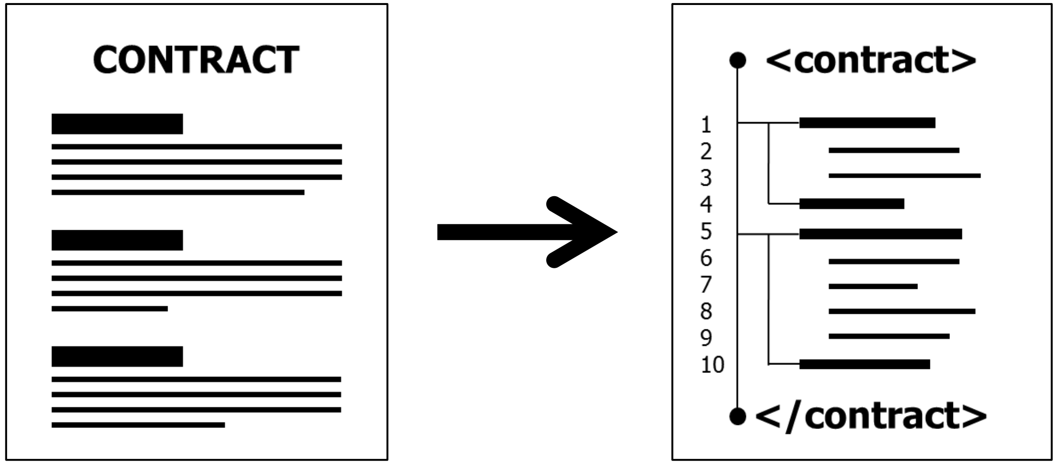
Smart Contracts

What is all the fuzz about?

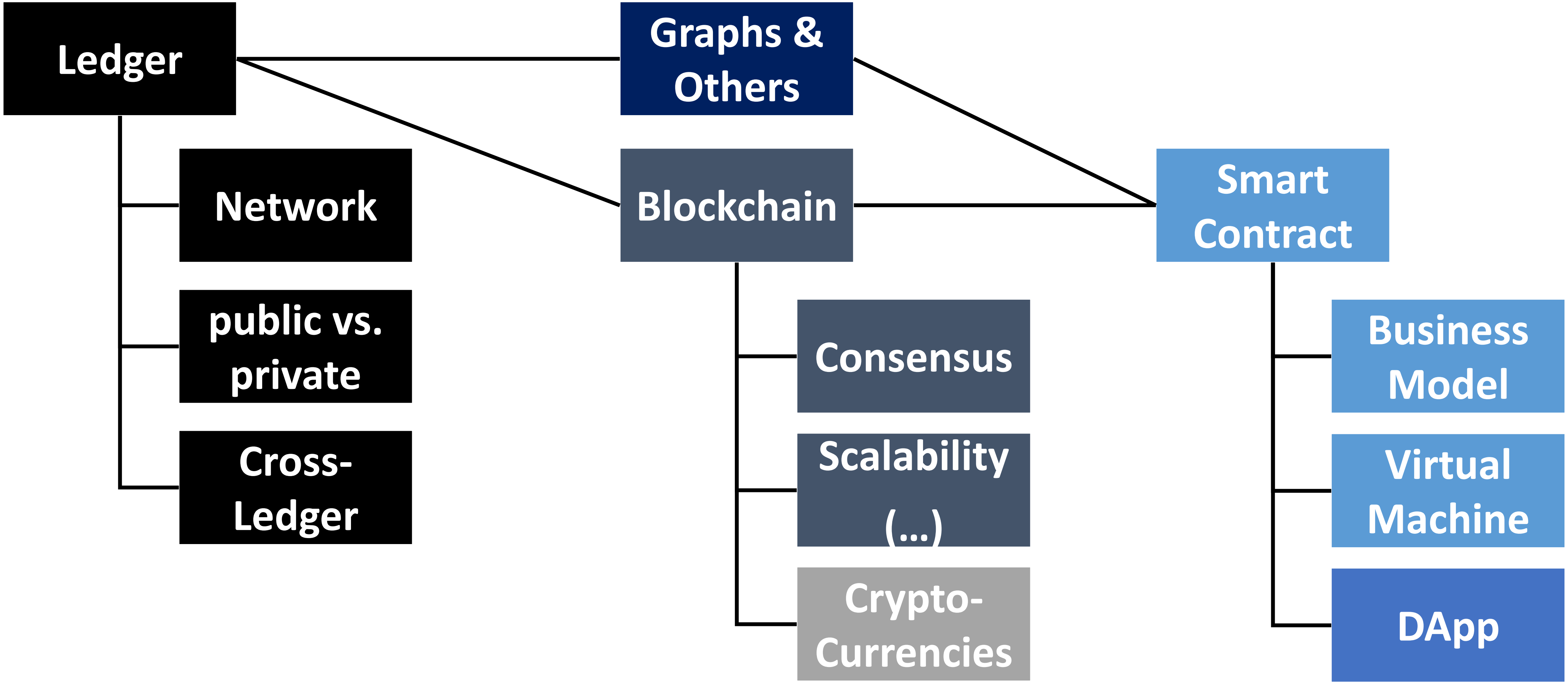


SENACOR

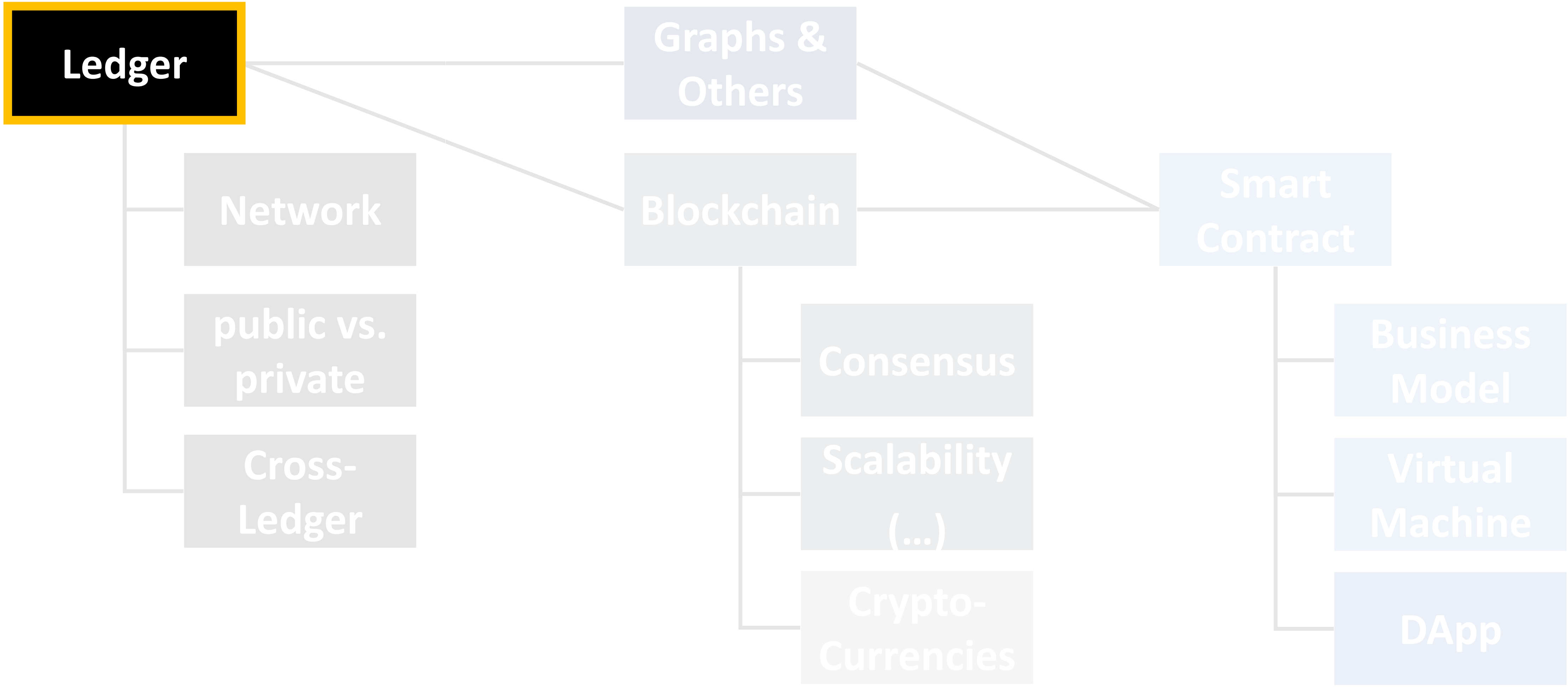
Intro & Terms

Ledger, Blockchain, Consensus, Proof Of..., Forks, Alt-/Side-Chain, Coins, Tokens

Fields of Interest



Let‘s focus on…



What is a ledger?

