CSE446: Blockchain & Cryptocurrencies

Lecture – 15: Ethereum – 4

BRAC UNIVERSITY

Inspiring Excellence

Agenda

- Ethereum Consensus

Memory-bound consensus algorithm

- To counteract this issue of Bitcoin's CPU-bound PoW algorithm, memory-bound PoWs have been proposed
- A memory-bound PoW requires the algorithm to access the main memory several times
- This ultimately binds the performance of the algorithm within the limit of access latency and/or bandwidth as well as the size of memory
    - The higher the memory the faster the performance

Memory-bound consensus algorithm

- This restricts ASIC rigs based miners not to have manifold performance advantage over CPU/GPU-based mining rigs
- The reason is even though thousands of ASICs could be combined they would have a performance threshold based on the size/bw/latency of the memory
  - Remember that you can't install unlimited memory within a PC
- This approach also limits the profit margin of the miners who have a mammoth ASIC-based mining rig
- Another motivation of this approach is to de-monopolise the mining concentrations around some central mining nodes

Ethereum PoW consensus algorithm

- Ethash (DAGGER-HASHIMOTO)/DAGGER is the consensus algorithm designed for Ethereum PoW
- Ethash is a memory-bound PoW algorithm with the goal to be ASIC-resistant for a long period of time
- Dagger is one of the earliest proposed memory-bound PoW algorithms which utilises a Directed Acyclic Graph (DAG)
    - a directed acyclic graph (DAG) is a directed graph with no directed cycles
- However it was found be vulnerable
- Ethereum combined Dagger and Hashimoto algorithms to be more secure

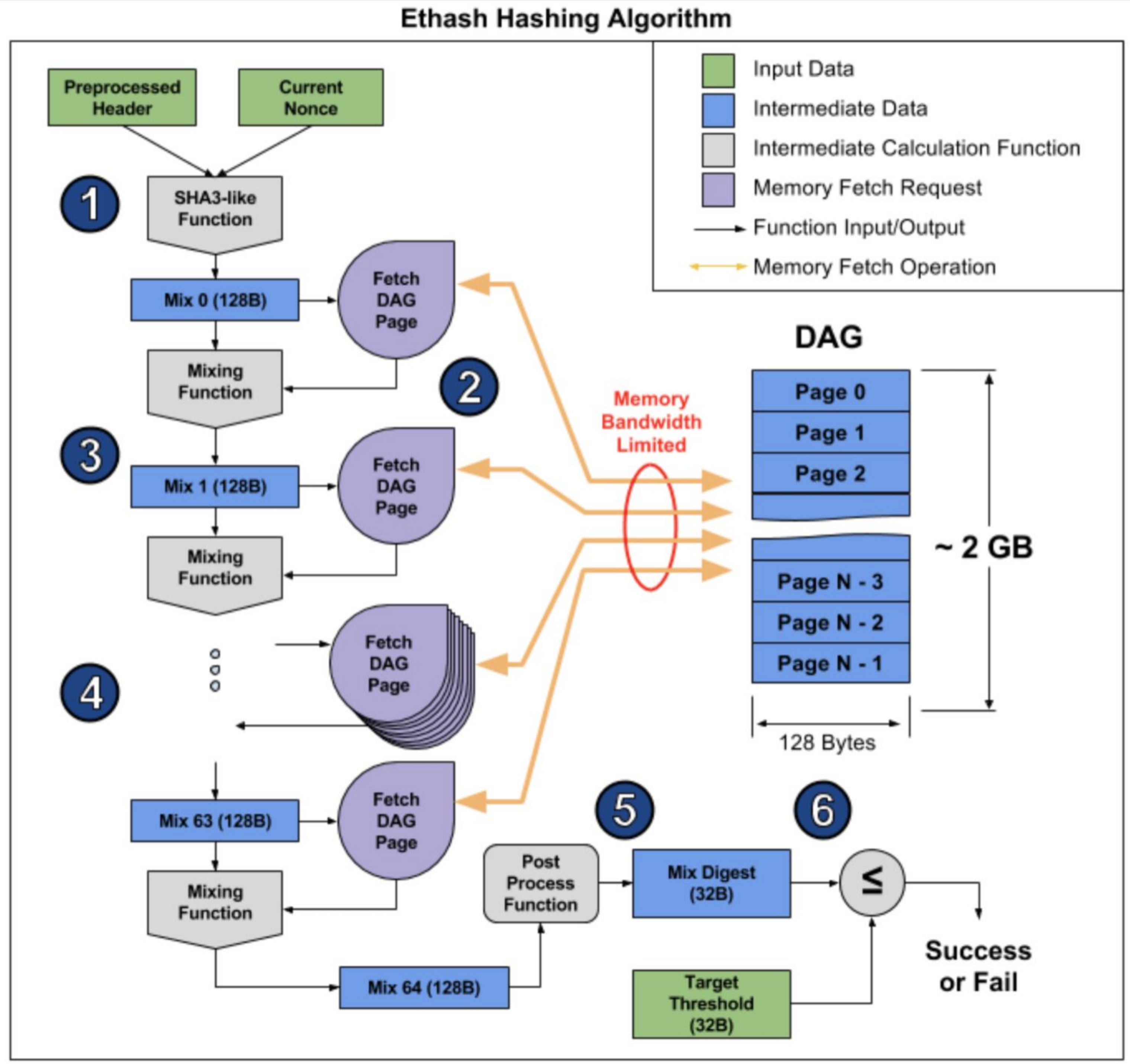
Ethash algorithm

- Ethash depends on a large pseudo-random dataset (the DAG), which is recomputed during each epoch
- Each epoch is determined by the time it takes to generate 30,000 blocks which is approximately five days
- During the DAG generation process, a seed is generated at first, which relies on the length of the chain
- The seed is then used to compute a 16 MB pseudo-random cache
- Then, each item of the DAG is generated by utilising a certain number of items from the pseudo-random cache

Ethash algorithm

- Then, the latest block header and the current candidate nonce are hashed using Keccak (SHA-3) hash function
- The resultant hash is mixed several times with data from the DAG, the mixHash data field
- The final hashed digest is compared to the difficulty target and accepted or discarded accordingly

Ethash algorithm



From Lecture Slides of Blockchains & Cryptocurrencies Course at Johns Hopkins University

Ethereum PoS algorithm

- Ethereum moved to a PoS consensus algorithm in 2022
- Like any PoS algorithm there are a number of validators
- To be a validator, one has to deposit 32 eth to an escrow contract
- On depositing their ETH, the user joins an activation queue that limits the rate of new validators joining the network
- Once activated, validators receive new blocks from peers on the Ethereum network
- The transactions delivered in the block are re-executed, and the block is verified
- The validator then sends a vote (called an attestation) in favour of that block across the network

https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

Ethereum PoS algorithm

- Under Ethash algorithm, the timing of blocks is determined by the mining difficulty
- In proof-of-stake, the block generation time is fixed
- Time in proof-of-stake Ethereum is divided into slots (12 seconds) and epochs (32 slots)
- One validator is randomly selected to be a block proposer in every slot
- This validator is responsible for creating a new block and sending it out to other nodes on the network

