6365 6974 6e6f 6170 6567 2073
387b 7b7d 7d38 7d7d 5c8a 7740 6972 6574
6966 656c 747b 636f 7b70 625c 6165 656d
4072 7573 7362 6365 6974 6e6f 6e69 6f74
2963 3370 7b7d 7d31 4c7b 616f 6964 676e
7420 6568 5420 6568 656d 6120 646e 5420
6568 656d 4f20 7470 6f69 736e 7b7d 7d39
387b 7b7d 7d31 0a7d 485c 7277 7469 6665
6c69 7865 616e 7d7e 5c7b 6568 6461 6f63
6d6d 6e61 2964 5c7b 6562 6d61 7265 7348
6275 6573 7463 6f69 656e 746e 7972 7628
7d38 337b 7b7d 7d31 397b 7b7d 6f4c 6461
```

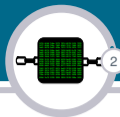
```
8a2e 5c8a 6674 7448 636f 5c3d 7277 7469
3765 5c8a 786f 6e65 756f 3774 3a20 6828
6574 786d 616c 6574 742e 636f 2a27 8a8a
745c 4866 6a73 3a6d 775c 6972 6574 8a38
6f5c 6570 6f6e 7475 2038 283d 7460 6865
6c70 7461 2a65 6a73 276d 8a2e 588a 6361
618a 656f 6120 7b74 7265 6579 646e 4928
9a6e 3a6f 4520 796d 7974 6820 6f67 286b
4288 6865 726f 4385 656c 7261 6f44 7563
6584 746e 2a27 6e6f 6928 796e 7475 6c20
6e69 2965 3432 2a32 588a 6361 616b 6567
6128 7b74 7265 6579 646e 4928 8a6e 3a6f
4520 796d 7974 6820 6f67 298a 4160 746e
7265 614c 7473 6053 7089 756f 2774 6f20
2a6e 6e69 7570 2074 696c 656e 3220 3234
2764 6f20 286e 6e69 7570 2074 696c 656e
```

```
7372 6f69 286e 337b 332e 7833 7d74 8a7d
485c 7277 7469 6665 6c69 7865 616e 7d7e
5c7b 6562 6d61 7265 6548 646e 8e69 7570
6974 6f66 8874 7265 6576 7372 6f69 286e
337b 332e 7833 7d74 8a7d 735c 6c65 6365
4074 616c 676e 6175 6567 657b 8f6e 896c
6873 8a70 485c 7277 7469 6665 6c69 7865
6f74 7863 5c7b 6573 658c 7463 6c48 6e61
756f 6761 7865 6e65 6c67 7369 7868 8a7d
485c 7277 7469 6665 6c69 7865 6f6c 7a6e
5c7b 6573 656c 7463 6c48 6e61 756f 6761
7865 6e65 6c67 7369 7868 8a7d 485c 7277
7469 6665 6c69 7865 6f6c 7d74 5c7b 6573
656c 7483 6c48 6e61 756f 6761 7865 6e65
6c67 7369 7468 8a7d 485c 7277 7469 6665
6c69 7865 616e 7d7e 5c7b 6568 6461 6f63
```

```
8665 6c69 7865 616e 7d7e 5c7b 6568 6461
6f63 686d 6e61 2964 5c7b 6562 6d61 7265
7340 6275 6573 7463 6f69 786e 6761 7365
7b20 7d33 347b 7d7d 8a7d 485c 7277 7469
6665 6c69 7865 6f74 7863 5c7b 6562 6d61
7265 7340 6275 6573 7463 6f69 696e 746e
7b20 7d32 317b 7b7d 6f53 7275 6563
7b20 6c69 7365 7b7d 7d35 387b 7b7d 7d32
8a7d 485c 7277 7469 6665 6c69 7865 616e
7d7e 5c7b 6568 6461 6f63 6d6d 6e61 2864
5c7b 6562 6d61 7265 7340 6275 6573 7463
6f69 656e 746e 7972 7b20 7d38 327b 7b7d
7d31 357b 7b7d 6f53 7275 6563 6a20 6c69
7365 7d7d 685c 6165 6364 6e6f 616d 646e
7b20 625c 6165 656d 4072 7573 7362 6385
6974 6e6f 6170 6567 2073 357b 7b7d 7d34
```



- ▶ IT student at BFH
 - ▶ Specialisation in Infosec
 - ▶ Been working with Blockchain Technology for 1.5 years
 - ▶ Bachelor thesis on the analysis of the Bitcoin Blockchain
- ▶ Founding member of the Chaostreff Bern



Systems for representing ownership

- State-transition systems
- Double-spend

Blockchain (without PoW)

- The underlying network
- Establishing a Consensus
- Double-spend on Blockchains

Blockchain

- PoW and mining
- Solving double-spend

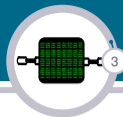
Bitcoin

- Blocks
- Light clients

Bonus Slides

- Branches in the chain
- Double-spend

Content II

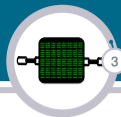


Mining



Section 1

Systems for representing ownership



Systems for representing ownership

State-transition systems

Double-spend

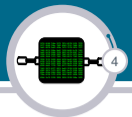
Blockchain (without PoW)

Blockchain

Bitcoin

Bonus Slides

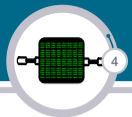
Systems of ownership as state-transition-systems



Systems representing ownership can be modelled as state-transition system.

- ▶ Finance
- ▶ Estate
- ▶ ...

Systems of ownership as state-transition-systems



Systems representing ownership can be modelled as state-transition system.

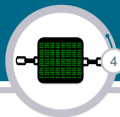
- ▶ Finance
- ▶ Estate
- ▶ ...

Mapping:

State Collection of who owns what

Transition Transferring ownership to someone else

Systems of ownership as state-transition-systems



Systems representing ownership can be modelled as state-transition system.

- ▶ Finance
- ▶ Estate
- ▶ ...

Mapping:

State Collection of who owns what

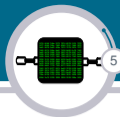
Transition Transferring ownership to someone else

Example for financial system:

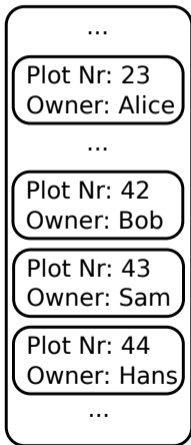
State Collection of all accounts

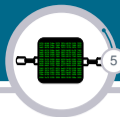
Account Owner and associated amount

Transition Transaction (Moving value from one account to another)

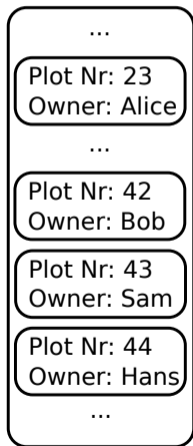


State A



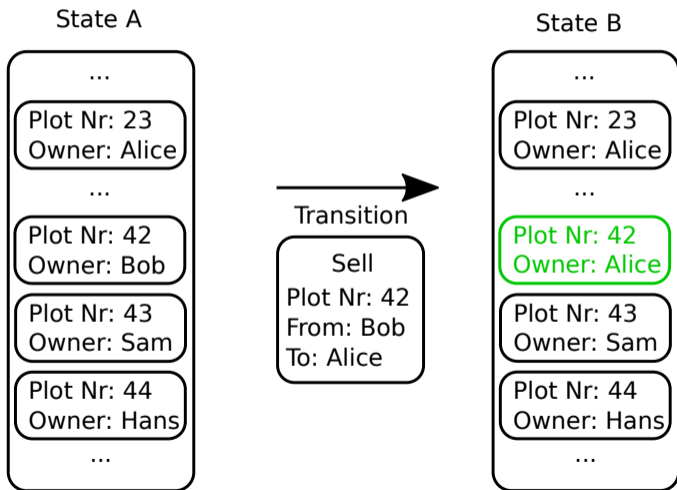
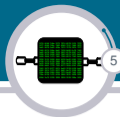


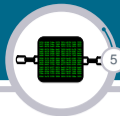
State A



Transition







Systems for representing ownership

State-transition systems

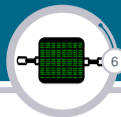
Double-spend

Blockchain (without PoW)

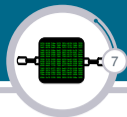
Blockchain

Bitcoin

Bonus Slides



- ▶ Consensus is very important
- ▶ Before each transaction, all parties need to agree on the current state!
- ▶ The reason for this: double-spend



- ▶ To spend something twice
 - ▶ Spending the same money twice
 - ▶ Selling the same plot twice
 - ▶ ...
- ▶ Obviously malicious
- ▶ Well known attack in the Blockchain / Bitcoin world

Double-spend estate example

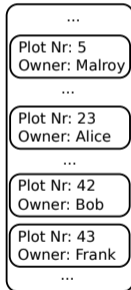


- ▶ Malroy has a nice property (Plot number 5)
- ▶ Alice is looking to buy a new property
- ▶ Bob also wants to buy a new property
- ▶ Malroy will attempt to sell the same plot to both of them!

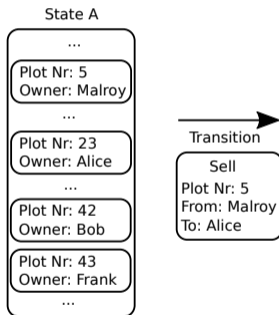
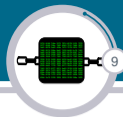
Example: Alice's view



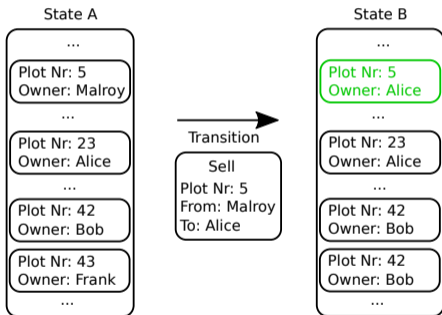
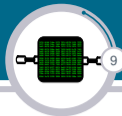
State A



Example: Alice's view

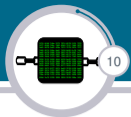


Example: Alice's view

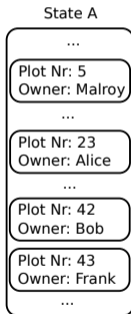


Alice has paid for the plot. She thinks it's now hers.

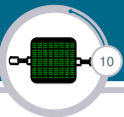
Example: Bob's view



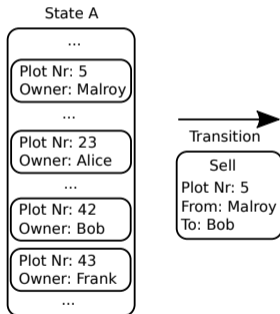
- ▶ Bob does not know about the transfer of ownership from Malroy to Alice
- ▶ He still thinks the state looks like this:



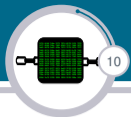
Example: Bob's view



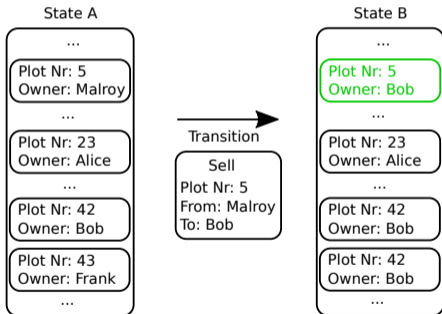
- ▶ Bob does not know about the transfer of ownership from Malroy to Alice



Example: Bob's view

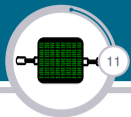


- ▶ Bob does not know about the transfer of ownership from Malroy to Alice

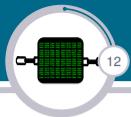


Bob has paid for the plot. He thinks it's now his.

Example: problem

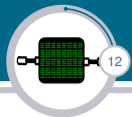


- ▶ Problem: Malroy sold the same plot twice!
- ▶ Alice and Bob do not agree on the current state of the system
- ▶ Their views on the state are incompatible
- ▶ This breaks the system:
 - ▶ Possibility to sell plots many times
 - ▶ People can't trade with each other



Simple solution in the estate world:

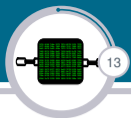
- ▶ Registry of deeds (Grundbuchamt)
- ▶ Central authority
- ▶ Controls the state of the system
- ▶ Every time an estate is sold (a transition is made), it has to be done via the registry of deeds
- ▶ For each transition, the central authority performs certain checks to make sure the transition is compatible with the current state and either accepts or rejects it



Simple solution in the estate world:

- ▶ Registry of deeds (Grundbuchamt)
- ▶ Central authority
- ▶ Controls the state of the system
- ▶ Every time an estate is sold (a transition is made), it has to be done via the registry of deeds
- ▶ For each transition, the central authority performs certain checks to make sure the transition is compatible with the current state and either accepts or rejects it

That way, everybody can agree on a certain state and that state is always valid.

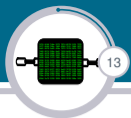


Requirements:

- ▶ All parties need to agree on the current state
- ▶ All parties need to agree on whether a transition is valid

Banks as central authority:

- ▶ Banks serve as central authorities
- ▶ They control the state and check all transitions
- ▶ Always the case for modern day money transfers



Requirements:

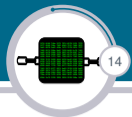
- ▶ All parties need to agree on the current state
- ▶ All parties need to agree on whether a transition is valid

Banks as central authority:

- ▶ Banks serve as central authorities
- ▶ They control the state and check all transitions
- ▶ Always the case for modern day money transfers

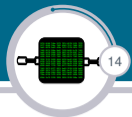
This works surprisingly well.