General Ledger				
Account Name: Cash				
Account Number: 001				
Date	Description	Debit	Credit	Balance
19.09.2017	Check from X	500 €		500 €
20.09.2017	Payment to Y		200 €	300 €
21.09.2017	Payment to Z		100 €	200 €

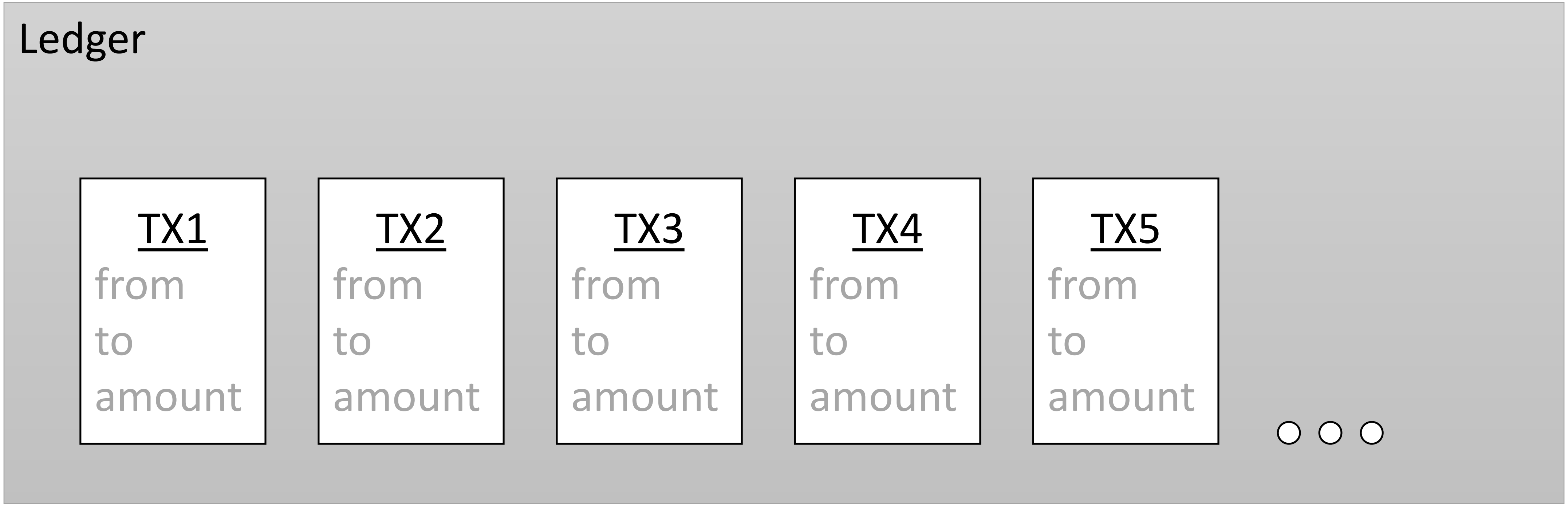
What is a ledger?

Account Ledger				
Account Name: Big Money Bank				
Account Number: DE44 5001 0517 5407 3249 31				
Date	**Description**	**Debit**	**Credit**	**Balance**
19.09.2017	Check from X	500 €		500 €
20.09.2017	Payment to Y		200 €	300 €
21.09.2017	Payment to Z		100 €	200 €

What is a ledger?

Bank Ledger					
Bank Name: Big Money Bank					
Date	**From**	**To**	**Amount**	**Balance From**	**Balance To**
19.09.2017	DE44 1…	DE44 2…	500 €		500 €
20.09.2017	DE44 2…	DE44 3…	200 €	300 €	600 €
21.09.2017	DE44 2…	DE44 4…	100 €	200 €	400 €

What is a ledger?