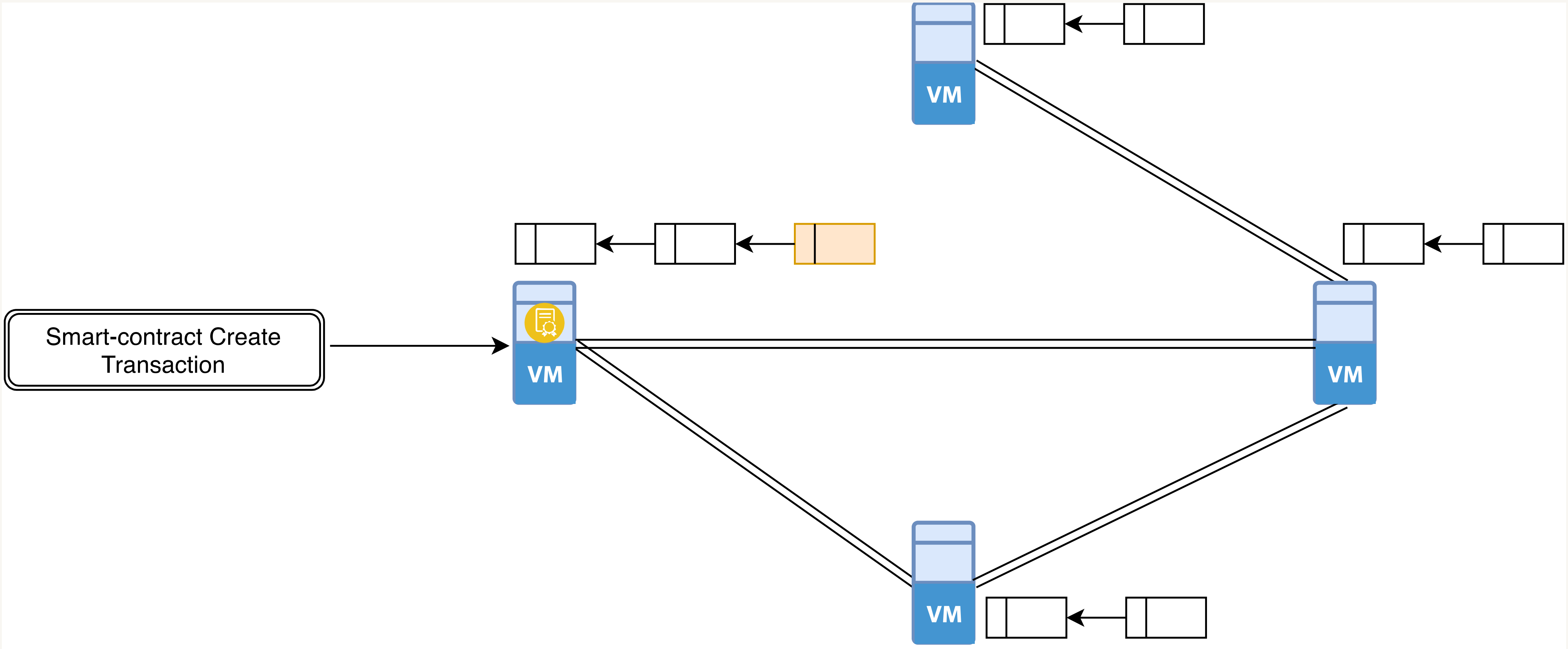
https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

Ethereum PoS algorithm

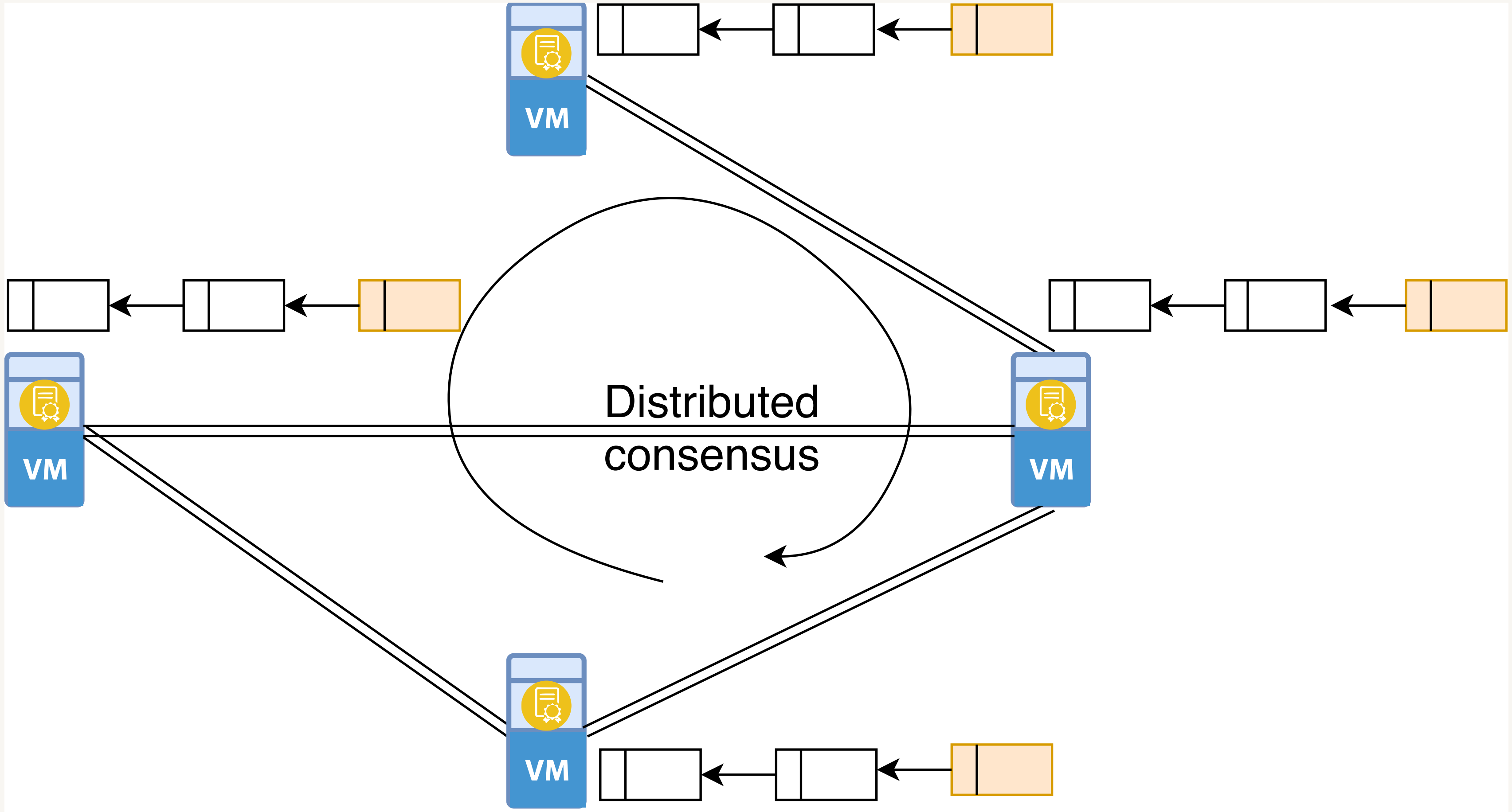
- Also in every slot, a committee of validators is randomly chosen, whose votes are used to determine the validity of the block being proposed
- If a validator is chosen to attest the next block, they are rewarded in ETH as a percentage of their stake
- Conversely, validators who do not perform their duties––if they are offline, for example––receive penalties, or slashes, in the form of small amounts of ETH subtracted from their stakes