A Solution for the internet



Requirements:

- ▶ Decentralized
- ▶ Needs to work on the internet

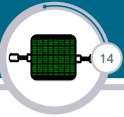


Requirements:

- ▶ Decentralized
- ▶ Needs to work on the internet

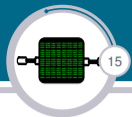
Central authority:

- ▶ Possibility of censorship
- ▶ Can be attacked
- ▶ Collects all the data



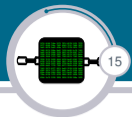
Section 2

Blockchain (without PoW)



Blockchain is an internet-age solution to the same problem. It provides these (amazing) properties:

- ▶ No central authority
- ▶ Parties do not need to trust each other
- ▶ Parties need no information on who is participating
- ▶ ... not even any information on how many others are participating
- ▶ And still they can all agree on a state



Systems for representing ownership

Blockchain (without PoW)

The underlying network

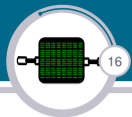
Establishing a Consensus

Double-spend on Blockchains

Blockchain

Bitcoin

Bonus Slides



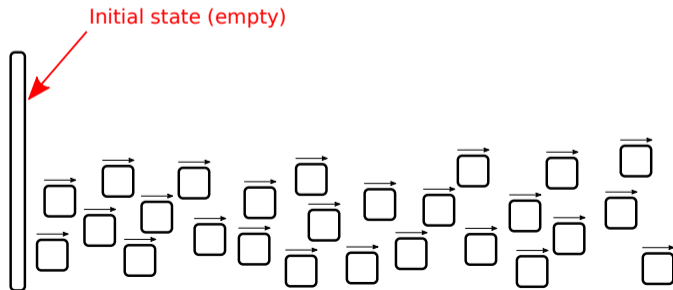
- ▶ We all agree on an initial state
 - ▶ Could be an empty state
 - ▶ Or something else
- ▶ We build a peer to peer network
- ▶ Transactions are published / announced to the p2p network
- ▶ Network relays transactions

Distributing transactions

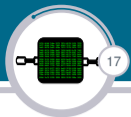


All sorts of problems:

- ▶ Network out of sync

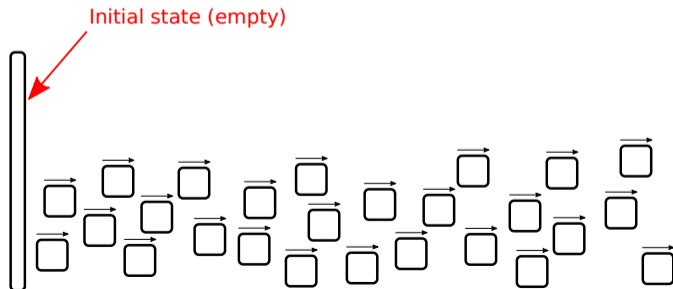


Distributing transactions



All sorts of problems:

- ▶ Network out of sync
- ▶ Order unclear

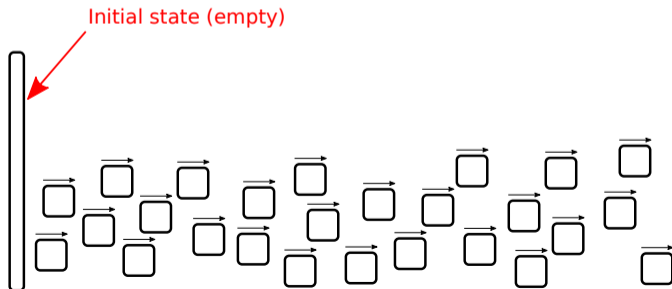


Distributing transactions

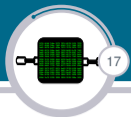


All sorts of problems:

- ▶ Network out of sync
- ▶ Order unclear
- ▶ Double-spend attempts

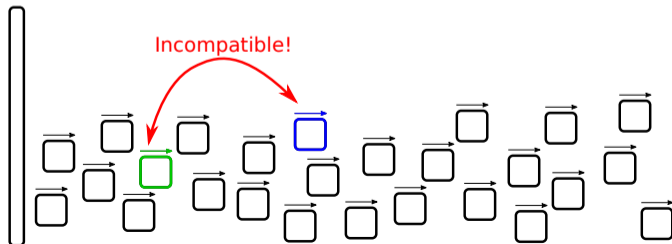


Distributing transactions



All sorts of problems:

- ▶ Network out of sync
- ▶ Order unclear
- ▶ Double-spend attempts
- ▶ Conflicting transactions

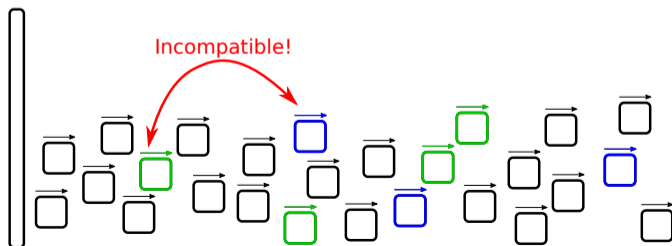


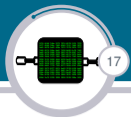
Distributing transactions



All sorts of problems:

- ▶ Network out of sync
- ▶ Order unclear
- ▶ Double-spend attempts
- ▶ Conflicting transactions
- ▶ Transactions dependant on conflicting transactions





Systems for representing ownership

Blockchain (without PoW)

The underlying network

Establishing a Consensus

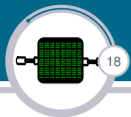
Double-spend on Blockchains

Blockchain

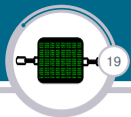
Bitcoin

Bonus Slides

Introducing: the Blockchain



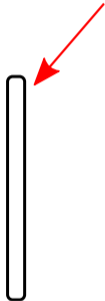
A method is needed to establish a consensus, to agree on a state.
We call this method: Blockchain
(For now: Without Proof of Work, PoW)



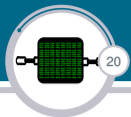
- ▶ Group transactions into blocks
- ▶ Blocks depend on one-another
- ▶ Anyone can form a new block at any time
- ▶ We define the current state as:
 - ▶ All transactions within the longest branch of blocks applied to the initial state
 - ▶ Transactions only become part of the state once they are inside a block
- ▶ Blocks have to meet certain criteria:
 - ▶ All transactions within the block must be valid
 - ▶ Transactions within the block have to be compatible with one another
 - ▶ Transactions within the block have to be compatible with transactions in earlier blocks



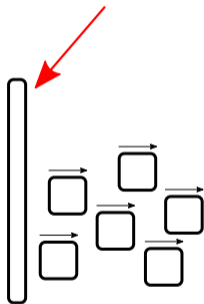
Initial state (empty)

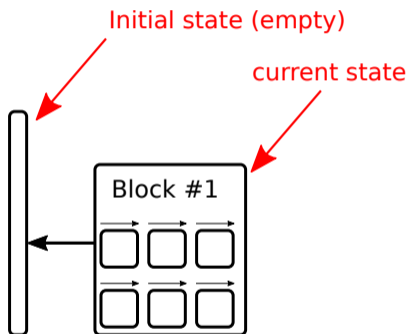
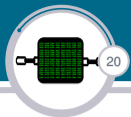


Visualization of Blockchain concepts

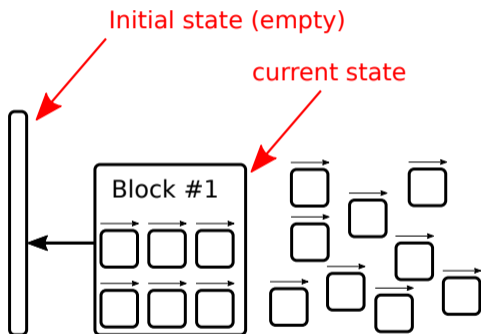
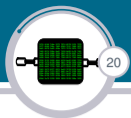


Initial state (empty)



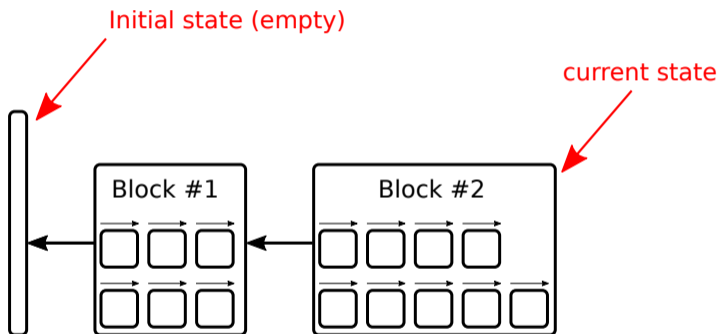
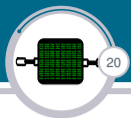


- ▶ Group transactions into blocks



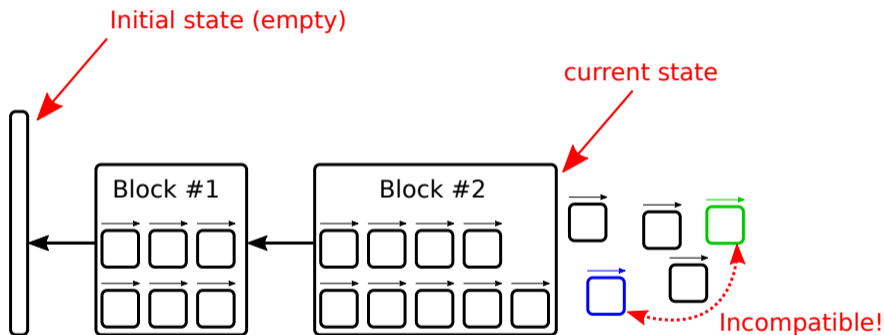
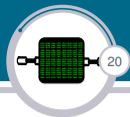
- ▶ Transactions only become part of the state once they are inside a block

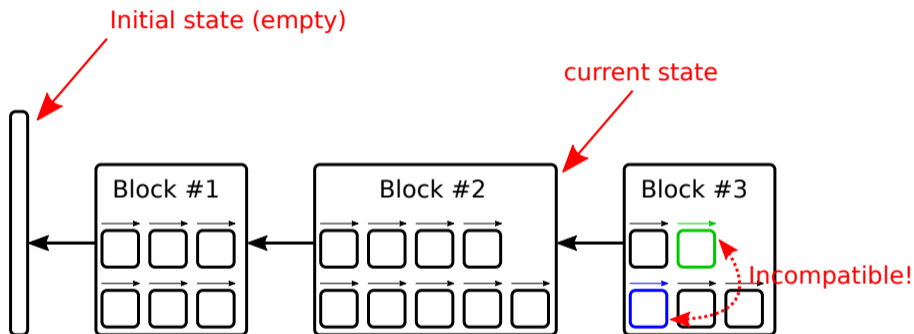
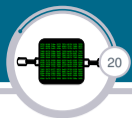