Ledger

TX1
from
to
amount

TX2
from
to
amount

TX3
from
to
amount

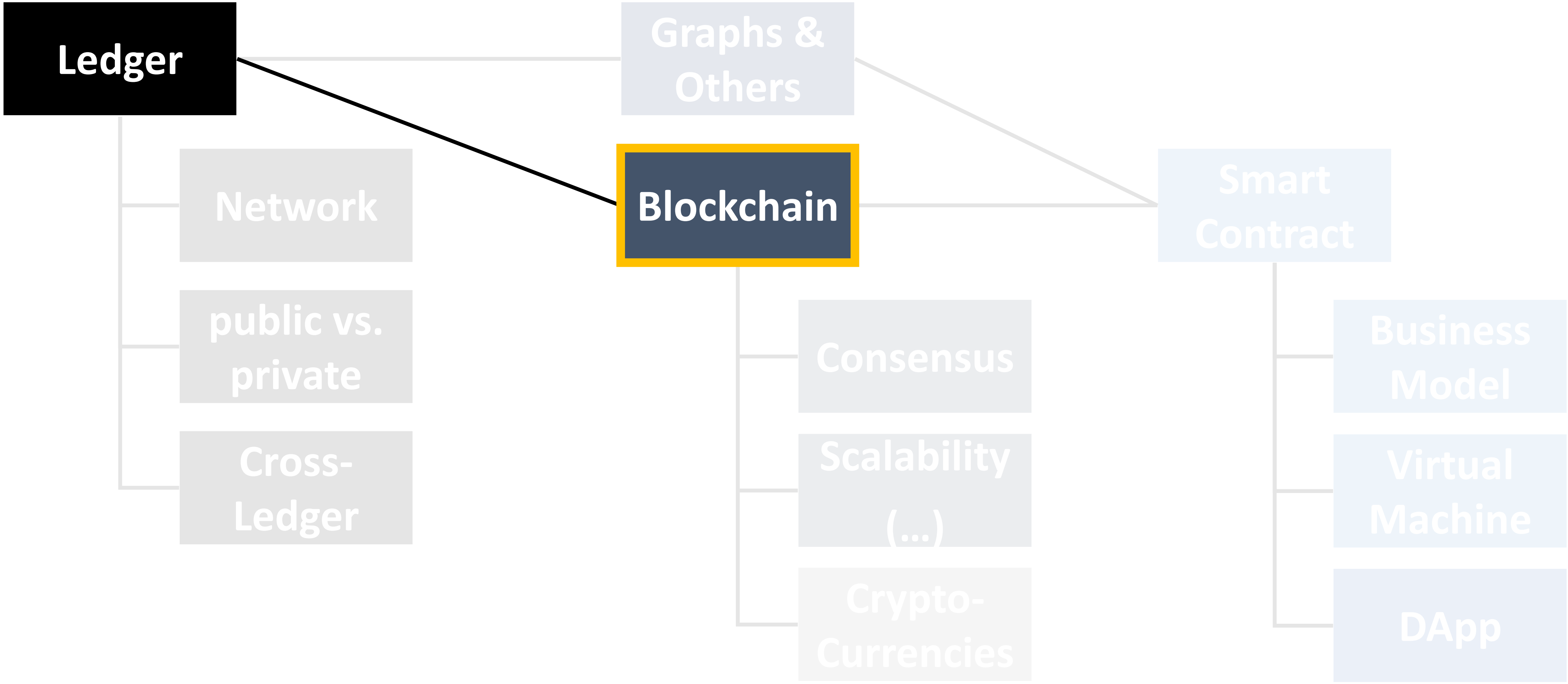
TX4
from
to
amount

TX5
from
to
amount

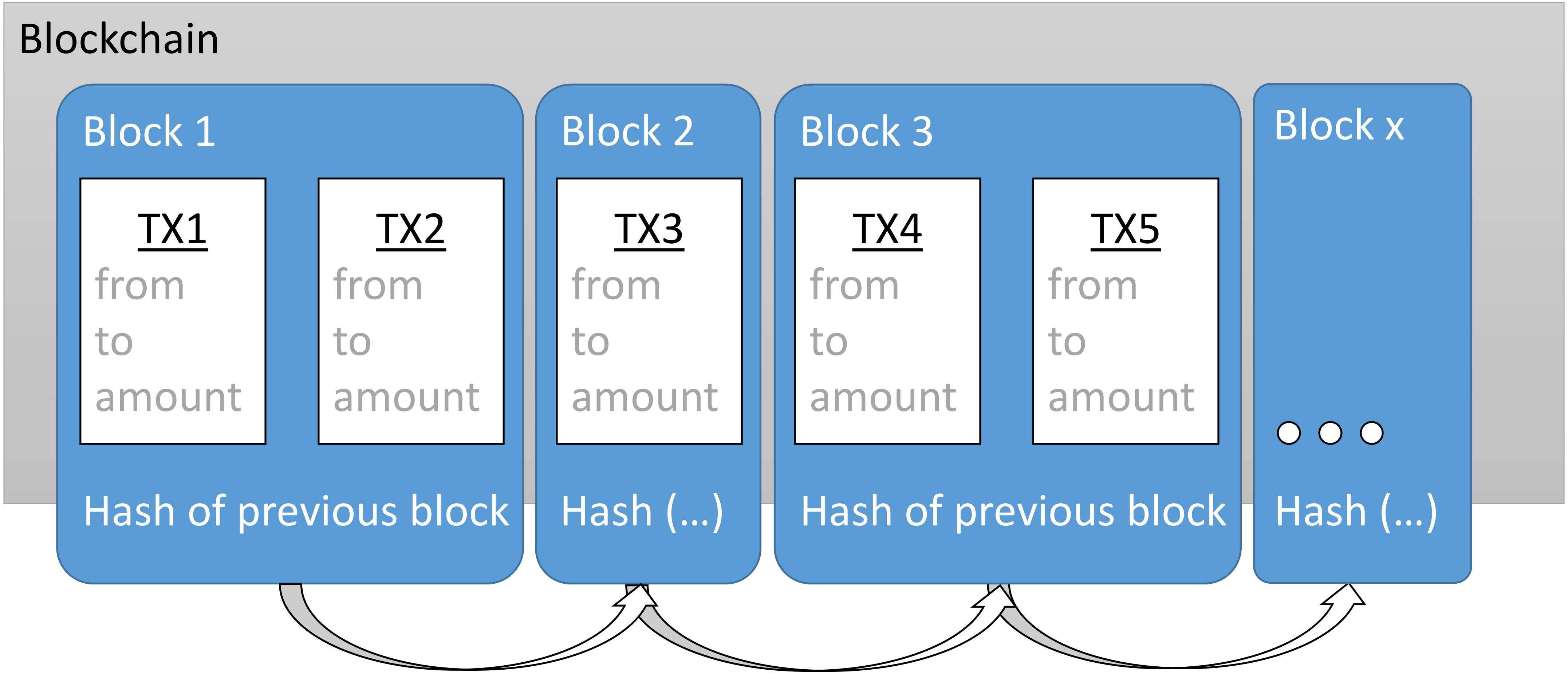
○ ○ ○

You cannot just „delete“ a transaction in a bank, you can just add transactions.