https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

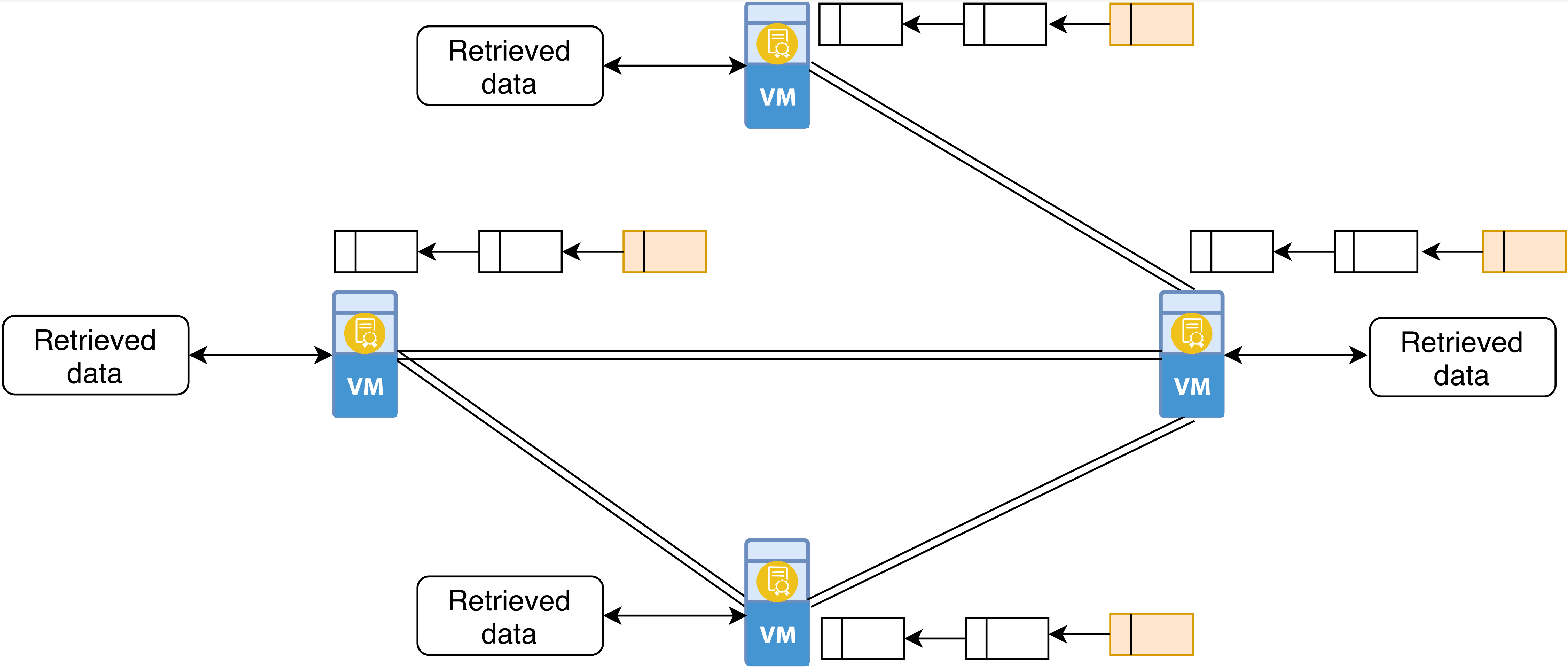
Ethereum illustration



Ethereum illustration



Ethereum illustration



Retrieved data

VM

Retrieved data

VM

VM

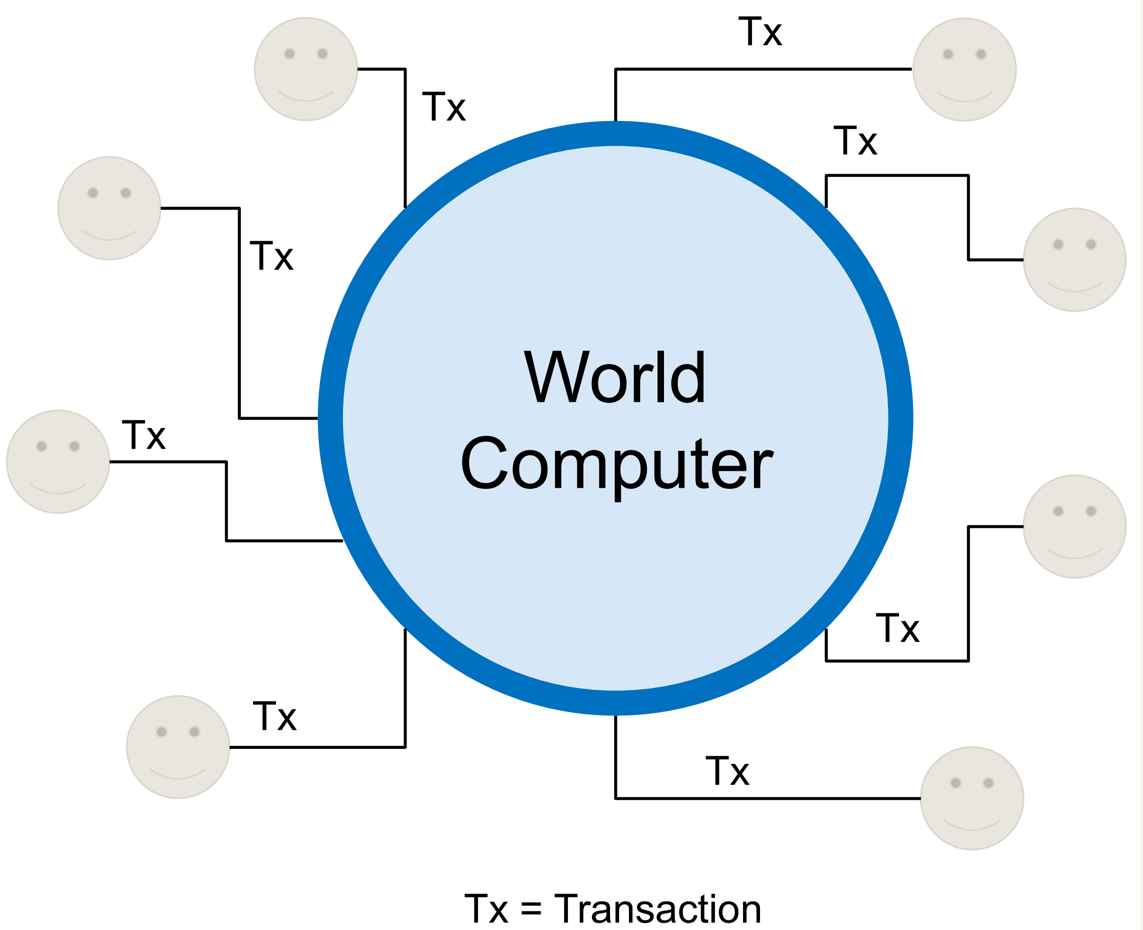
Retrieved data

Retrieved data

VM

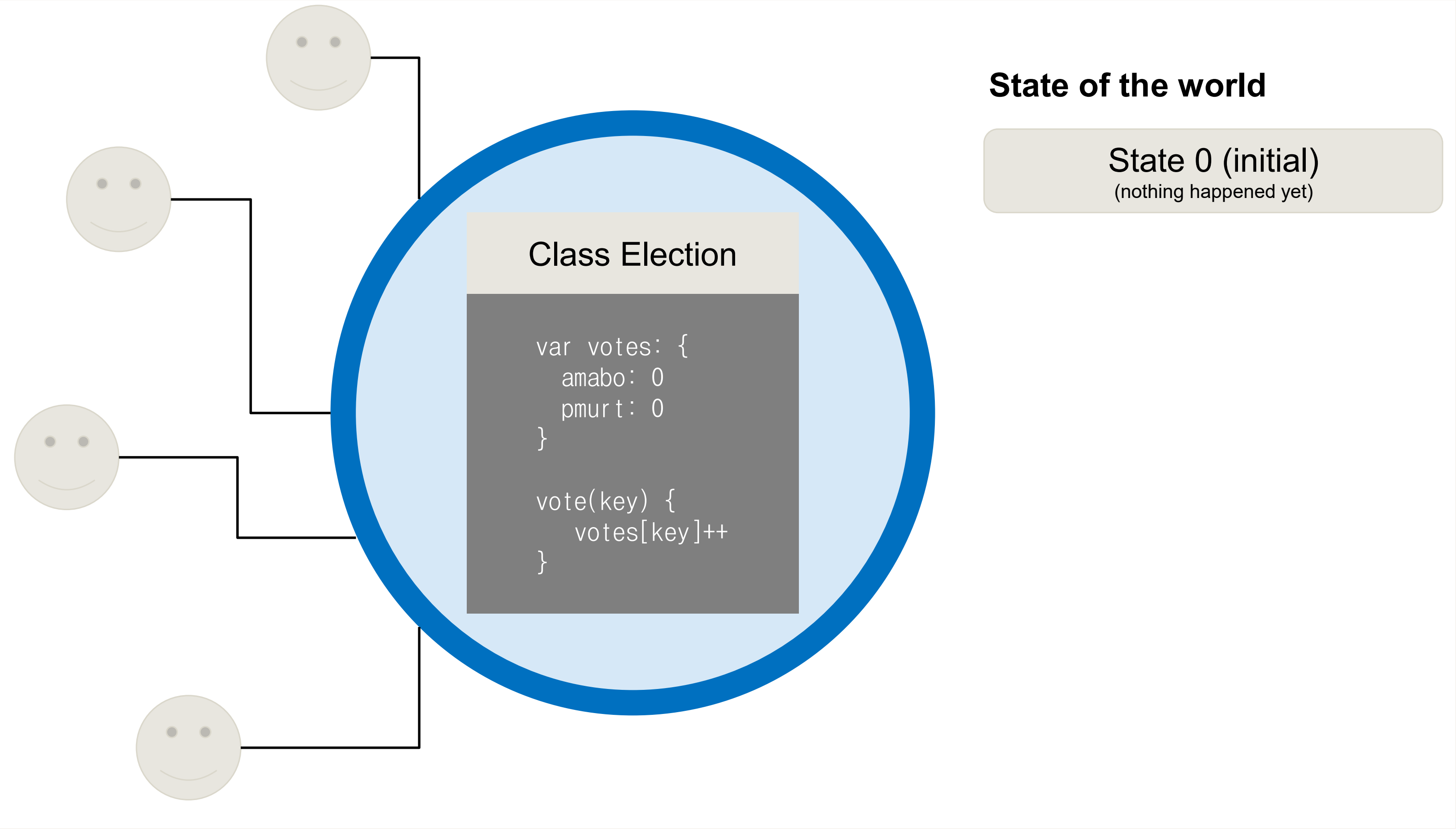
Ethereum visualisation

- All participants are using the "same" computer