Visualization of Blockchain concepts



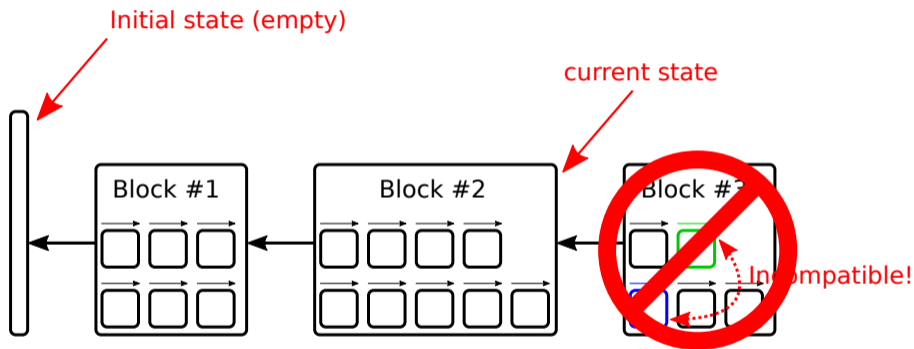
- ▶ Blocks depend on one-another
- ▶ Anyone can form a new block at any time

Visualization of Blockchain concepts



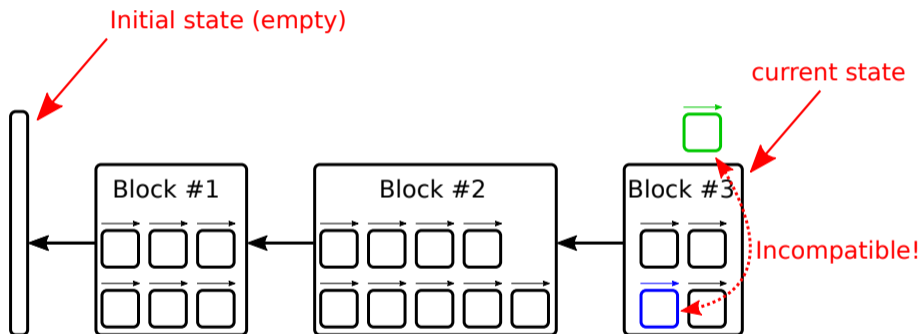
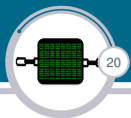


- ▶ Blocks have to meet certain criteria:
 - ▶ Transactions within the block have to be compatible with one another



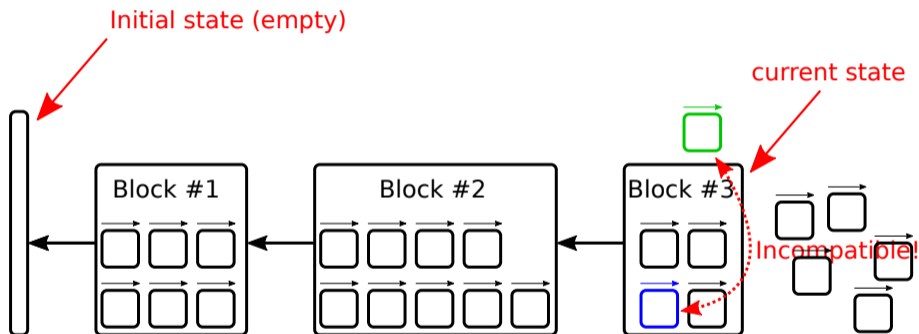
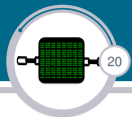
- ▶ Blocks have to meet certain criteria:
 - ▶ Transactions within the block have to be compatible with one another

Visualization of Blockchain concepts

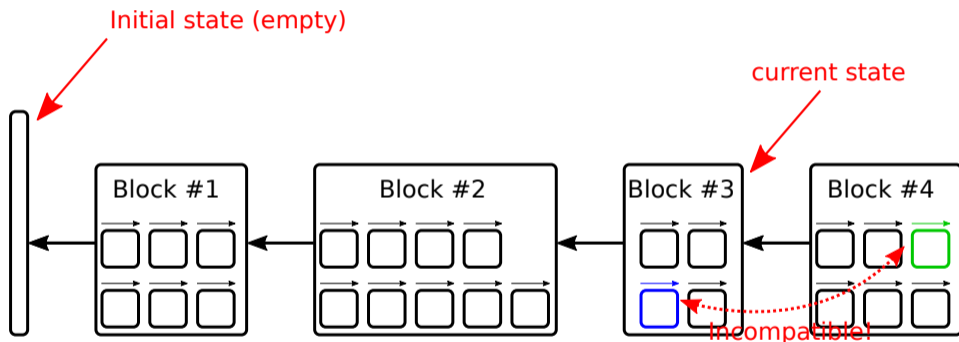
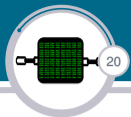


- ▶ Blocks have to meet certain criteria:
 - ▶ Transactions within the block have to be compatible with one another

Visualization of Blockchain concepts

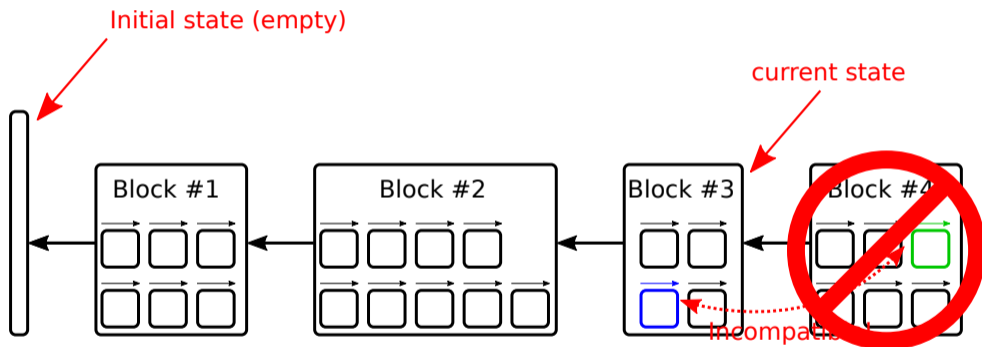
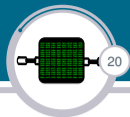


Visualization of Blockchain concepts



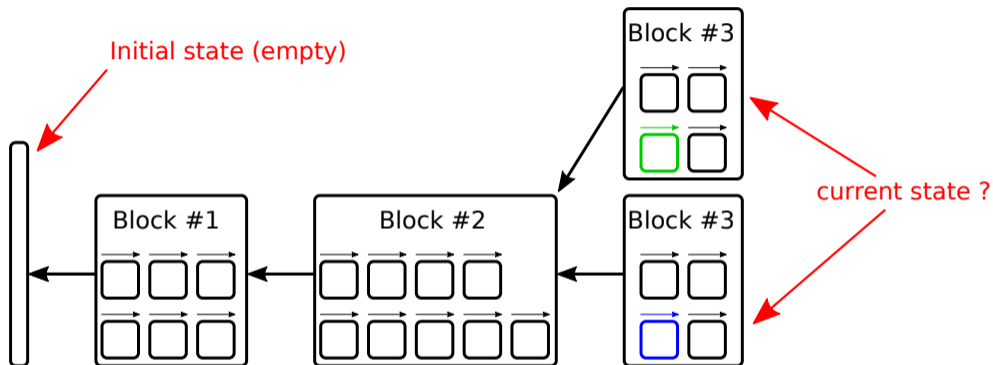
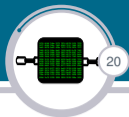
- ▶ Blocks have to meet certain criteria:
 - ▶ Transactions within the block have to be compatible with transactions in earlier blocks

Visualization of Blockchain concepts



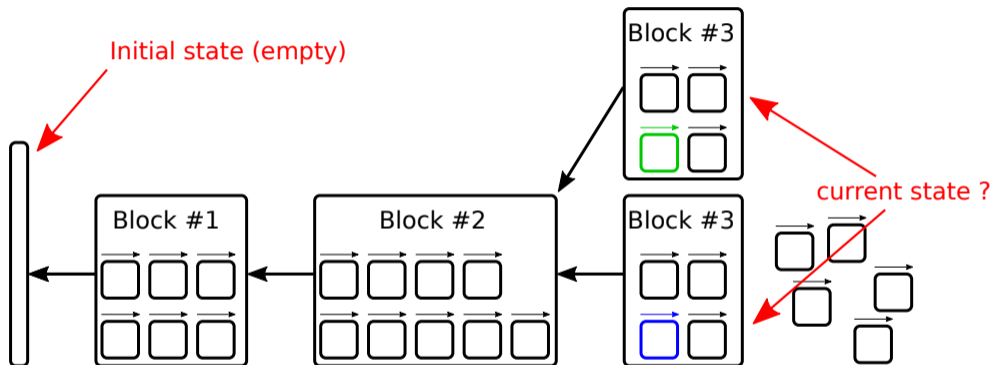
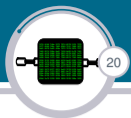
- ▶ Blocks have to meet certain criteria:
 - ▶ Transactions within the block have to be compatible with transactions in earlier blocks

Visualization of Blockchain concepts



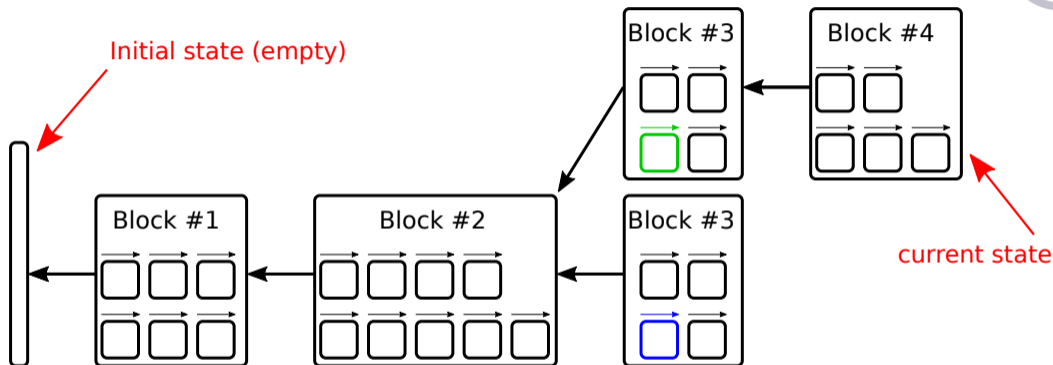
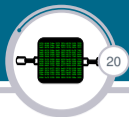
- We define the current state as: All transactions within the longest branch of blocks applied on an empty state

Visualization of Blockchain concepts

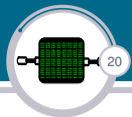


- We define the current state as: All transactions within the longest branch of blocks applied on an empty state

Visualization of Blockchain concepts



- We define the current state as: All transactions within the longest branch of blocks applied on an empty state



Systems for representing ownership

Blockchain (without PoW)

The underlying network

Establishing a Consensus

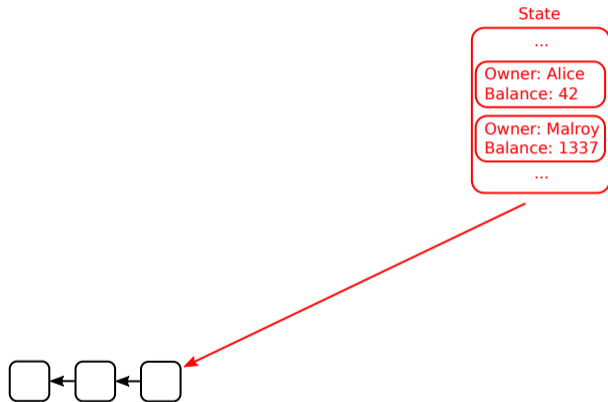
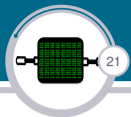
Double-spend on Blockchains

Blockchain

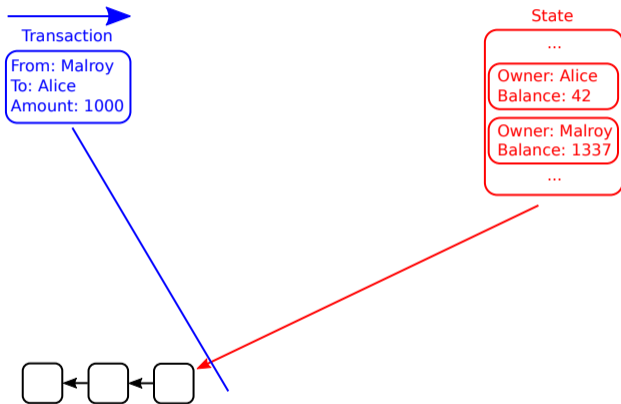
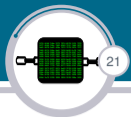
Bitcoin

Bonus Slides

Double-spend on a Blockchain without POW

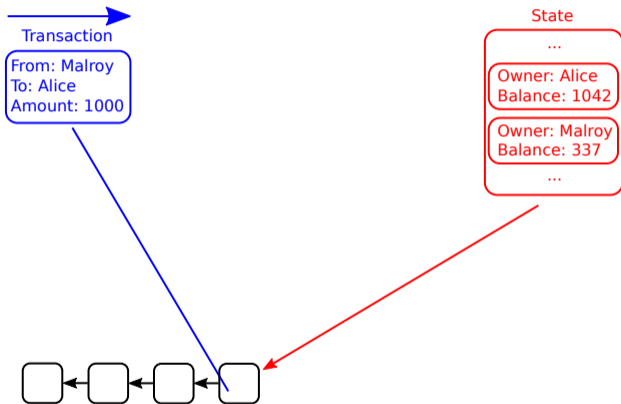
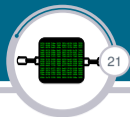


Double-spend on a Blockchain without POW



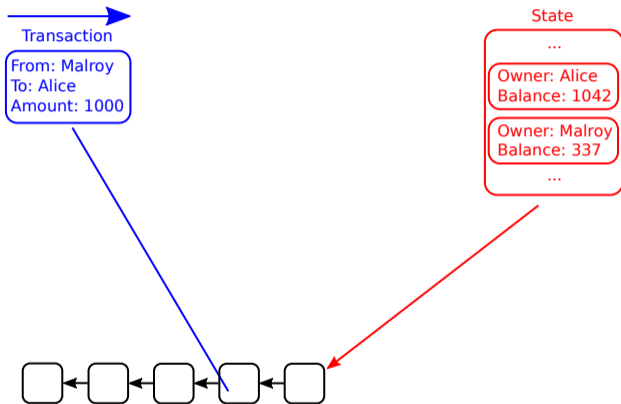
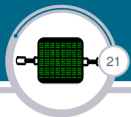
- ▶ Malroy creates the transaction to buy the bicycle

Double-spend on a Blockchain without POW

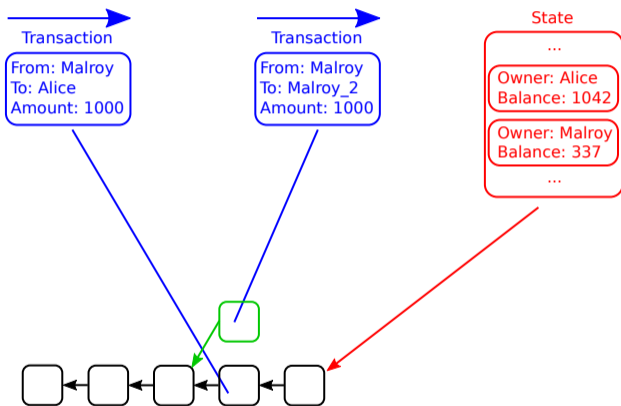
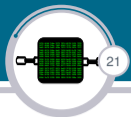


- ▶ The transaction becomes part of the state
- ▶ Alice now has the money
- ▶ Malroy gets the bicycle

Double-spend on a Blockchain without POW

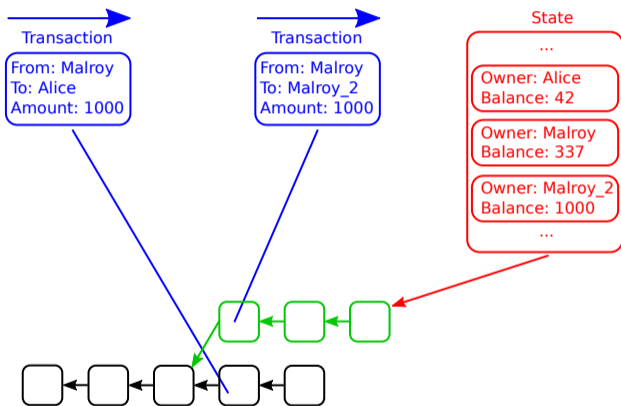


Double-spend on a Blockchain without POW



- ▶ Malroy creates a new block with a different transaction

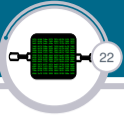
Double-spend on a Blockchain without POW



- ▶ Malroy creates a lot of blocks very quickly
- ▶ Malroy's branch becomes the state
- ▶ Alice does not have the money
- ▶ Malroy still has the bicycle

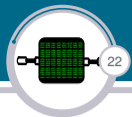


- ▶ Double-spend works
- ▶ It is easy and cheap to create blocks
 - ▶ Anyone can create any number of blocks at no cost



Section 3

Blockchain



Systems for representing ownership

Blockchain (without PoW)

Blockchain

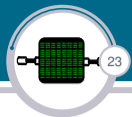
PoW and mining

Solving double-spend

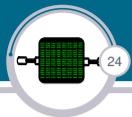
Bitcoin

Bonus Slides

The problem

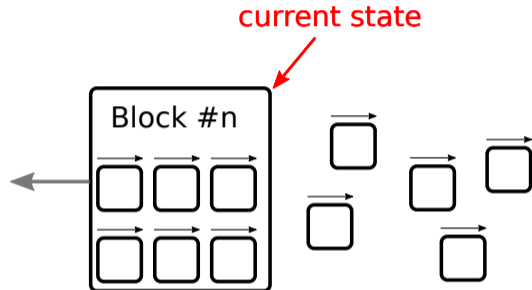
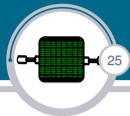


- ▶ Problem: Creating blocks is easy