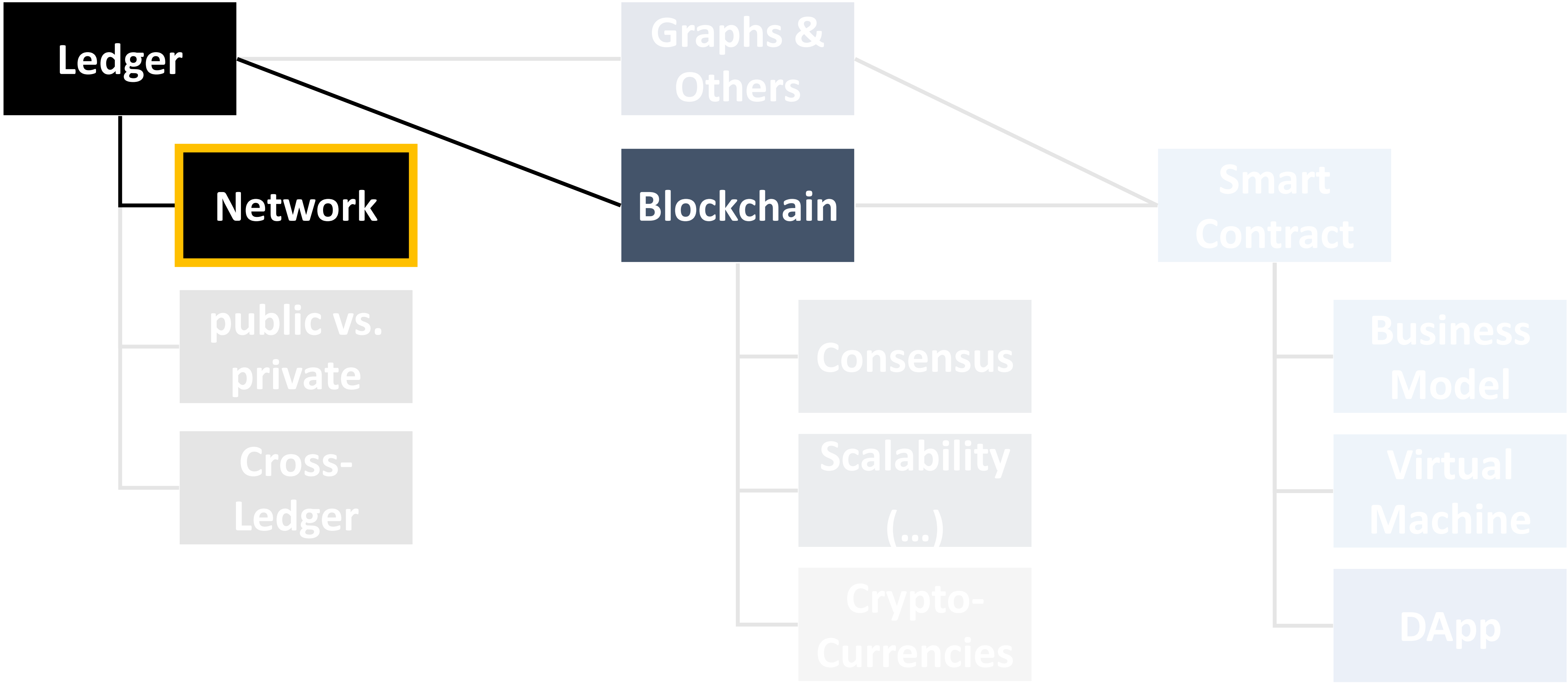
Let‘s focus on…



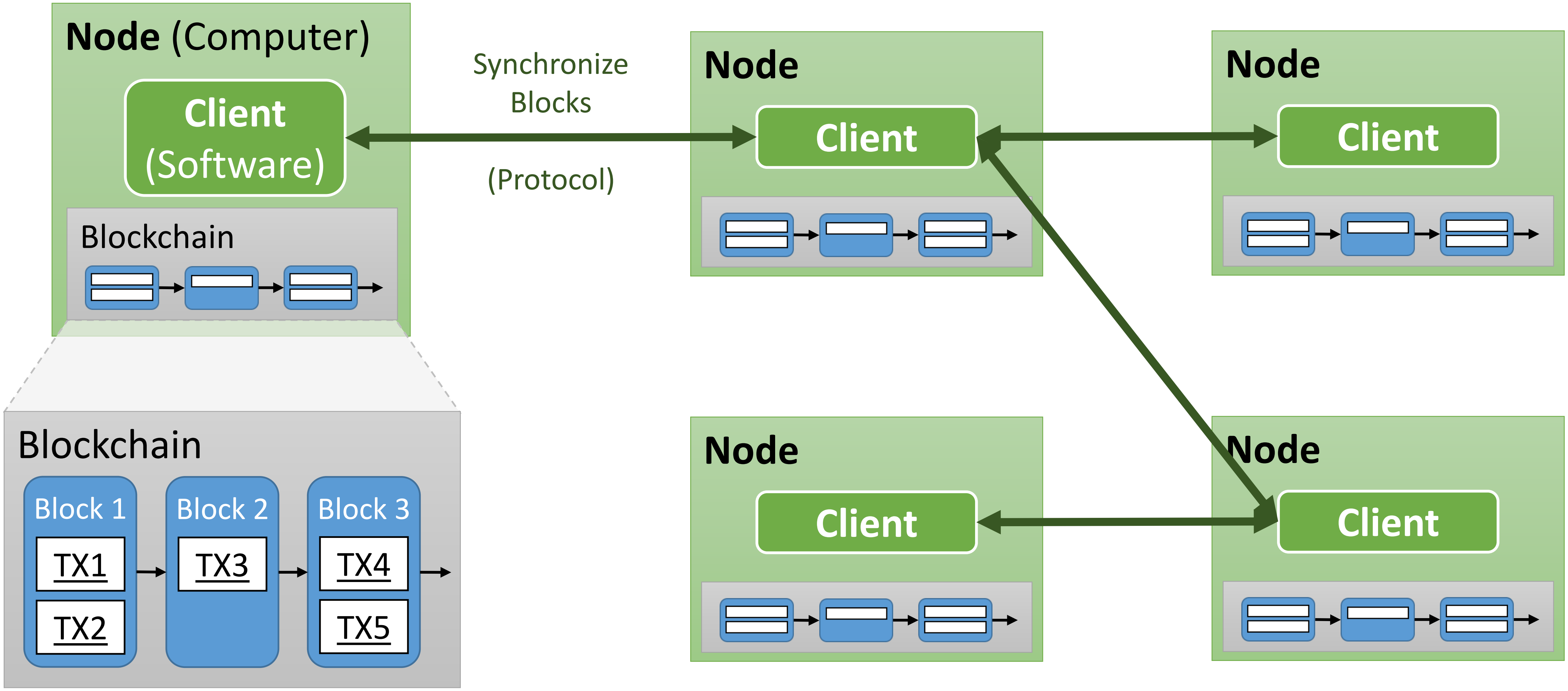
What is the Blockchain?



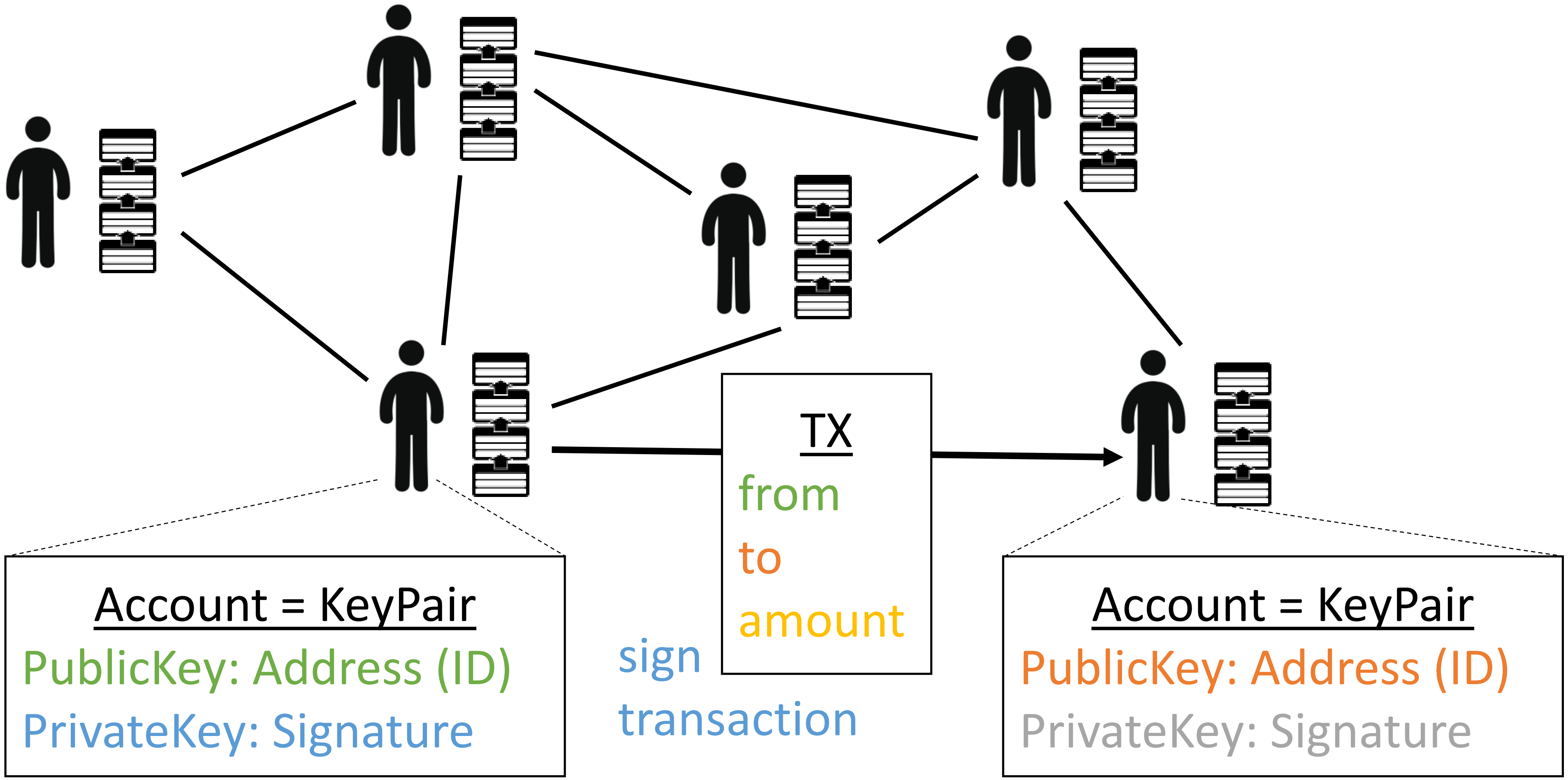
Let‘s focus on…



What is the Blockchain Network?