- Users issue transactions to call programs on the computer
- Everyone shares the same resources and storage
- The computer has no explicit, single owner
- Using the computer's resources costs money (eth)



07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualisation: election example



Class Election

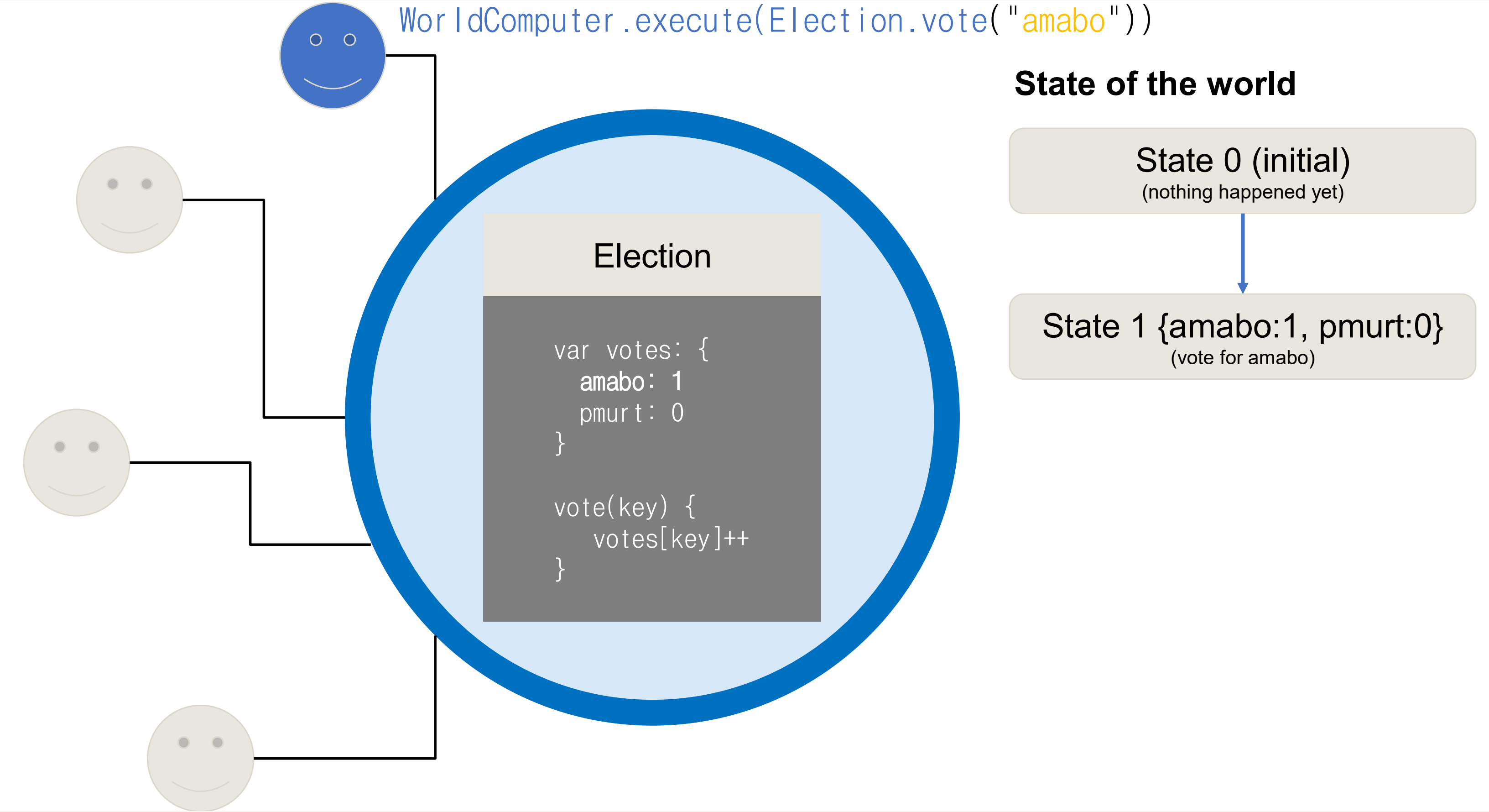
```
var votes: {
  amabo: 0
  pmurt: 0
}

vote(key) {
   votes[key]++
}
```