- ▶ Solution: Make creating blocks hard

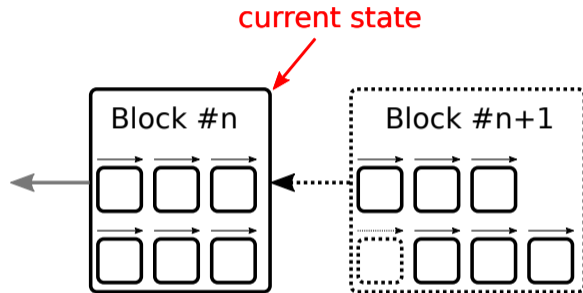
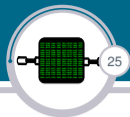


- ▶ Define a challenge taking a block as input
- ▶ For every new block created, the challenge has to be solved
 - ▶ Expensive in time and compute power
- ▶ The solution is called a Proof of Work (PoW)
 - ▶ The process of creating a PoW is called mining
- ▶ The PoW is to be published with the new block
- ▶ Only blocks with a valid PoW are valid blocks

Creating PoW protected blocks

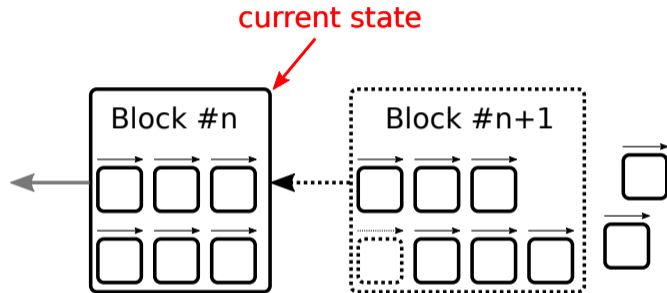
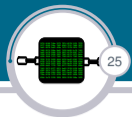


Creating PoW protected blocks



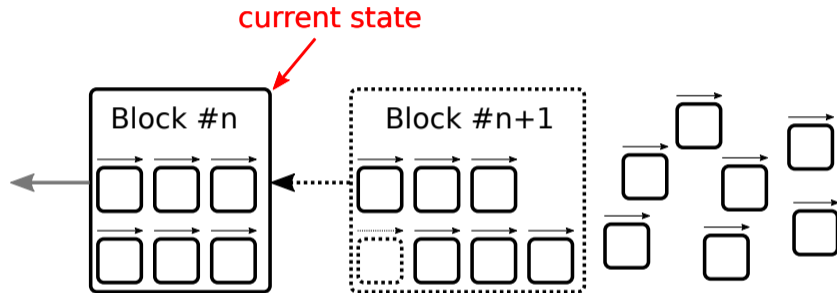
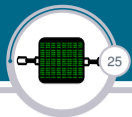
- ▶ (The additional transaction is a Coinbase (see below))

Creating PoW protected blocks



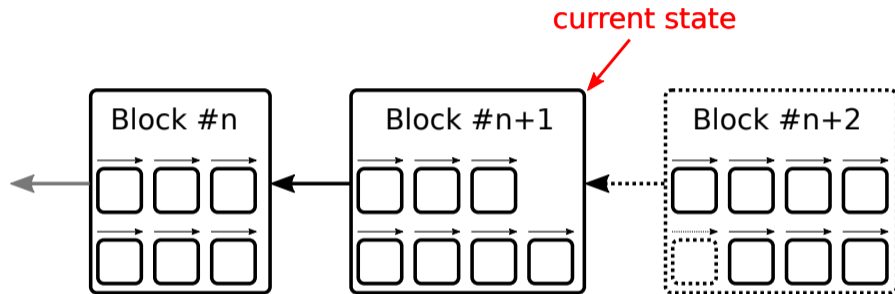
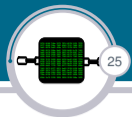
- ▶ Work on the block is going on
- ▶ New transactions arrive

Creating PoW protected blocks



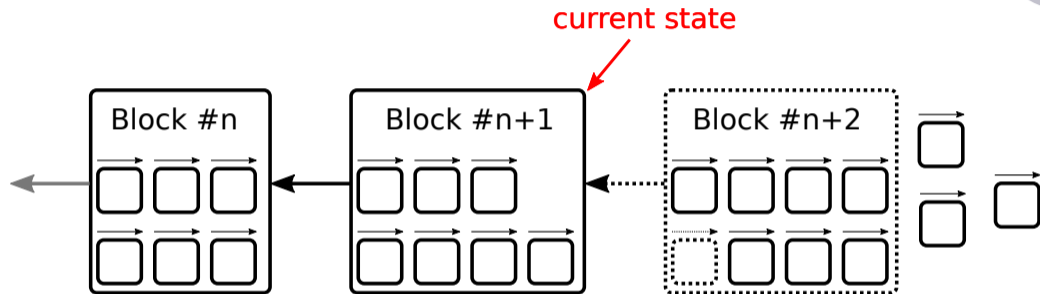
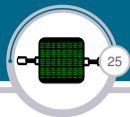
- ▶ Work on the block is going on
- ▶ New transactions arrive

Creating PoW protected blocks

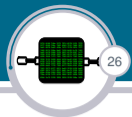


- ▶ The block is completed
 - ▶ PoW created successfully
- ▶ Work on the next block starts immediately

Creating PoW protected blocks

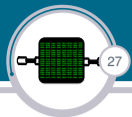


- ▶ The block is completed
 - ▶ PoW created successfully
- ▶ Work on the next block starts immediately



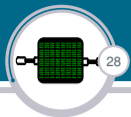
The function / challenge used for the PoW has to meet certain requirements:

- ▶ Be hard! Solvable only by brute force
- ▶ Validation needs to be fast and easy
- ▶ Dependant on the exact block it is produced for
 - ▶ To prevent pre-compute attacks
- ▶ Variable difficulty

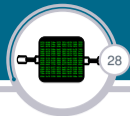


For mining to be worthwhile, a reward is needed:

- ▶ A transaction from nowhere to the miner
 - ▶ Transaction fees
 - ▶ Coinbase / new money
- ▶ Included in the block
- ▶ Makes blocks individual
 - ▶ Ensures distribution of success amongst all miners



- ▶ Which branch should a miner work on?
- ▶ For the reward to be spendable, it needs to be part of the state
- ▶ The state is the "longest" branch
- ▶ So it only makes sense to work on the "longest" branch
- ▶ Works without any coordination!



Systems for representing ownership

Blockchain (without PoW)

Blockchain

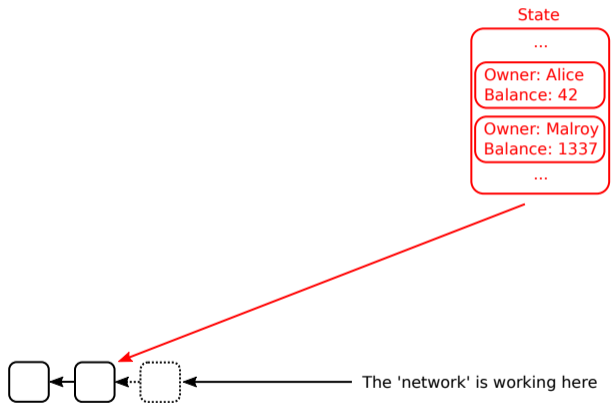
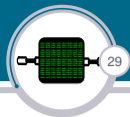
PoW and mining

Solving double-spend

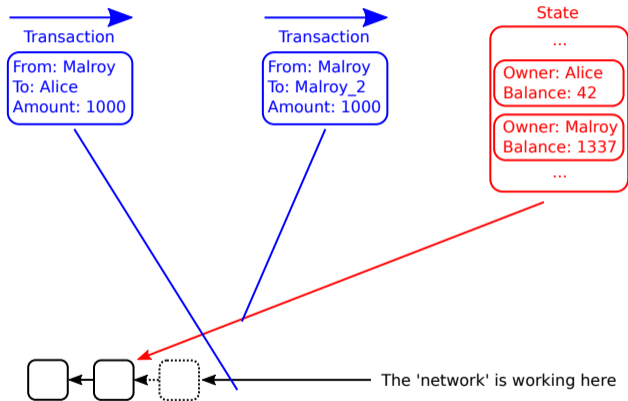
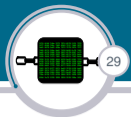
Bitcoin

Bonus Slides

Double-spend

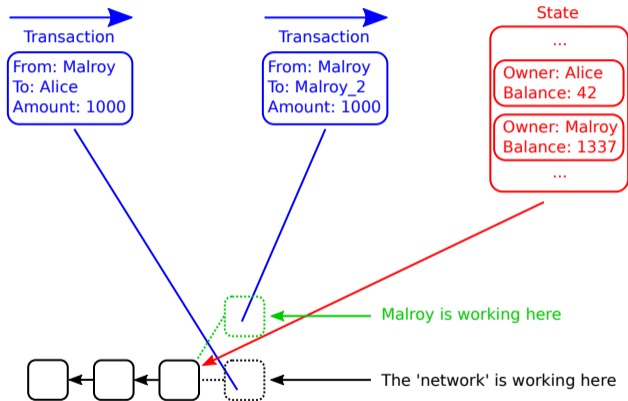
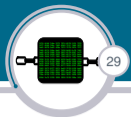


Double-spend



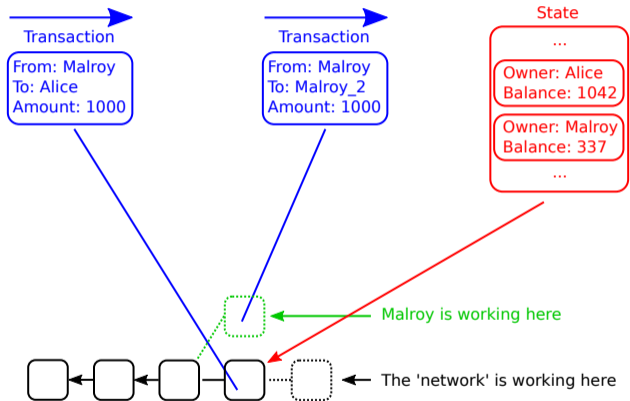
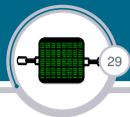
- ▶ Malroy creates two transactions
- ▶ Only the legitimate one gets published

Double-spend



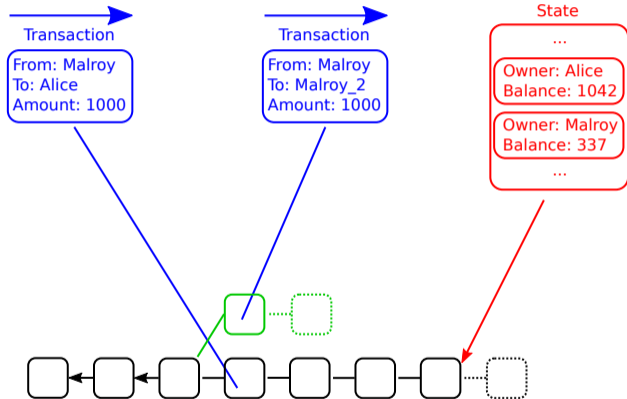
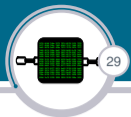
- ▶ Work on the next block starts
- ▶ Malroy is the only one working on a different block

Double-spend



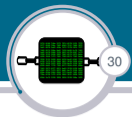
- ▶ The network has more power than Malroy
- ▶ Thus they create new blocks faster

Double-spend



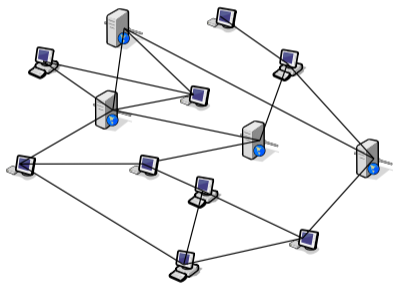
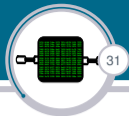
- ▶ Malroy's chain never becomes the longest

50% attack



- ▶ Double-spend is possible
- ▶ >50% of network's power is needed
- ▶ With that, Malroy could produce new blocks faster than the rest of the network

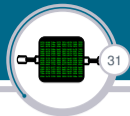
p2p networking attack setup



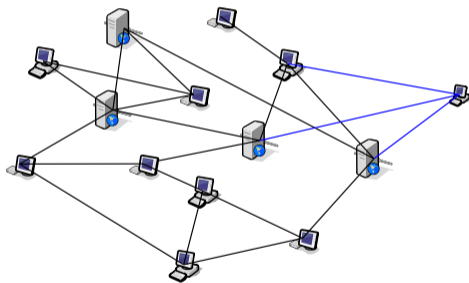
Most Blockchain currencies use a p2p network

- ▶ to distribute the transactions
- ▶ to distribute the blocks

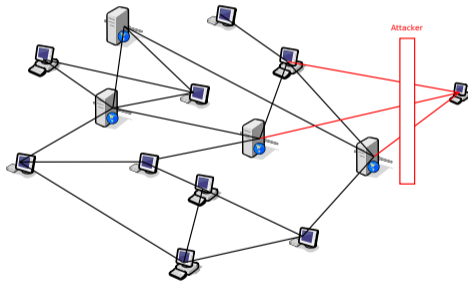
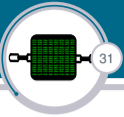
p2p networking attack setup



Malroy wants to control the victim's (Bob's) view of the network by...