Identity in the Network: Accounts



TX

from

to

amount

sign transaction

Account = KeyPair

PublicKey: Address (ID)

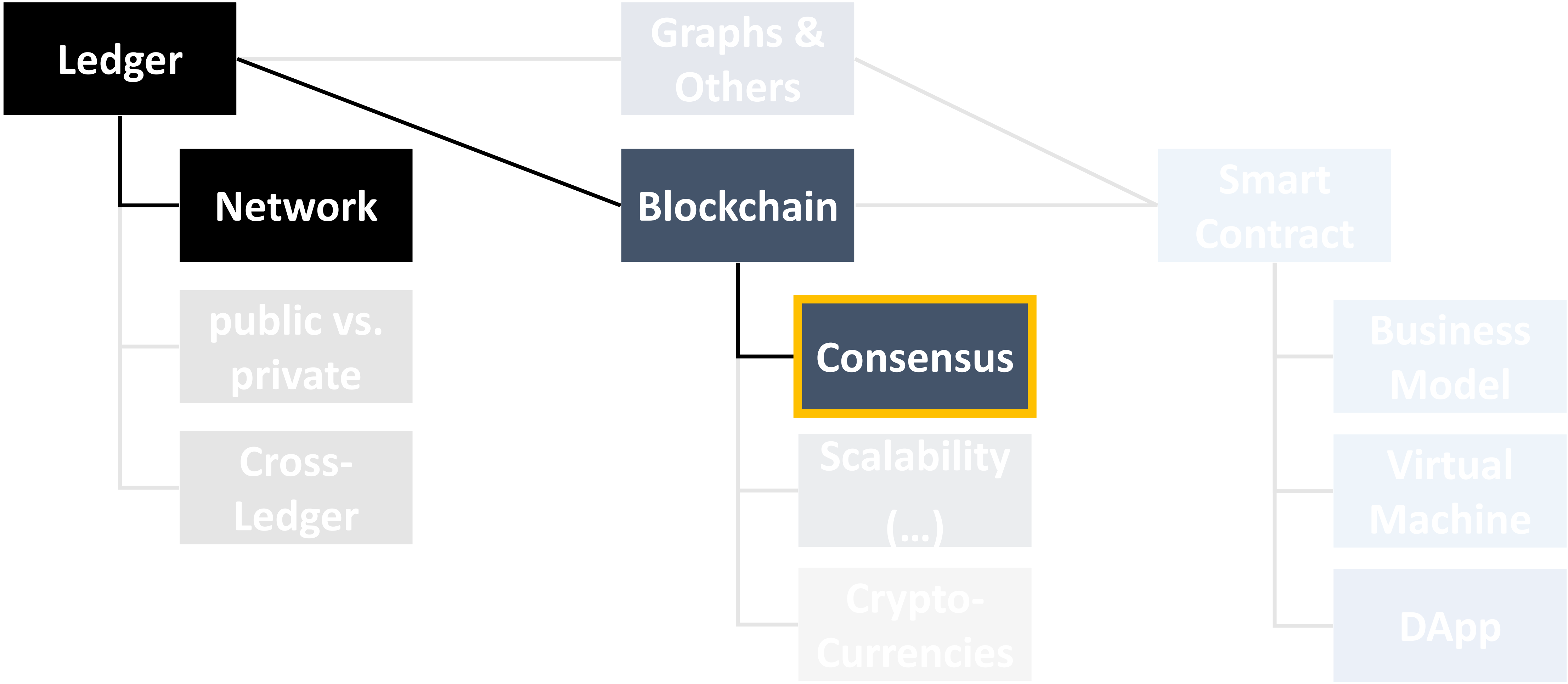
PrivateKey: Signature

Account = KeyPair

PublicKey: Address (ID)

PrivateKey: Signature

Let‘s focus on…



What is the Consensus?

- **Parties** the **don‘t trust** each other **agree** on the **state** of a system at a certain **time**.
- Reaching an Agreement:
  1. Collect state-changes (transactions)
  2. Define a “truth-giver“
  3. Truth-giver validates state-changes
  4. Truth-giver publishes new truth (state) to all others
  5. At least 51% of the nodes confirm the truth

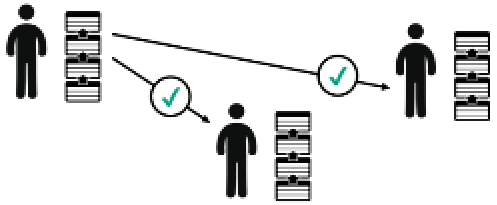
Mining: Building Consensus through PoW

Proof of Work

- Solve a “cryptographic riddle” brute-force
- Difficult to solve – easy to validate (you can imagine a Sudoku)
- Solving takes time, recalculation is virtually impossible

Reach Consensus

- Agree on current state of system based on POW



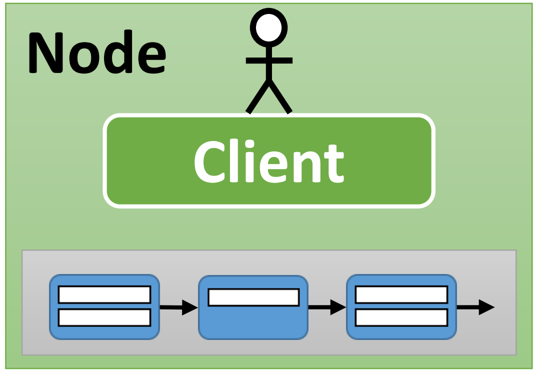
Motivation for Miner

- Transaction fees → Miner
- Money Creation, new money → Miner

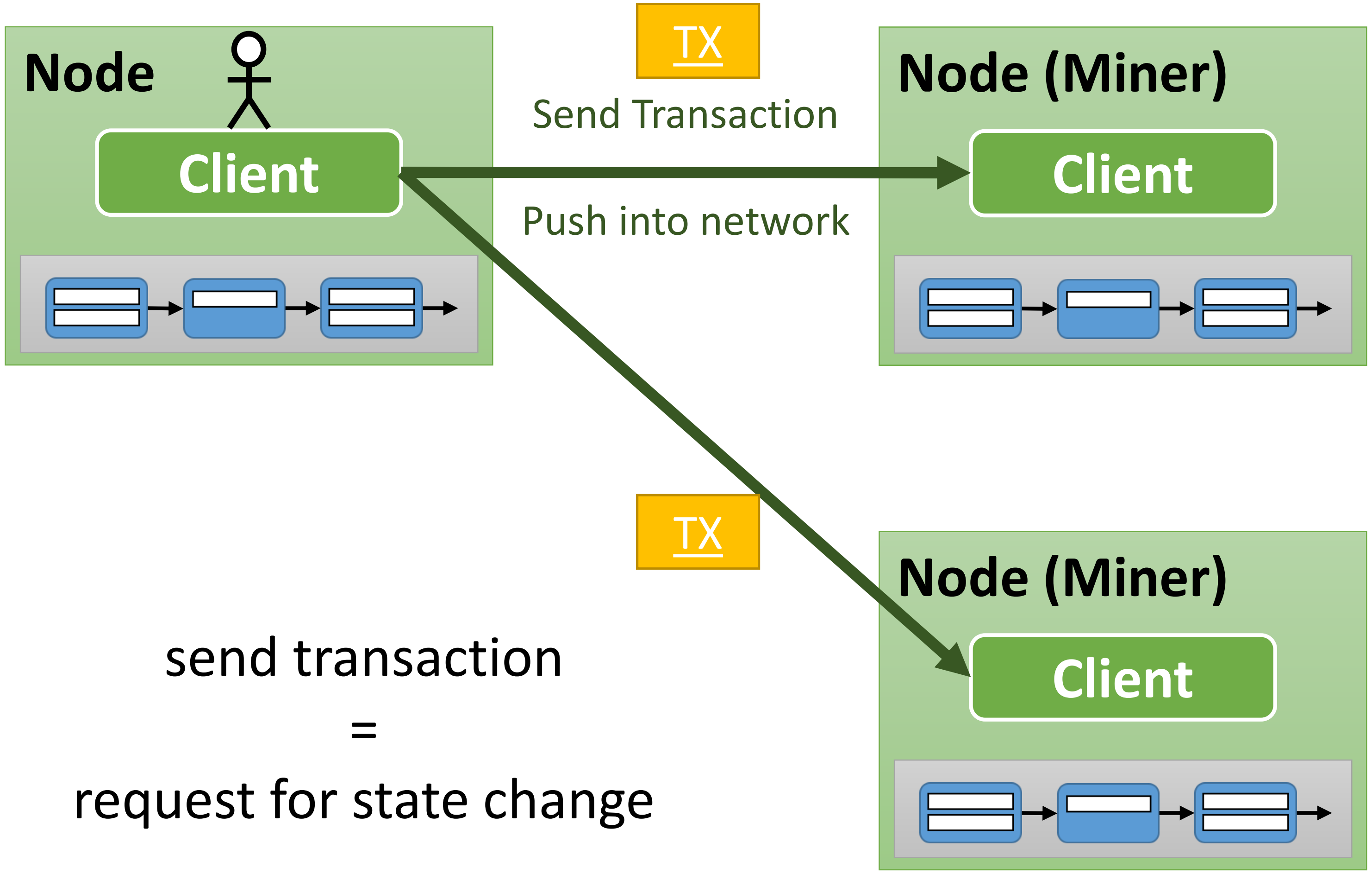
Securing the Network

- Control of the hash value for the last block Verification
- Sender signature
- State change validation
  - Required funds available, …