State of the world

State 0 (initial)
(nothing happened yet)

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



```
WorldComputer.execute(Election.vote("amabo"))
```

State of the world

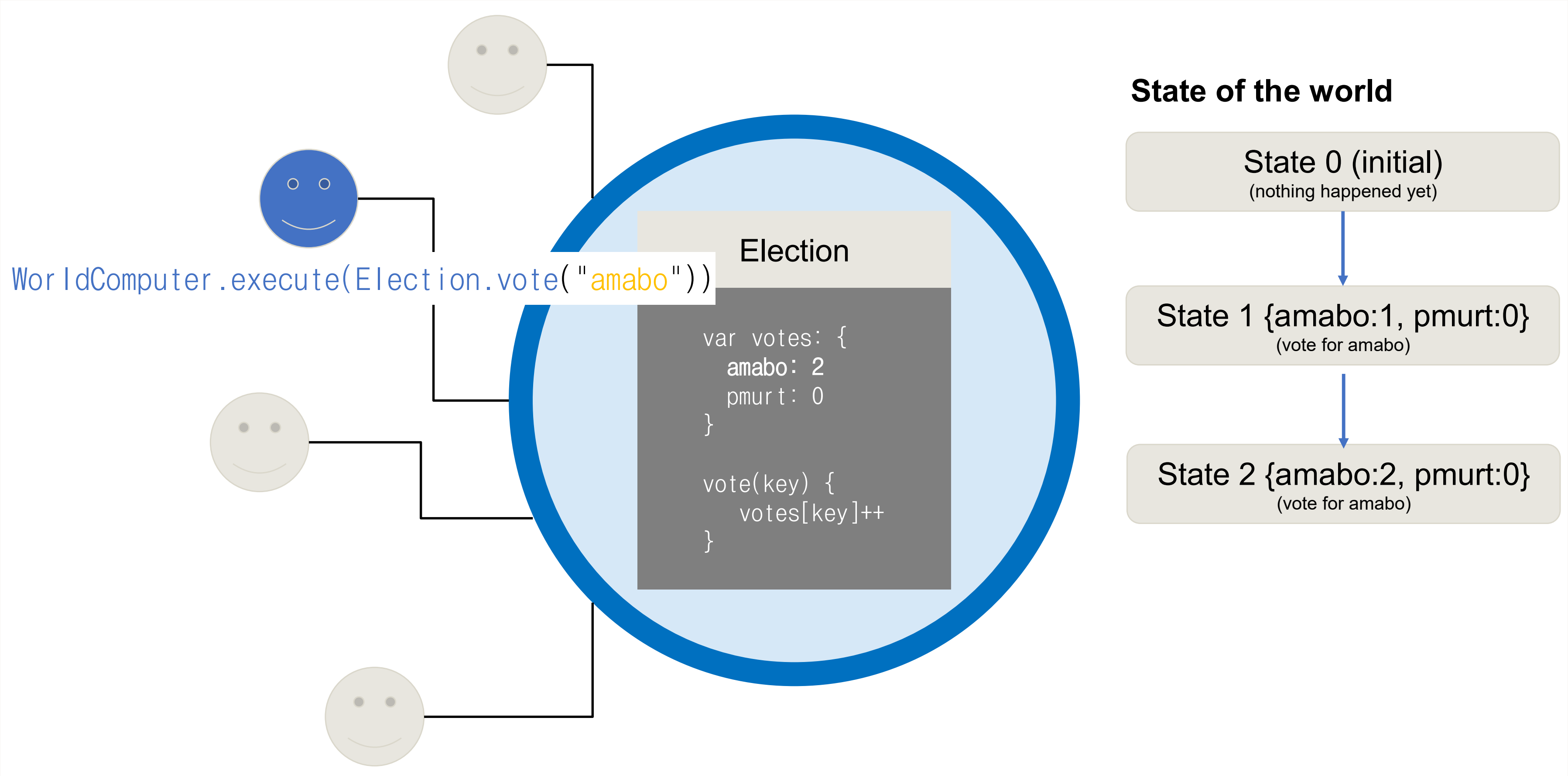
State 0 (initial)
(nothing happened yet)

State 1 {amabo:1, pmurt:0}
(vote for amabo)

```
var votes: {
  amabo: 1
  pmurt: 0
}

vote(key) {
   votes[key]++
}
```

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



```
WorldComputer.execute(Election.vote("amabo"))
```

```
var votes: {
  amabo: 2
  pmurt: 0
}

vote(key) {
    votes[key]++
}
```