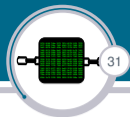
p2p networking attack setup



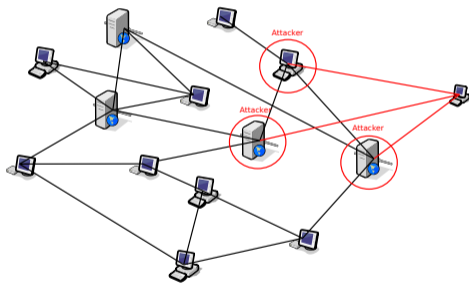
...Controlling Bob's network connection. That can happen sometimes

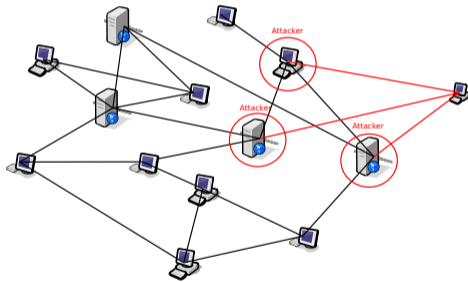
- ▶ network provider
- ▶ public wifi

p2p networking attack setup



...Controlling the peers Bob is connected to.



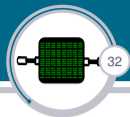


This allows Malroy to:

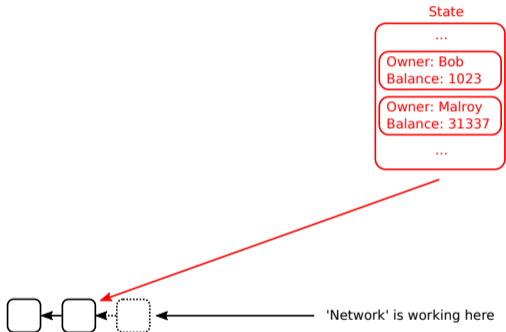
- ▶ Hide transactions from Bob
- ▶ Hide blocks from Bob
- ▶ Present extra transactions and / or blocks to Bob

In short, Malroy can maintain a fork of the Blockchain just for Bob. This allows Malroy to stage a different kind of double-spend attack:

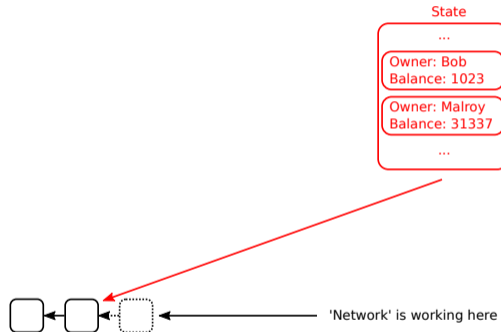
Network attack



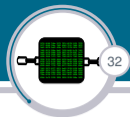
Public network view:



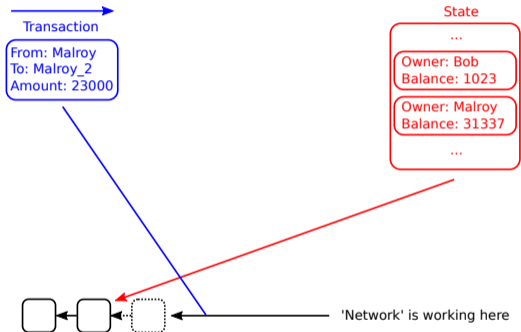
Bob's view:



Network attack



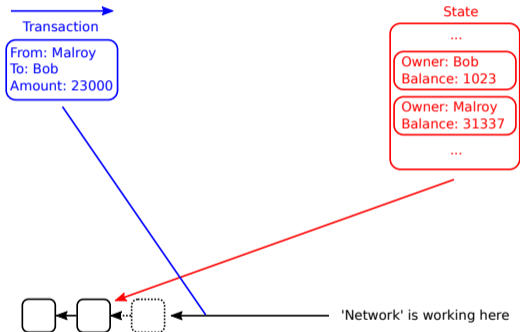
Public network view:



Malroy:

- ▶ Creates a transaction to Malroy_2
- ▶ Puts it onto the network

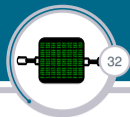
Bob's view:



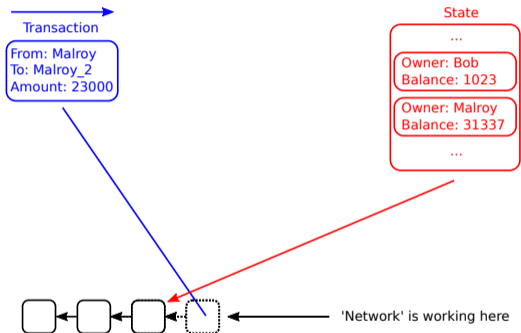
Malroy:

- ▶ Creates a transaction to Bob
- ▶ Sends it to Bob only

Network attack

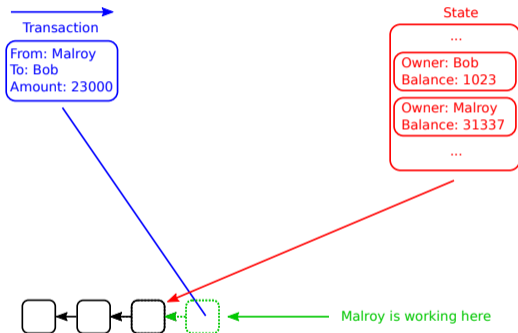


Public network view:



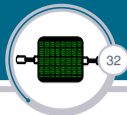
► Work on the next block starts

Bob's view:

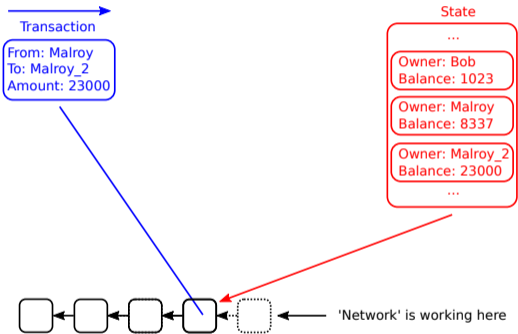


► Malroy starts to work on the new block

Network attack

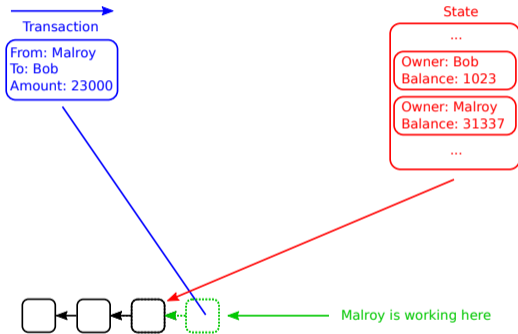


Public network view:



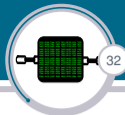
► The block gets created

Bob's view:

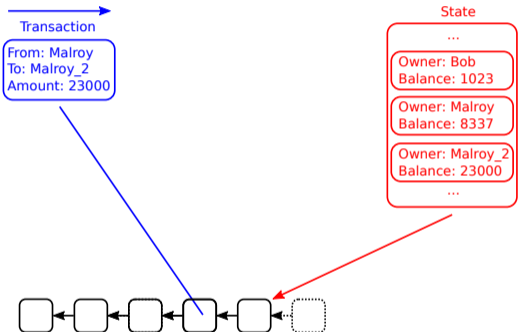


► Malroy is the only one working on this branch

Network attack

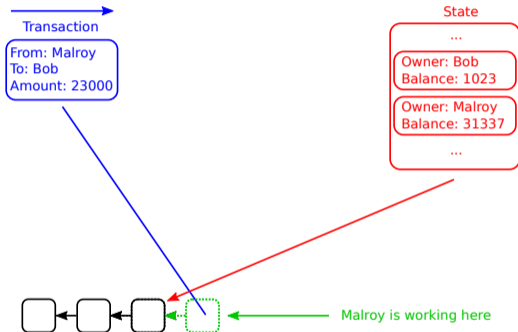


Public network view:



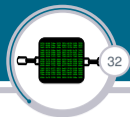
- New blocks get added to the chain

Bob's view:

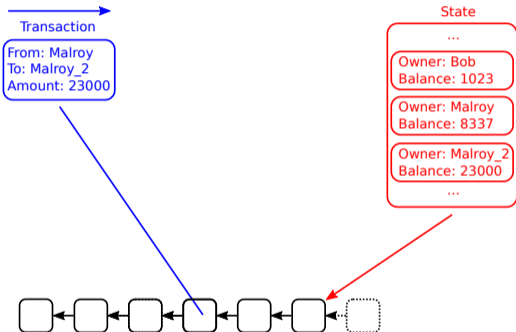


- Malroy is the only one working on this branch

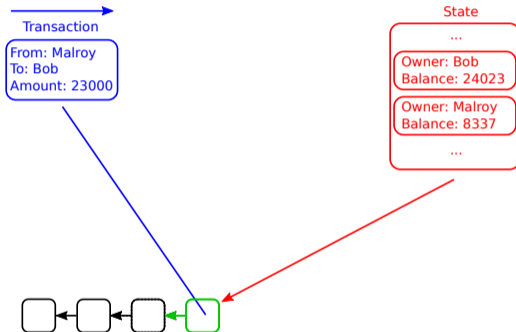
Network attack



Public network view:



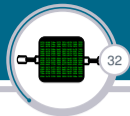
Bob's view:



- ▶ Bob can't see the other, longer branch
- ▶ Thus in his view, the transaction becomes part of the state

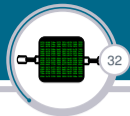
He has the money, hands over the car

Network attack

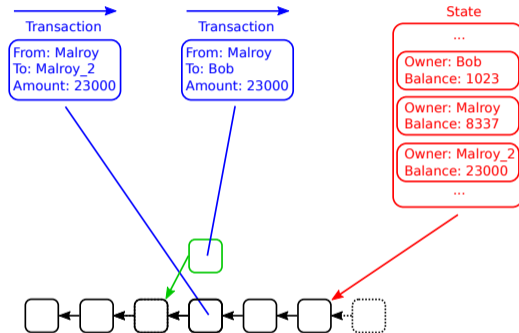


- ▶ The attack stops
- ▶ Bob synchronizes with the rest of the network
- ▶ The two chains merge

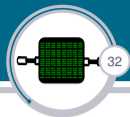
Network attack



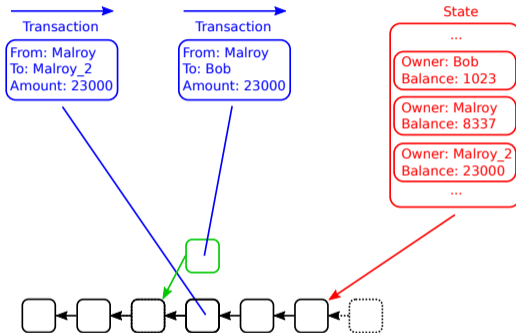
- ▶ The attack stops
- ▶ Bob synchronizes with the rest of the network
- ▶ The two chains merge



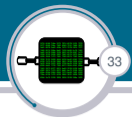
Network attack



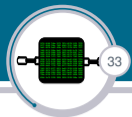
- ▶ The transaction to Bob is not part of the state
- ▶ Bob does not have the money



What happened



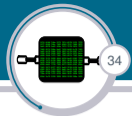
- ▶ The attack still worked
- ▶ It's now costly though
- ▶ Malroy's computer can either:
 - ▶ Perform the attack
 - ▶ Mine on the network
- ▶ For the time of the attack, Malroy has to decide
- ▶ This makes using the computer for the attack expensive



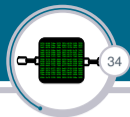
- ▶ The attack still worked
- ▶ It's now costly though
- ▶ Malroy's computer can either:
 - ▶ Perform the attack
 - ▶ Mine on the network
- ▶ For the time of the attack, Malroy has to decide
- ▶ This makes using the computer for the attack expensive

Just how expensive exactly?