Proof of Work



Proof of Work

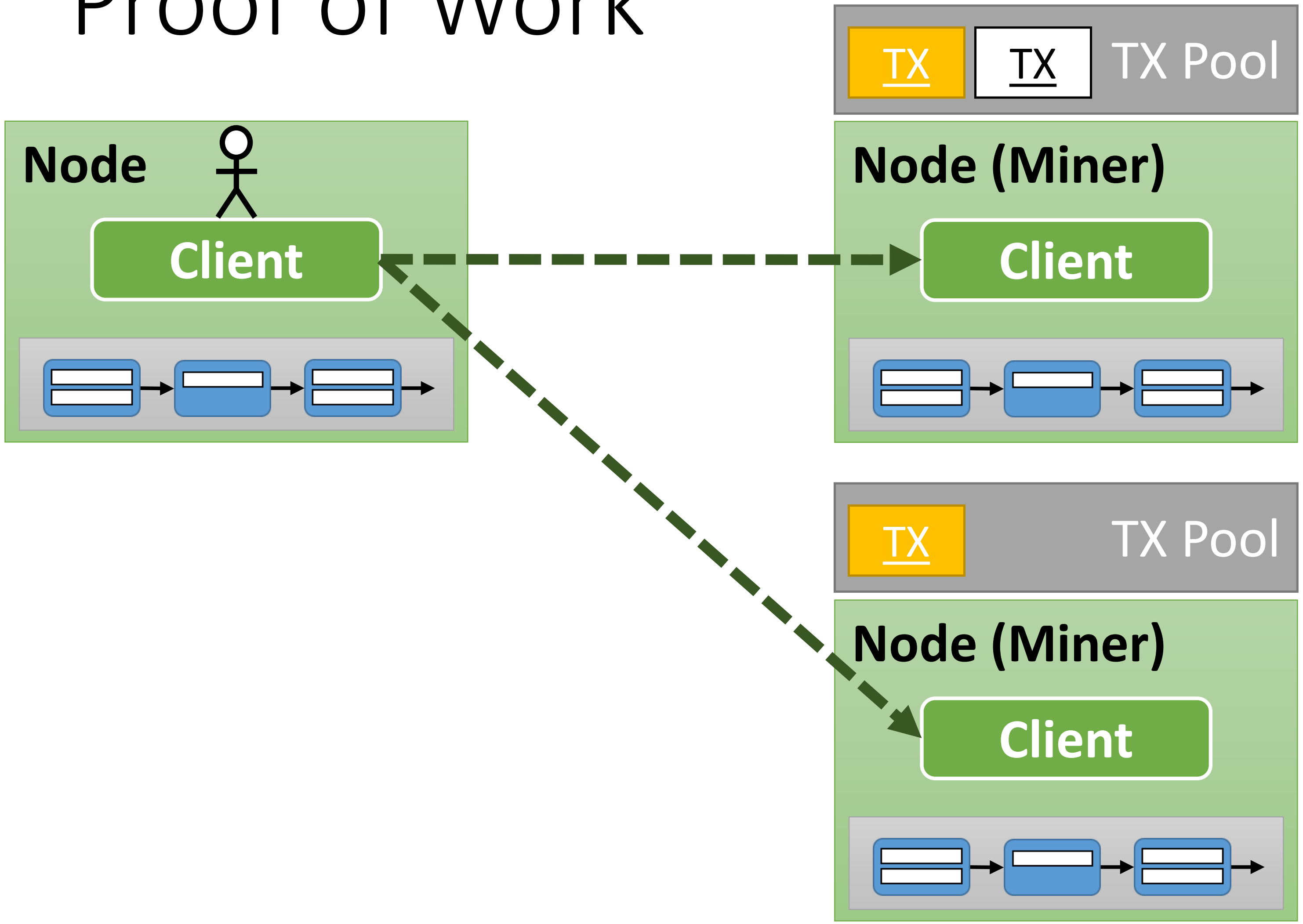


send transaction

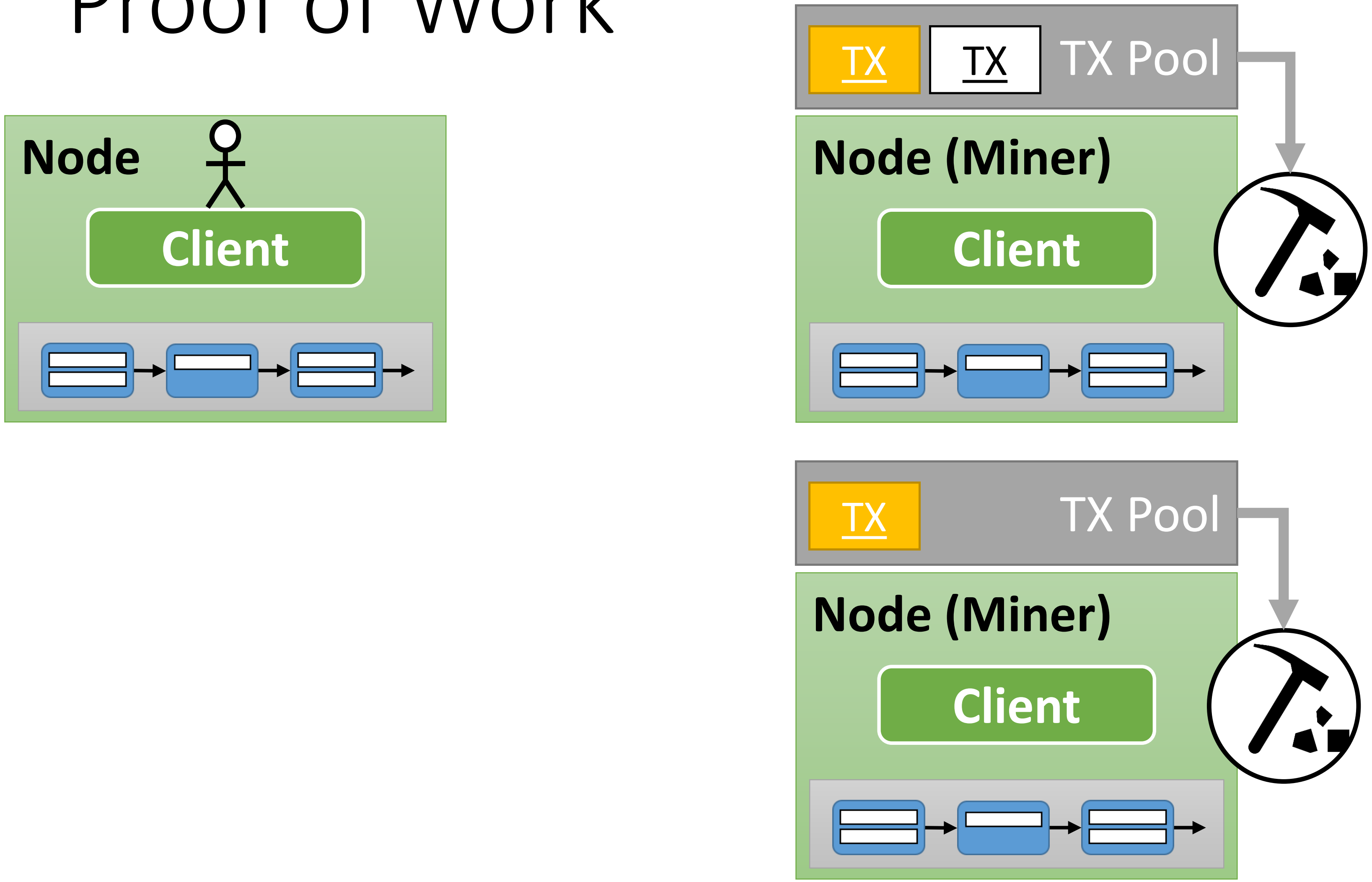
=

request for state change