State of the world

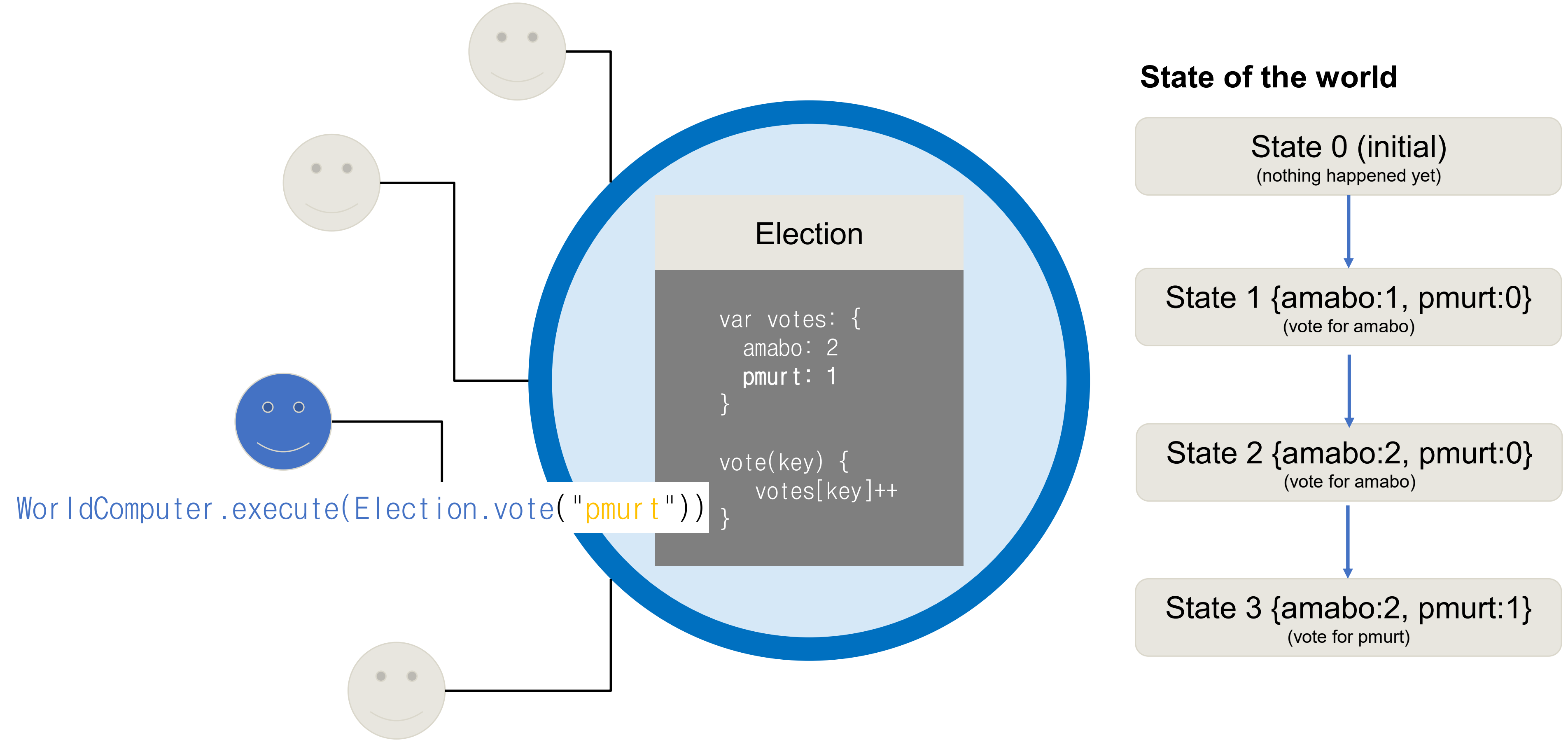
State 0 (initial)
(nothing happened yet)

State 1 {amabo:1, pmurt:0}
(vote for amabo)

State 2 {amabo:2, pmurt:0}
(vote for amabo)

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



Election

```
var votes: {
  amabo: 2
  pmurt: 1
}

vote(key) {
  votes[key]++
}
```

```
WorldComputer.execute(Election.vote("pmurt"))
```

State of the world

State 0 (initial)
(nothing happened yet)

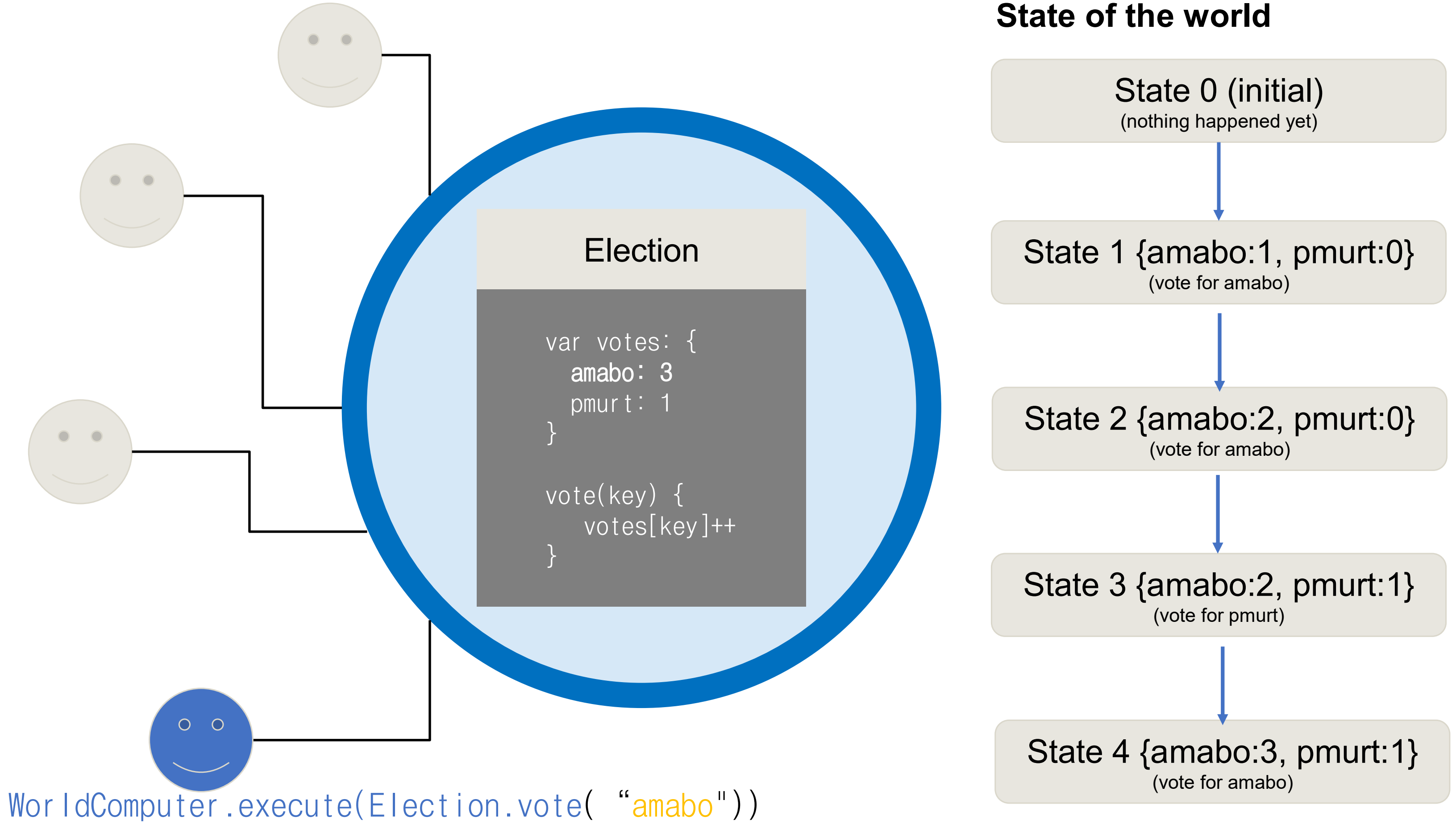
State 1 {amabo:1, pmurt:0}
(vote for amabo)

State 2 {amabo:2, pmurt:0}
(vote for amabo)

State 3 {amabo:2, pmurt:1}
(vote for pmurt)

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



```
var votes: {
  amabo: 3
  pmurt: 1
}

vote(key) {
    votes[key]++
}
```

```
WorldComputer.execute(Election.vote( “amabo"))
```

State of the world

State 0 (initial)
(nothing happened yet)