Cost of the attack



- ▶ Malroy's compute power: $20\% = \frac{1}{5}$ of the network
- ▶ Network average block time: 10min
- ▶ Block reward: 1000 Euro

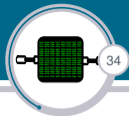


- ▶ Malroy's compute power: $20\% = \frac{1}{5}$ of the network
- ▶ Network average block time: 10min
- ▶ Block reward: 1000 Euro

Malroy needs to create one block:

Time Malroy needs to to create one block

$$100\% \text{ Power} \Leftrightarrow 10 \text{ min} \implies 20\% \text{ Power} \Leftrightarrow 50 \text{ min}$$



- ▶ Malroy's compute power: $20\% = \frac{1}{5}$ of the network
- ▶ Network average block time: 10min
- ▶ Block reward: 1000 Euro

Malroy needs to create one block:

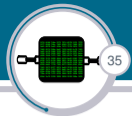
Time Malroy needs to to create one block

$100\%Power \Leftrightarrow 10min \implies 20\%Power \Leftrightarrow 50min$

What if Malroy was mining for the same 50 minutes instead?

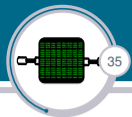
Reward in 50min mining

$100\%Power \Leftrightarrow 5Blocks \implies 20\%Power \Leftrightarrow 1Block \implies 1000Euro$



Malroy can chose:

- ▶ Do the attack
 - ▶ Steal a 23000 Euro car
- ▶ Not do the attack
 - ▶ Earn 1000 euro mining

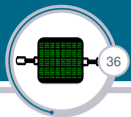


Malroy can chose:

- ▶ Do the attack
 - ▶ Steal a 23000 Euro car
- ▶ Not do the attack
 - ▶ Earn 1000 euro mining

Attacker's power

The power of the attacker does not matter. An attacker with more power needs less time, but he looses more money per time.

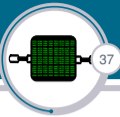


By waiting another block, Bob makes the attack twice as expensive.
So the larger a transaction we protect, the longer we have to wait. You can easily calculate for how long you have to wait to be secure:

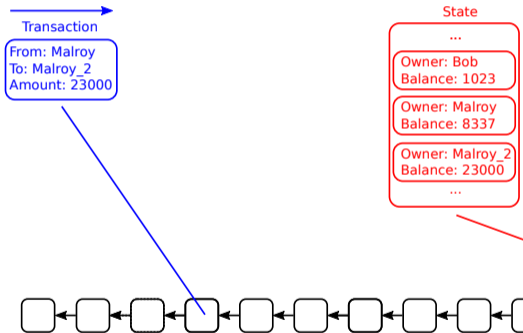
How long to wait?

$$\lceil \textit{amount} \div \textit{Blockreward} \rceil + 1$$

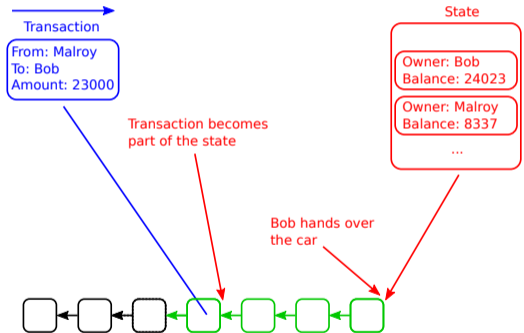
Bob's protection



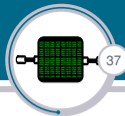
Public network view:



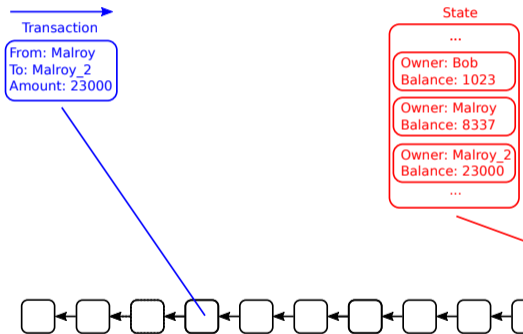
Bob's view:



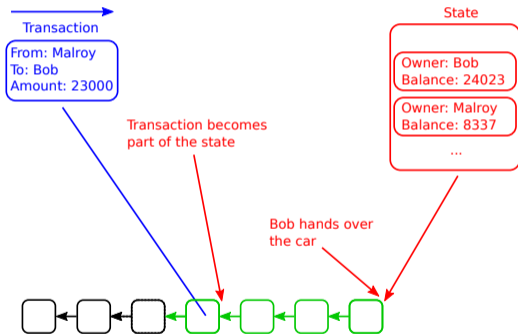
Bob's protection



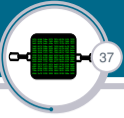
Public network view:



Bob's view:

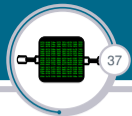


Bob will have to wait for 24 Blocks before handing over the car.



Section 4

Bitcoin



Systems for representing ownership

Blockchain (without PoW)

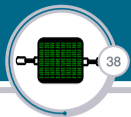
Blockchain

Bitcoin

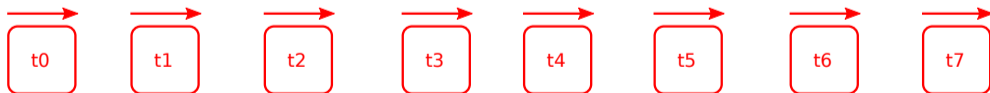
Blocks

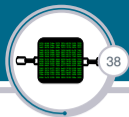
Light clients

Bonus Slides



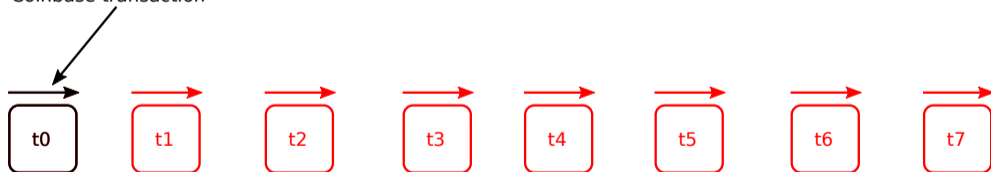
- ▶ Contains transactions
- ▶ Ordered
 - ▶ Dependant transactions are in the correct order

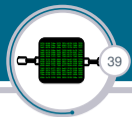




- ▶ Contains transactions
- ▶ Ordered
 - ▶ Dependant transactions are in the correct order
- ▶ The first transactions is Coinbase

Coinbase transaction

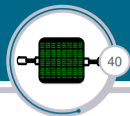




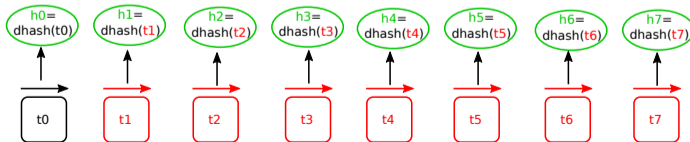
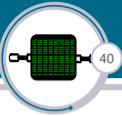
Bitcoin uses a Merkle tree to secure the transactions. Root of the Merkle tree is part of the Block header.

- ▶ Binary tree
- ▶ A node is the hash of the two child nodes
- ▶ Bitcoin uses sha256 double hashes
 - ▶ aka. dhash, double-sha256
 - ▶ $sha256 sha256(\dots)$

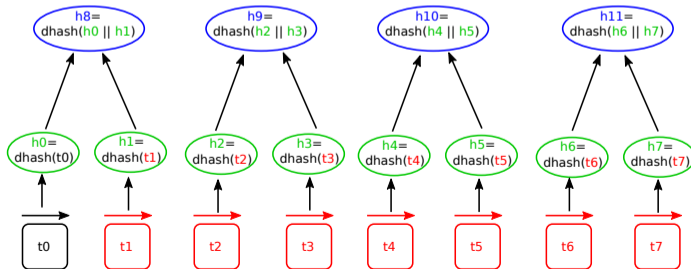
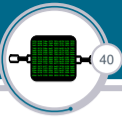
Merkle tree visualization



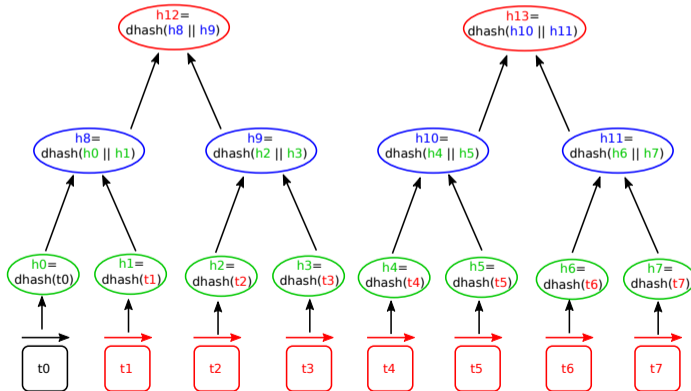
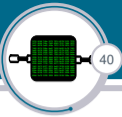
Merkle tree visualization



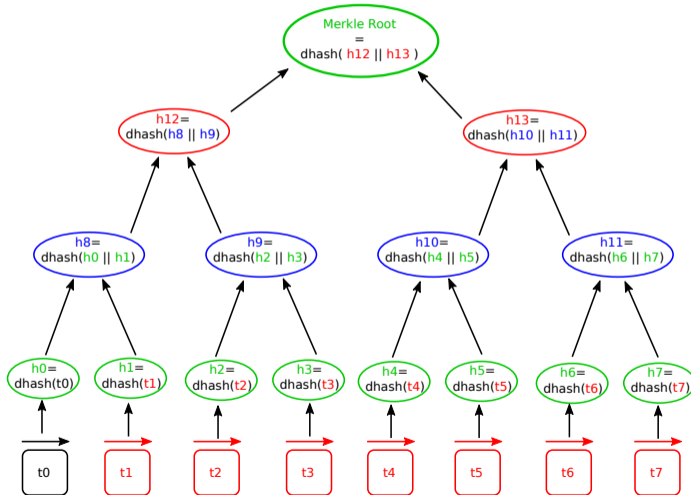
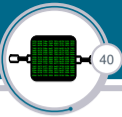
Merkle tree visualization



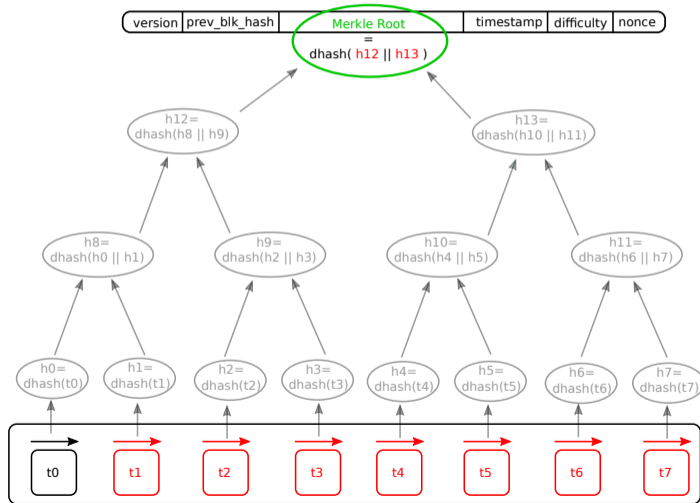
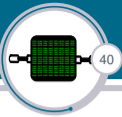
Merkle tree visualization

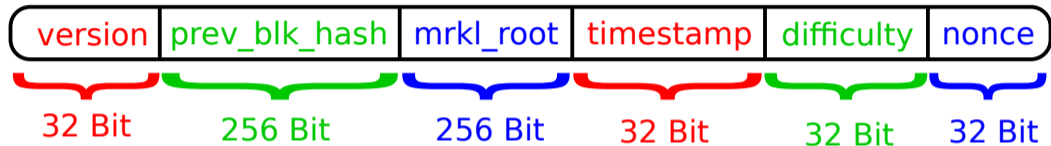
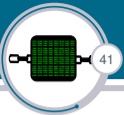


Merkle tree visualization



Merkle tree visualization





version Block version. Currently at 4

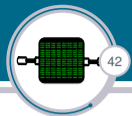
prev_blk_hash Hash of the previous block header. Reference to the previous block.

mrkl_root Reference to the transactions. Root node of the Merkle tree.

timestamp Standard UNIX timestamp. Complicated rules here.

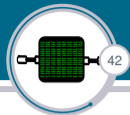
difficulty Difficulty of the block. Network difficulty is recalculated every 2016 blocks.

nonce Nonce used to manipulate the block hash. Note: too short.