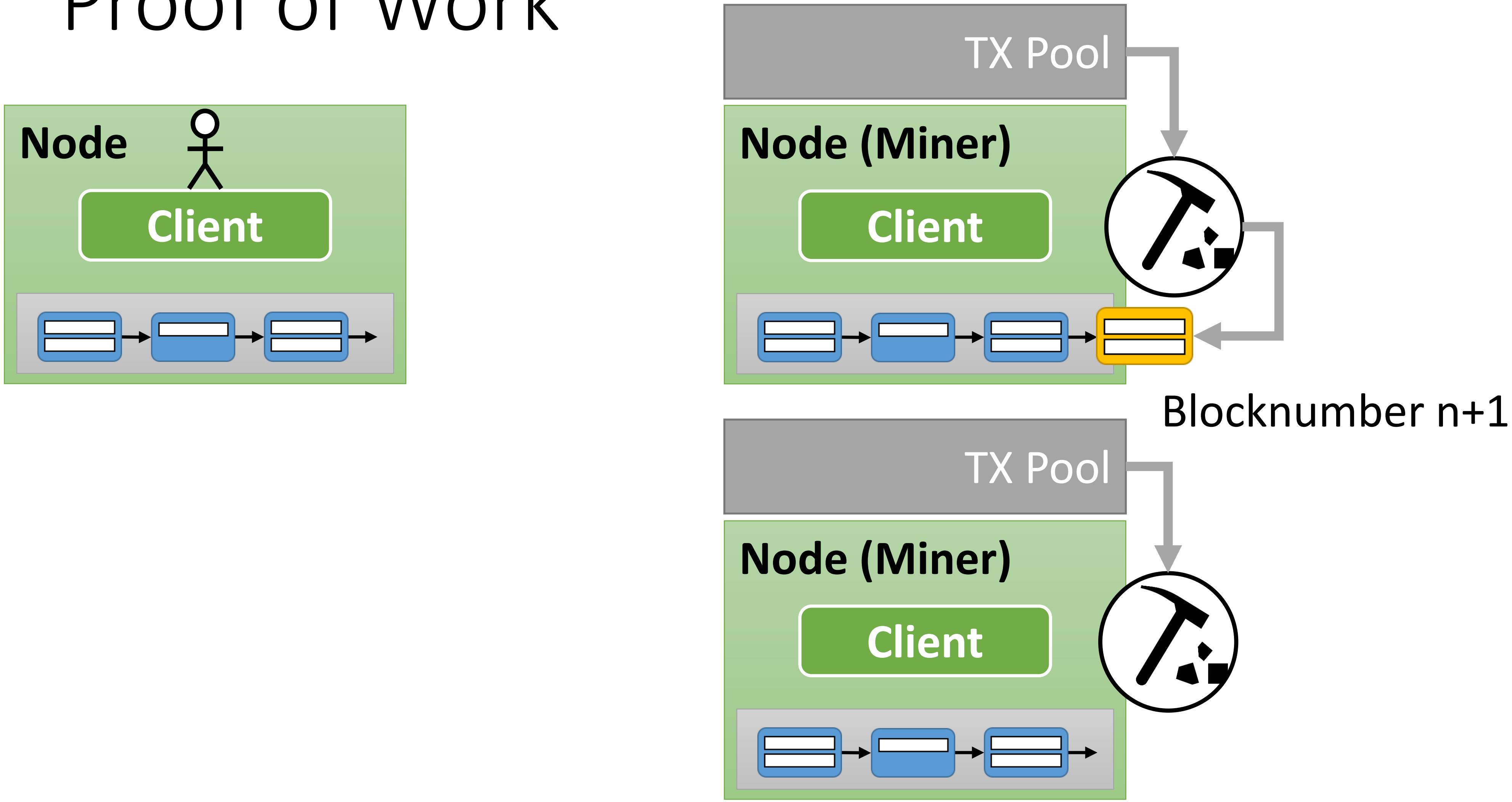
Proof of Work



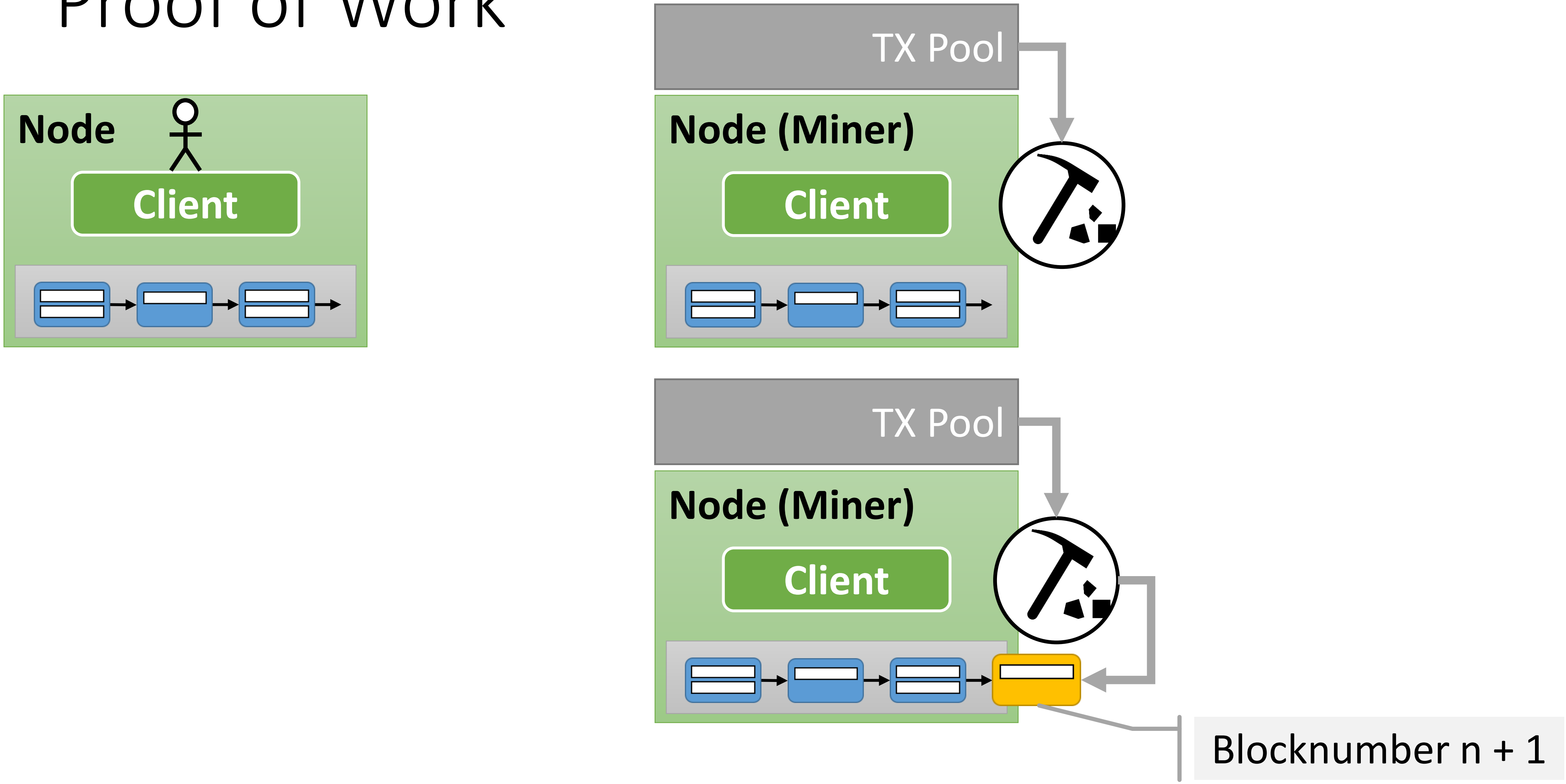
Proof of Work



Proof of Work



Proof of Work



Node

Client

TX Pool

Node (Miner)