State 1 {amabo:1, pmurt:0}
(vote for amabo)

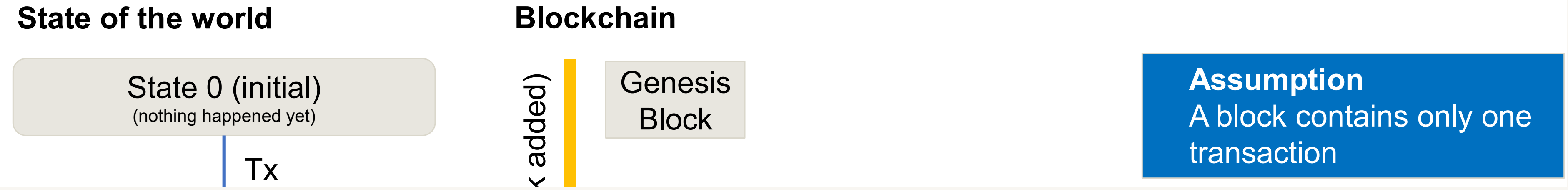
State 2 {amabo:2, pmurt:0}
(vote for amabo)

State 3 {amabo:2, pmurt:1}
(vote for pmurt)

State 4 {amabo:3, pmurt:1}
(vote for amabo)

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



State of the world

State 0 (initial)
(nothing happened yet)

Tx

Blockchain

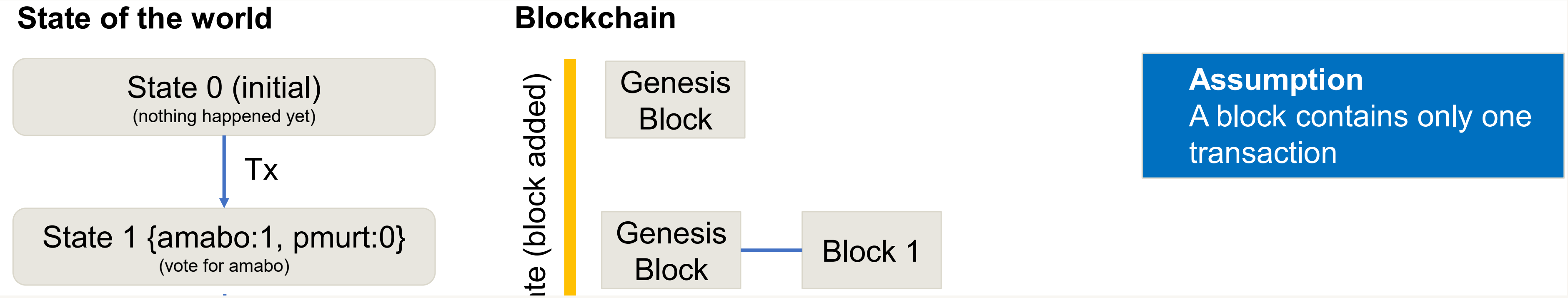
k added)

Genesis Block

Assumption
A block contains only one transaction

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



State of the world

State 0 (initial)
(nothing happened yet)

Tx

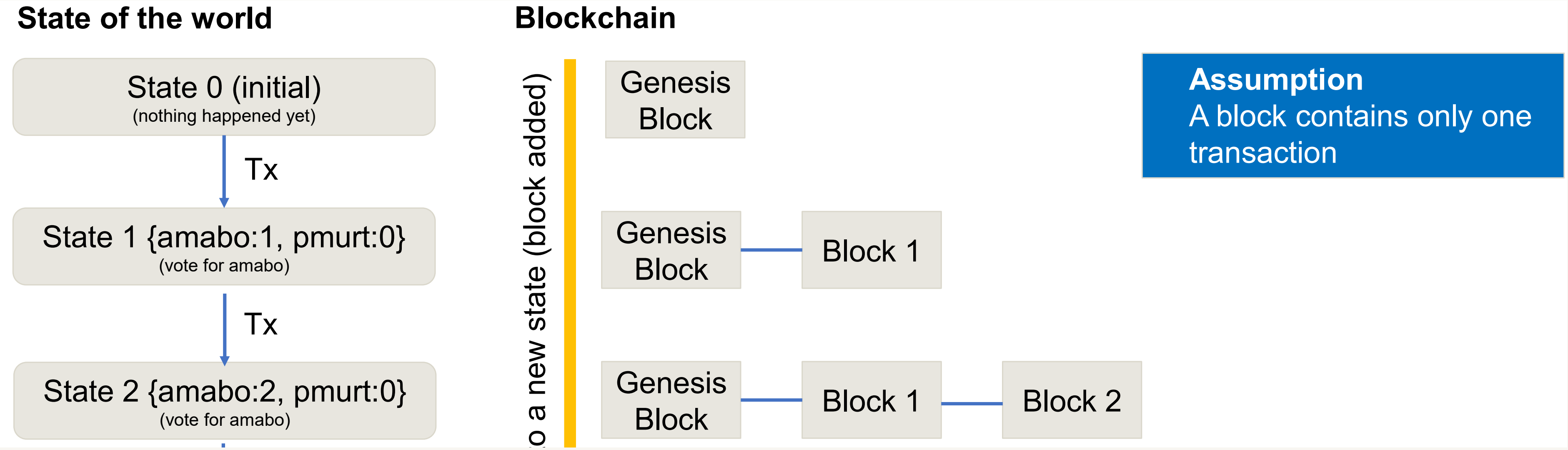
State 1 {amabo:1, pmurt:0}
(vote for amabo)

Blockchain

Assumption
A block contains only one transaction

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



State of the world

State 0 (initial)
(nothing happened yet)

Tx

State 1 {amabo:1, pmurt:0}
(vote for amabo)

Tx

State 2 {amabo:2, pmurt:0}
(vote for amabo)

Blockchain

o a new state (block added)