PoW:

- ▶ Bitcoin uses the Block Hash for PoW
 - ▶ $dhash(block_header)$
- ▶ PoW: First n Bits of the Block Hash need to be 0



PoW:

- ▶ Bitcoin uses the Block Hash for PoW
 - ▶ $dhash(block_header)$
- ▶ PoW: First n Bits of the Block Hash need to be 0
- ▶ Mining by incrementing the `nonce` field
 - ▶ Too short (32bit nonce for 256bit Hash)
- ▶ Secondary nonce
 - ▶ tx input / input script of Coinbase transaction
 - ▶ Change in transaction changes Merkle root



PoW:

- ▶ Bitcoin uses the Block Hash for PoW
 - ▶ $dhash(block_header)$
- ▶ PoW: First n Bits of the Block Hash need to be 0
- ▶ Mining by incrementing the `nonce` field
 - ▶ Too short (32bit nonce for 256bit Hash)
- ▶ Secondary nonce
 - ▶ tx input / input script of Coinbase transaction
 - ▶ Change in transaction changes Merkle root

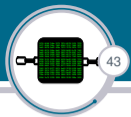
Difficulty:

- ▶ Difficulty is adjusted every 2016 Blocks (14 days)
- ▶ Network speed target: one Block ever 10 minutes



The miner of a new block gets a reward. The reward is the sum of:

- ▶ Fixed reward/ newly generated money
 - ▶ Started at 50 Btc per Block
 - ▶ Halves every 210000 Blocks (ca. 4 years)
- ▶ The transaction fees of all transactions in the block

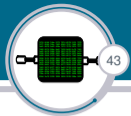


The miner of a new block gets a reward. The reward is the sum of:

- ▶ Fixed reward/ newly generated money
 - ▶ Started at 50 Btc per Block
 - ▶ Halves every 210000 Blocks (ca. 4 years)
- ▶ The transaction fees of all transactions in the block

To do this, the miner creates an additional transaction:

- ▶ So called Coinbase transaction
- ▶ First (left most) transaction in the Block



Systems for representing ownership

Blockchain (without PoW)

Blockchain

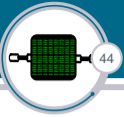
Bitcoin

Blocks

Light clients

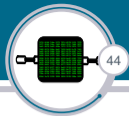
Bonus Slides

Types of Bitcoin clients



- ▶ Full node
 - ▶ Stores the entire Blockchain
 - ▶ Seeds the Blockchain to the network
 - ▶ Validates every block
 - ▶ Validates every transaction

Types of Bitcoin clients



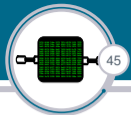
- ▶ Full node
 - ▶ Stores the entire Blockchain
 - ▶ Seeds the Blockchain to the network
 - ▶ Validates every block
 - ▶ Validates every transaction
- ▶ Pruning client
 - ▶ Validates all blocks
 - ▶ Validates all transactions
 - ▶ Only stores parts of the Blockchain
 - ▶ See the bonus slides

Types of Bitcoin clients

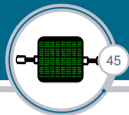


- ▶ Full node
 - ▶ Stores the entire Blockchain
 - ▶ Seeds the Blockchain to the network
 - ▶ Validates every block
 - ▶ Validates every transaction
- ▶ Pruning client
 - ▶ Validates all blocks
 - ▶ Validates all transactions
 - ▶ Only stores parts of the Blockchain
 - ▶ See the bonus slides
- ▶ Light client / SPV client
 - ▶ See below

Chain of Block headers



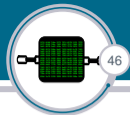
- ▶ Download just the headers
- ▶ PoW can be checked from just the header
- ▶ Previous block can be checked from just the header
- ▶ Creating a header is as hard as creating a block!



- ▶ Download just the headers
- ▶ PoW can be checked from just the header
- ▶ Previous block can be checked from just the header
- ▶ Creating a header is as hard as creating a block!

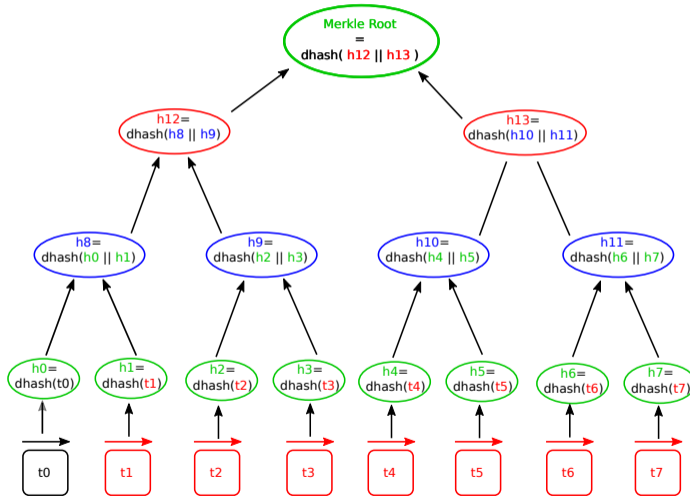
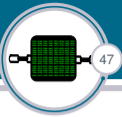
Advantage:

- ▶ 450000 blocks
- ▶ Blockchain: >95 GB (>100GB with indexing)
- ▶ Block headers: <100 MB

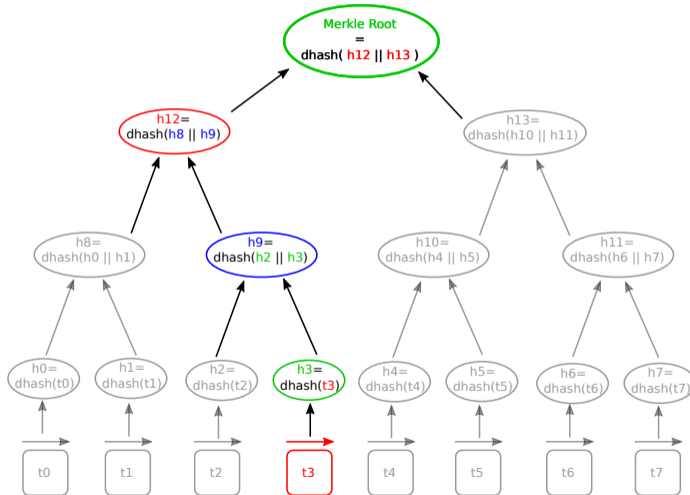
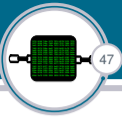


- ▶ We have the Merkle root
 - ▶ We can get it securely from the Block header
- ▶ We do not have the elements it consists of
- ▶ We are interested in a single element
 - ▶ A single transaction

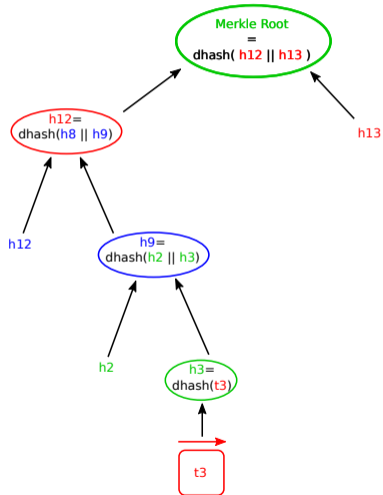
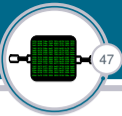
Merkle branch visualization



Merkle branch visualization



Merkle branch visualization



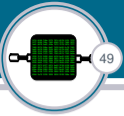


Download a Merkle branch instead of all transactions:

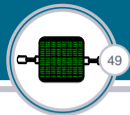
- ▶ Less elements
- ▶ Smaller (hashes instead of transactions)
- ▶ The advantages get larger the bigger the tree

Example: Block with 1600 transactions:

- ▶ 1600 transactions
- ▶ 1 transaction + 11 hashes

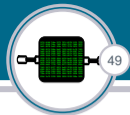


- ▶ Simple Payment Verification
- ▶ Proof that a transaction is part of the chain
- ▶ Avoid downloading the entire chain

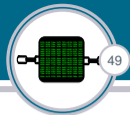


- ▶ Simple Payment Verification
- ▶ Proof that a transaction is part of the chain
- ▶ Avoid downloading the entire chain

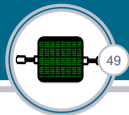
1. Retrieve all block headers



- ▶ Simple Payment Verification
 - ▶ Proof that a transaction is part of the chain
 - ▶ Avoid downloading the entire chain
1. Retrieve all block headers
 2. Rebuild longest branch



- ▶ Simple Payment Verification
 - ▶ Proof that a transaction is part of the chain
 - ▶ Avoid downloading the entire chain
1. Retrieve all block headers
 2. Rebuild longest branch
 3. Retrieve a transaction



- ▶ Simple Payment Verification
 - ▶ Proof that a transaction is part of the chain
 - ▶ Avoid downloading the entire chain
1. Retrieve all block headers
 2. Rebuild longest branch
 3. Retrieve a transaction
 4. Retrieve the Merkle branch for the transaction



- ▶ Simple Payment Verification
 - ▶ Proof that a transaction is part of the chain
 - ▶ Avoid downloading the entire chain
1. Retrieve all block headers
 2. Rebuild longest branch
 3. Retrieve a transaction
 4. Retrieve the Merkle branch for the transaction
 5. Match the Merkle branch to the chain of block headers

Questions?

Ask them now or contact me:

33c3 DECT / GSM

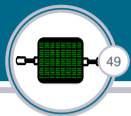
- ▶ 5346 (LEGO)
- ▶ 7869 (PUNX)

email

- ▶ niklaus@mykolab.ch

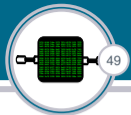
XMPP

- ▶ vimja@xmpp.honet.ch



Section 5

Bonus Slides



Systems for representing ownership

Blockchain (without PoW)

Blockchain

Bitcoin

Bonus Slides

Branches in the chain

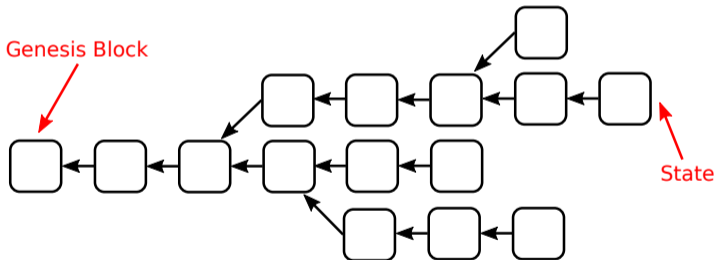
Double-spend

Mining

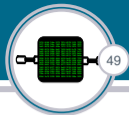
Status switch between branches



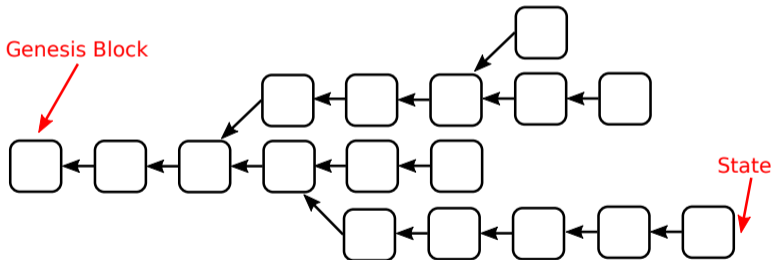
The longest chain of blocks represents the current state. If we display the chain as a tree, then that is the longest branch. The root of that tree is called the Genesis Block.



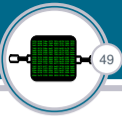
Status switch between branches



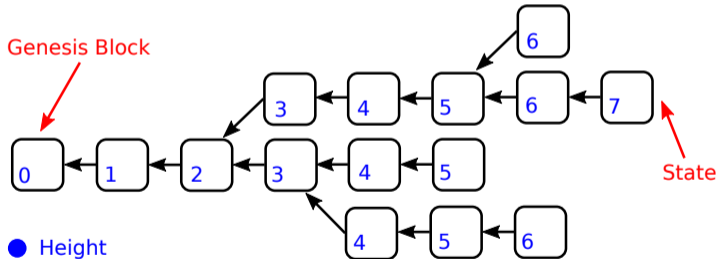
The longest chain of blocks represents the current state. If we display the chain as a tree, then that is the longest branch. The root of that tree is called the Genesis Block.