Genesis Block

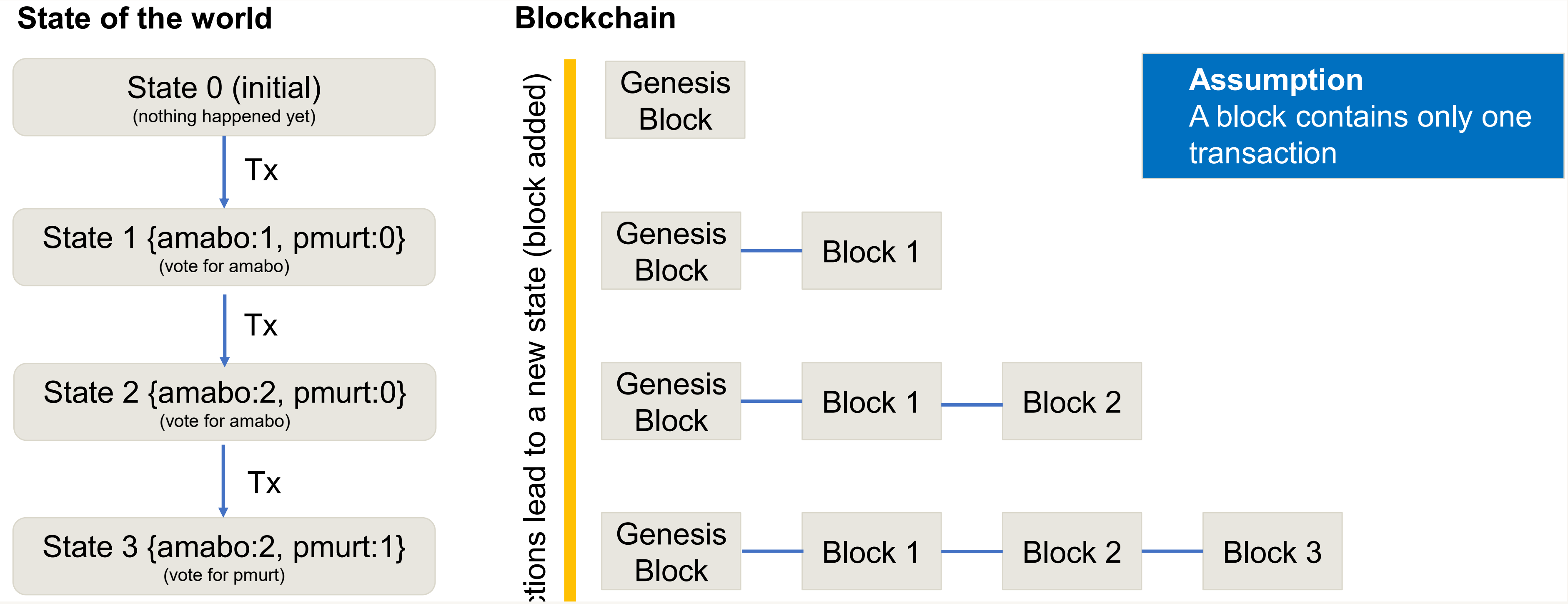
Genesis Block — Block 1

Genesis Block — Block 1 — Block 2

Assumption
A block contains only one transaction

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



State of the world

State 0 (initial)
(nothing happened yet)

Tx

State 1 {amabo:1, pmurt:0}
(vote for amabo)

Tx

State 2 {amabo:2, pmurt:0}
(vote for amabo)

Tx

State 3 {amabo:2, pmurt:1}
(vote for pmurt)

Blockchain

ctions lead to a new state (block added)

Genesis Block

Genesis Block — Block 1

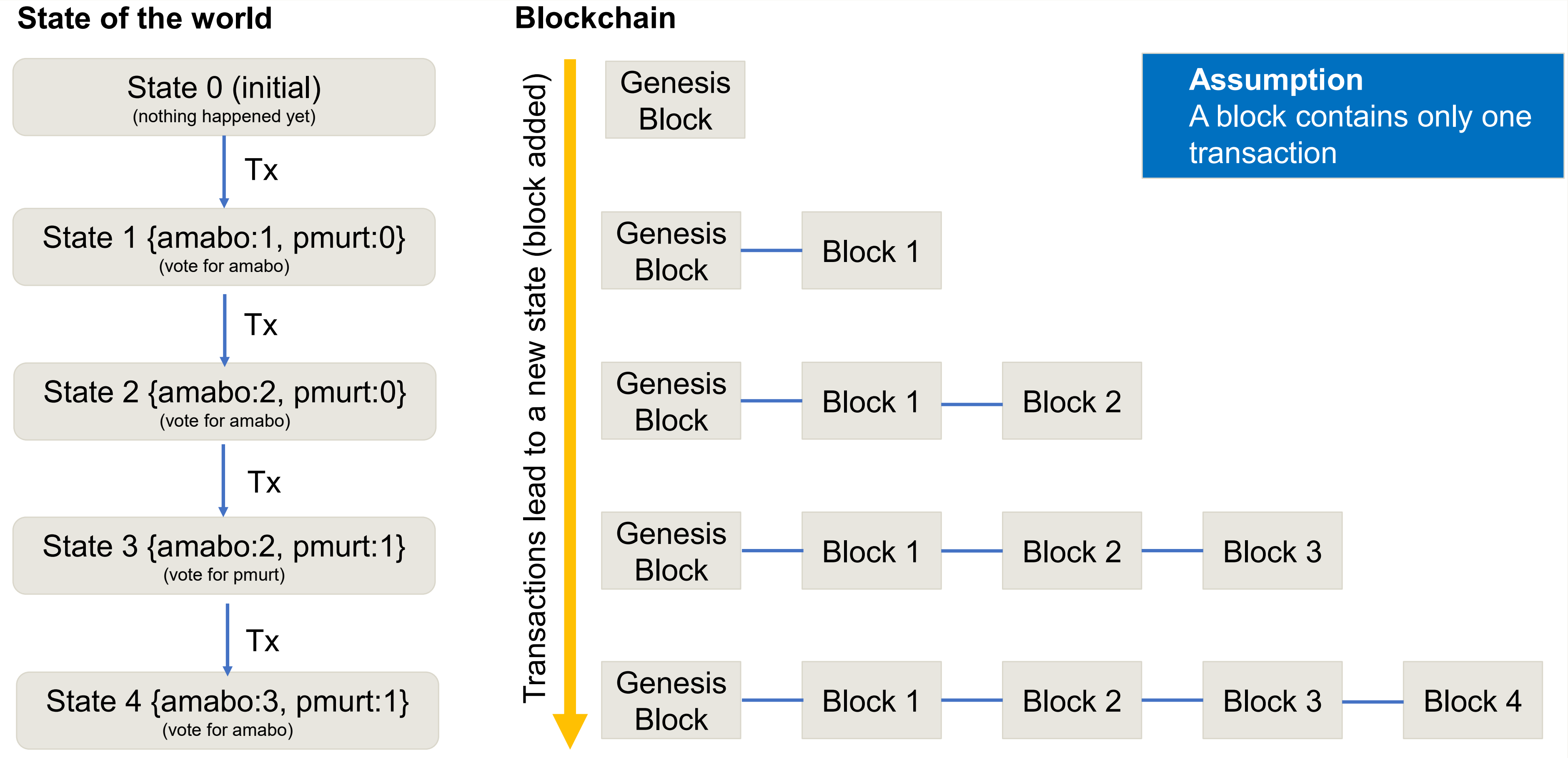
Genesis Block — Block 1 — Block 2

Genesis Block — Block 1 — Block 2 — Block 3

Assumption
A block contains only one transaction

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Ethereum visualization: election example



State of the world

State 0 (initial)
(nothing happened yet)

Tx

State 1 {amabo:1, pmurt:0}
(vote for amabo)

Tx

State 2 {amabo:2, pmurt:0}
(vote for amabo)

Tx

State 3 {amabo:2, pmurt:1}
(vote for pmurt)

Tx

State 4 {amabo:3, pmurt:1}
(vote for amabo)

Blockchain

Transactions lead to a new state (block added)

Genesis Block

Genesis Block — Block 1

Genesis Block — Block 1 — Block 2

Genesis Block — Block 1 — Block 2 — Block 3

Genesis Block — Block 1 — Block 2 — Block 3 — Block 4

Assumption
A block contains only one transaction

07 Ethereum Basics - Gallersdörfer, U., Holl, P., & Matthes, F. (2020). "Blockchain-based Systems Engineering". Lecture Slides. TU Munich.

Question?