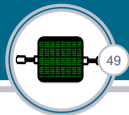
Block height and depth



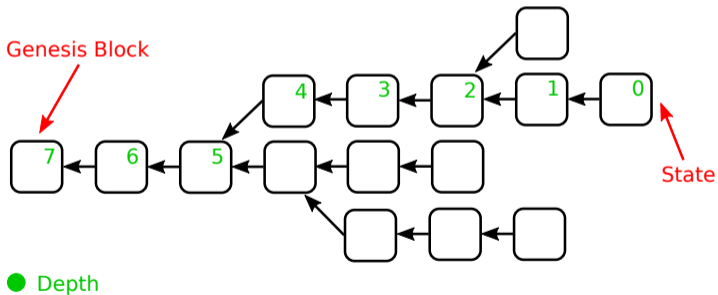
Height identifier



Block height and depth



Depth expression of security

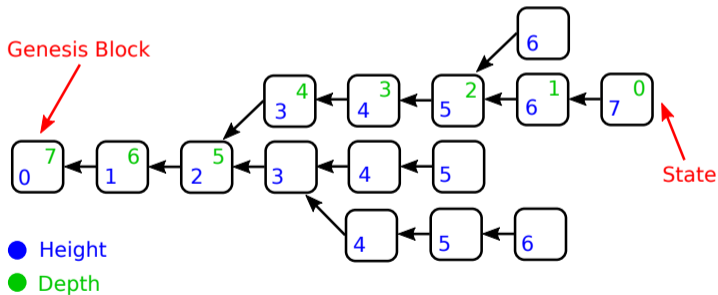


Block height and depth

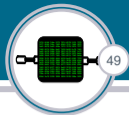


Height identifier

Depth expression of security

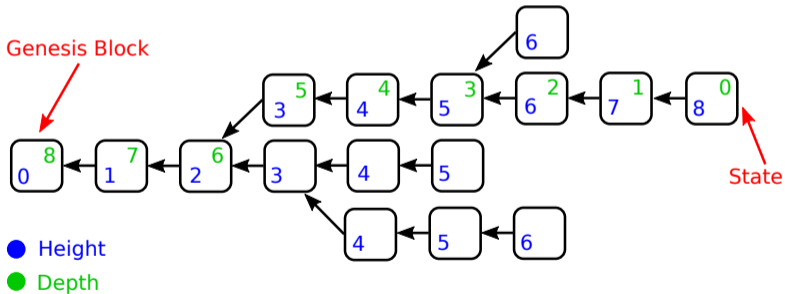


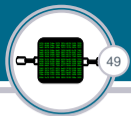
Block height and depth



Height identifier

Depth expression of security





Systems for representing ownership

Blockchain (without PoW)

Blockchain

Bitcoin

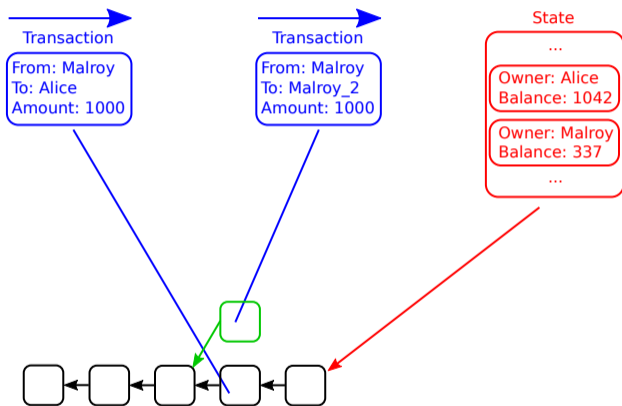
Bonus Slides

Branches in the chain

Double-spend

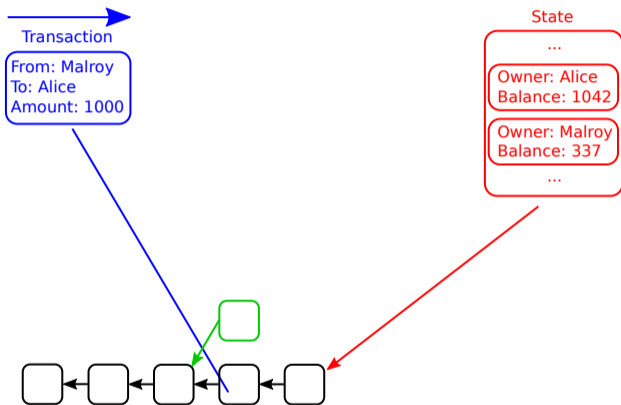
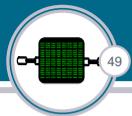
Mining

Conflicting transaction in double-spend attacks



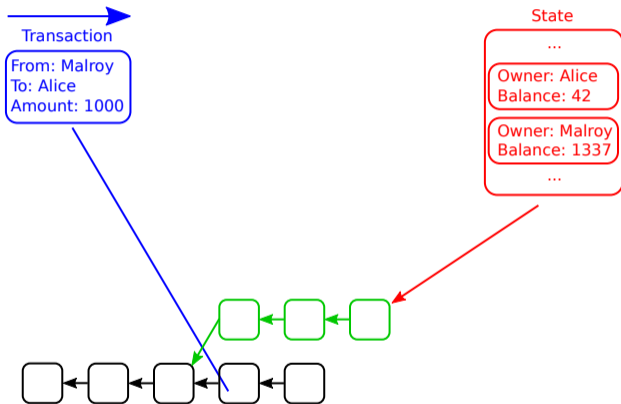
- What is the additional transaction for?

Conflicting transaction in double-spend attacks

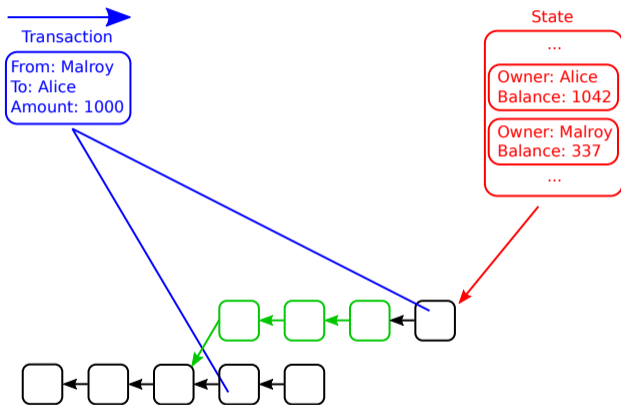
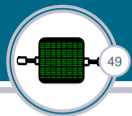


- ▶ Let's see what happens without it

Conflicting transaction in double-spend attacks

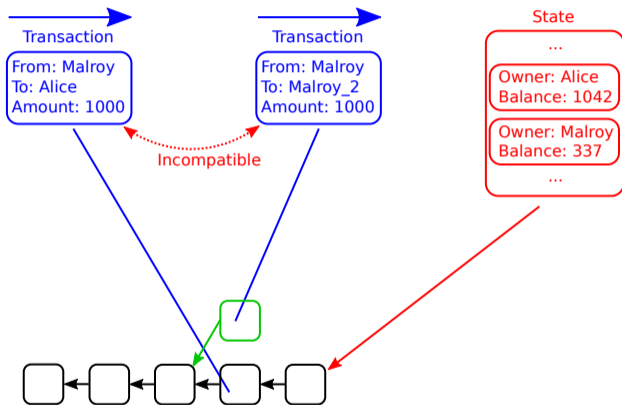
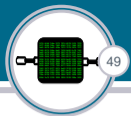


Conflicting transaction in double-spend attacks



- ▶ Someone could create a new block containing the transaction
- ▶ This would make a valid block
 - ▶ It is not in conflict with any previous block
- ▶ The transaction becomes part of the new state
- ▶ Malroy's money is gone

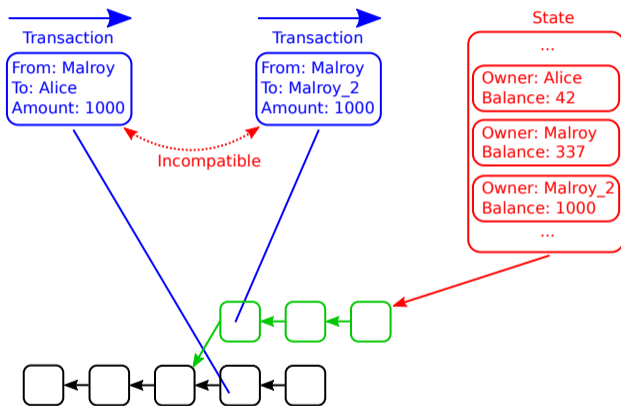
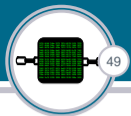
Conflicting transaction in double-spend attacks



So to prevent that:

- ▶ Malroy creates a conflicting transaction
- ▶ Puts it into the state

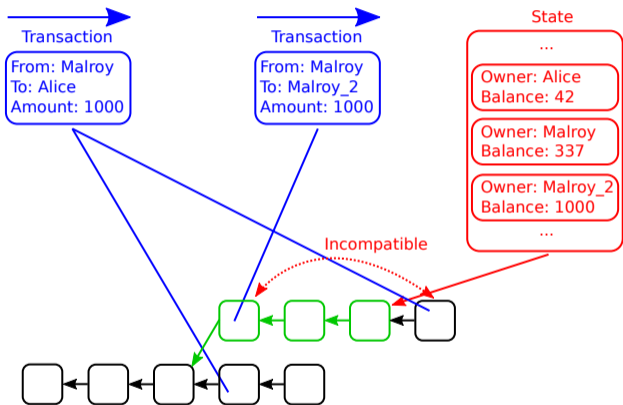
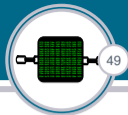
Conflicting transaction in double-spend attacks



So to prevent that:

- ▶ Malroy creates a conflicting transaction
- ▶ Puts it into the state

Conflicting transaction in double-spend attacks



- ▶ New blocks containing the old transaction would conflict
- ▶ They would be invalid
- ▶ And they could not become part of the state



Systems for representing ownership

Blockchain (without PoW)

Blockchain

Bitcoin

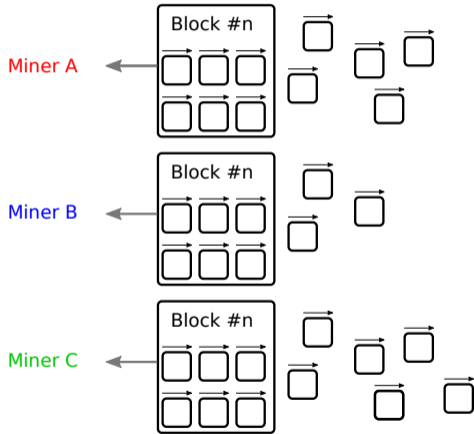
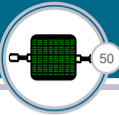
Bonus Slides

Branches in the chain

Double-spend

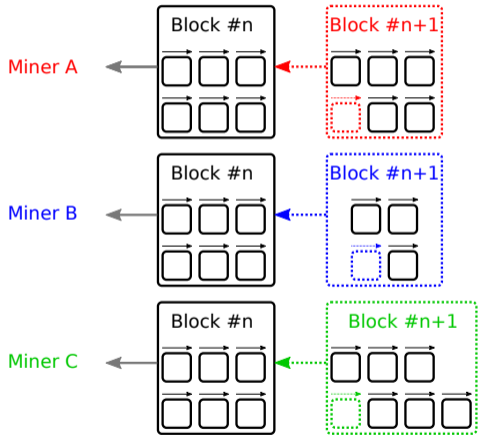
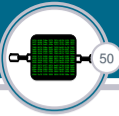
Mining

Example scenario with multiple miners



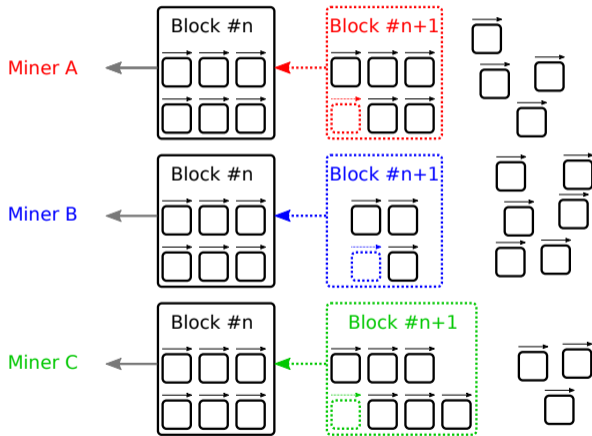
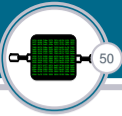
- ▶ Miners have different sets of transactions in their transaction pools
- ▶ Due to network delays

Example scenario with multiple miners



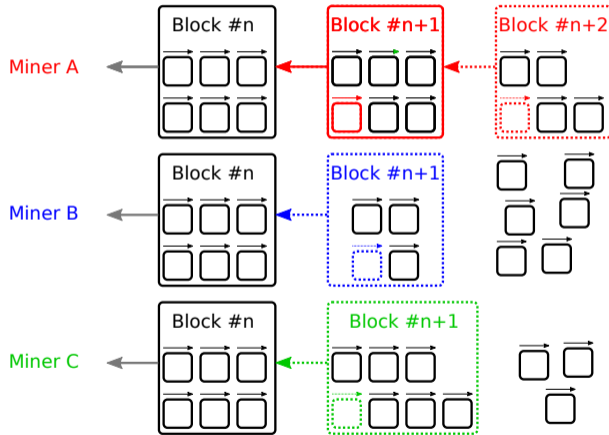
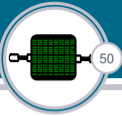
- ▶ Block #n gets completed
- ▶ All miners immediately start work on Block #n+1

Example scenario with multiple miners



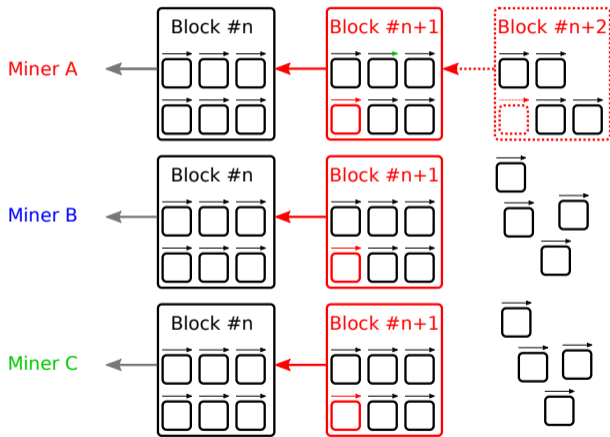
- ▶ New transactions keep arriving
- ▶ Miners add them to their transaction pools

Example scenario with multiple miners



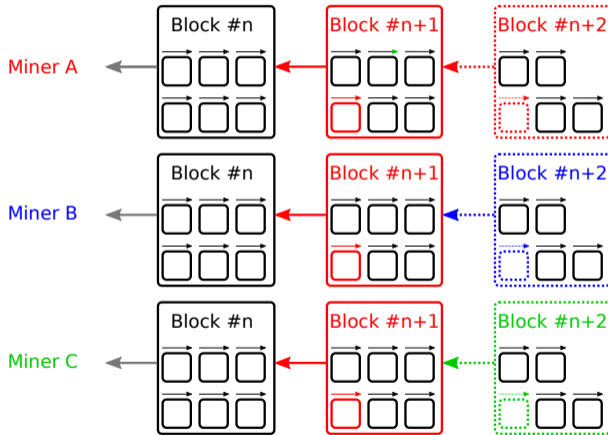
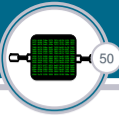
- ▶ Miner A finishes Block #n+1
- ▶ Miner A immediately starts work on #n+2

Example scenario with multiple miners



- ▶ Miner A broadcasts Block #n+1
- ▶ Miners B and C both receive the completed Block #n+1 from miner A
- ▶ They change their transaction pools accordingly

Example scenario with multiple miners



- ▶ Miners B and C immediately start work on Block #n+2