Client

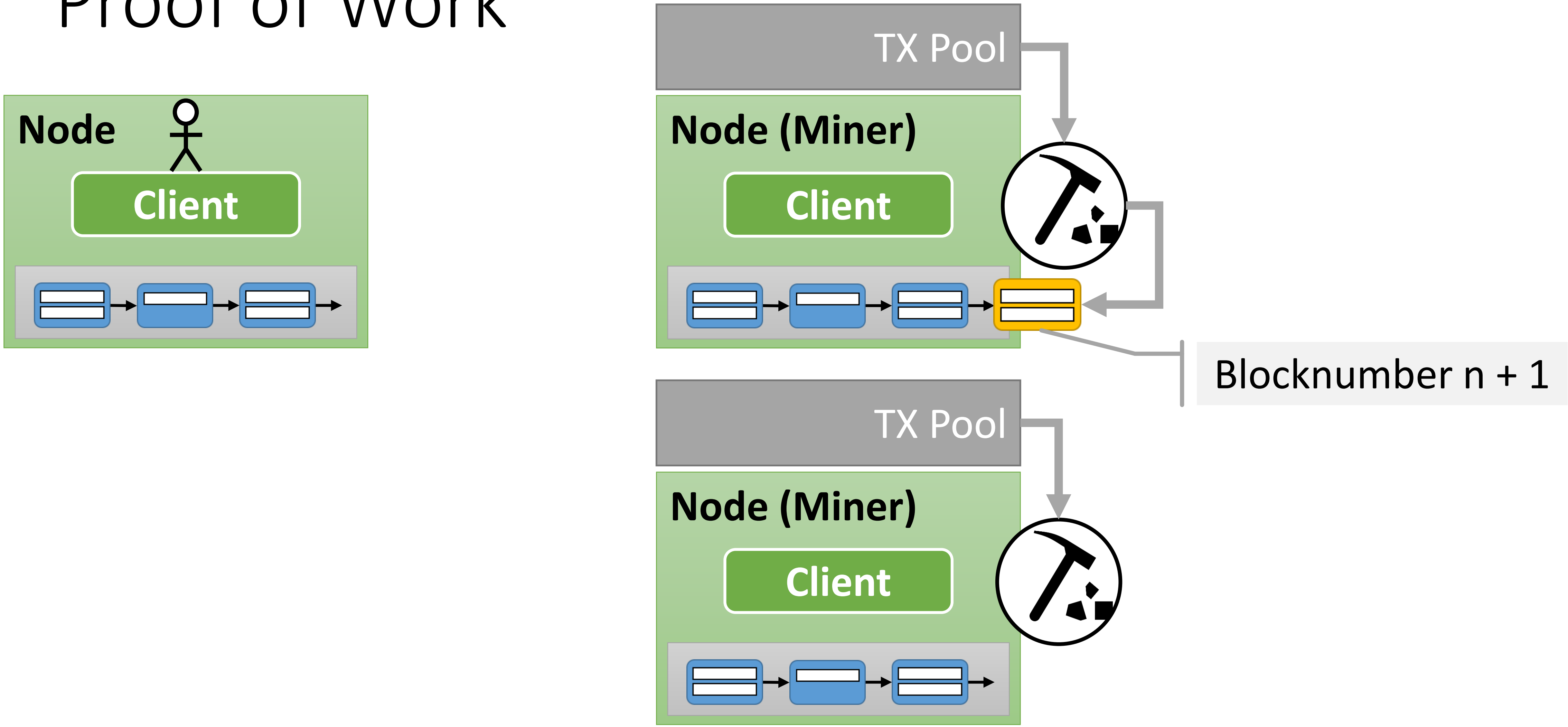
TX Pool

Node (Miner)

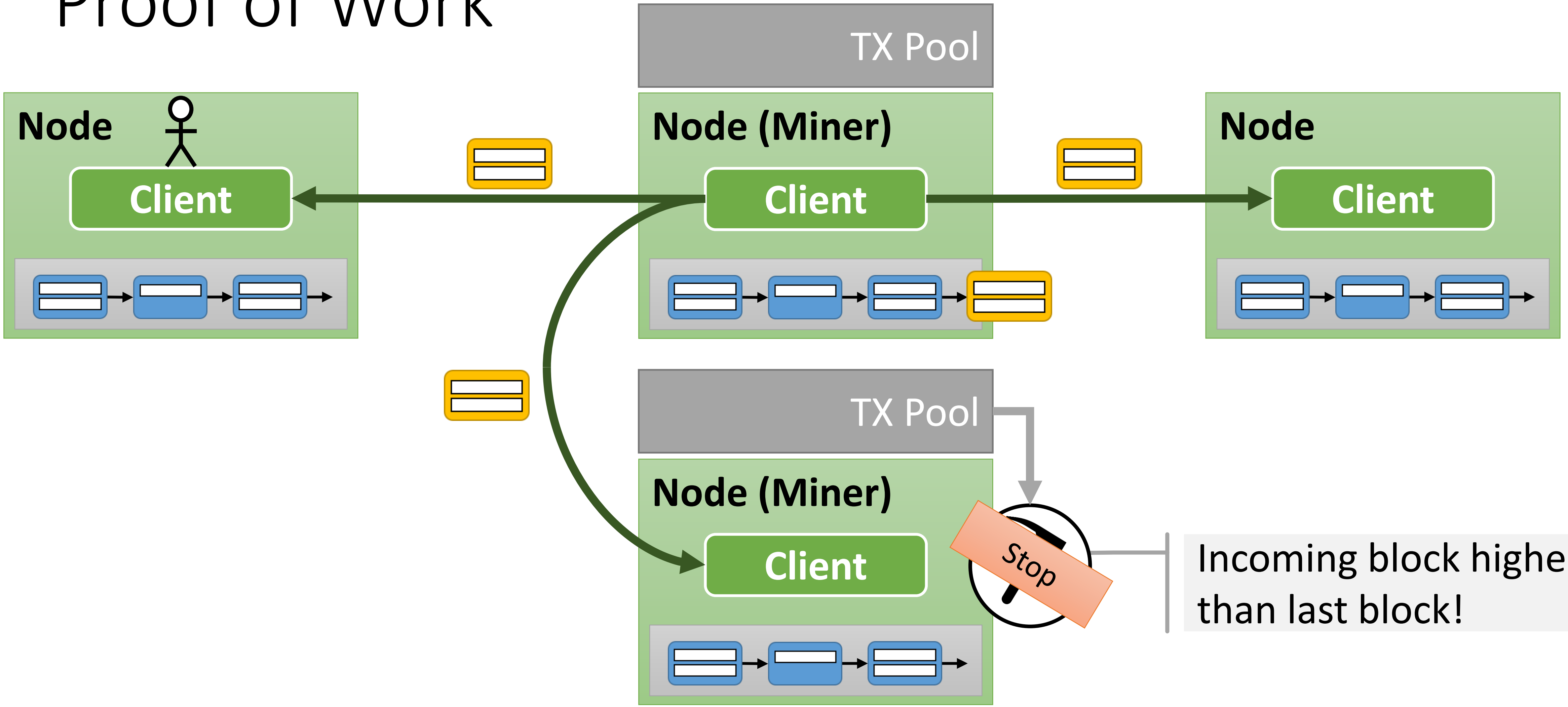
Client

Blocknumber n + 1

Proof of Work

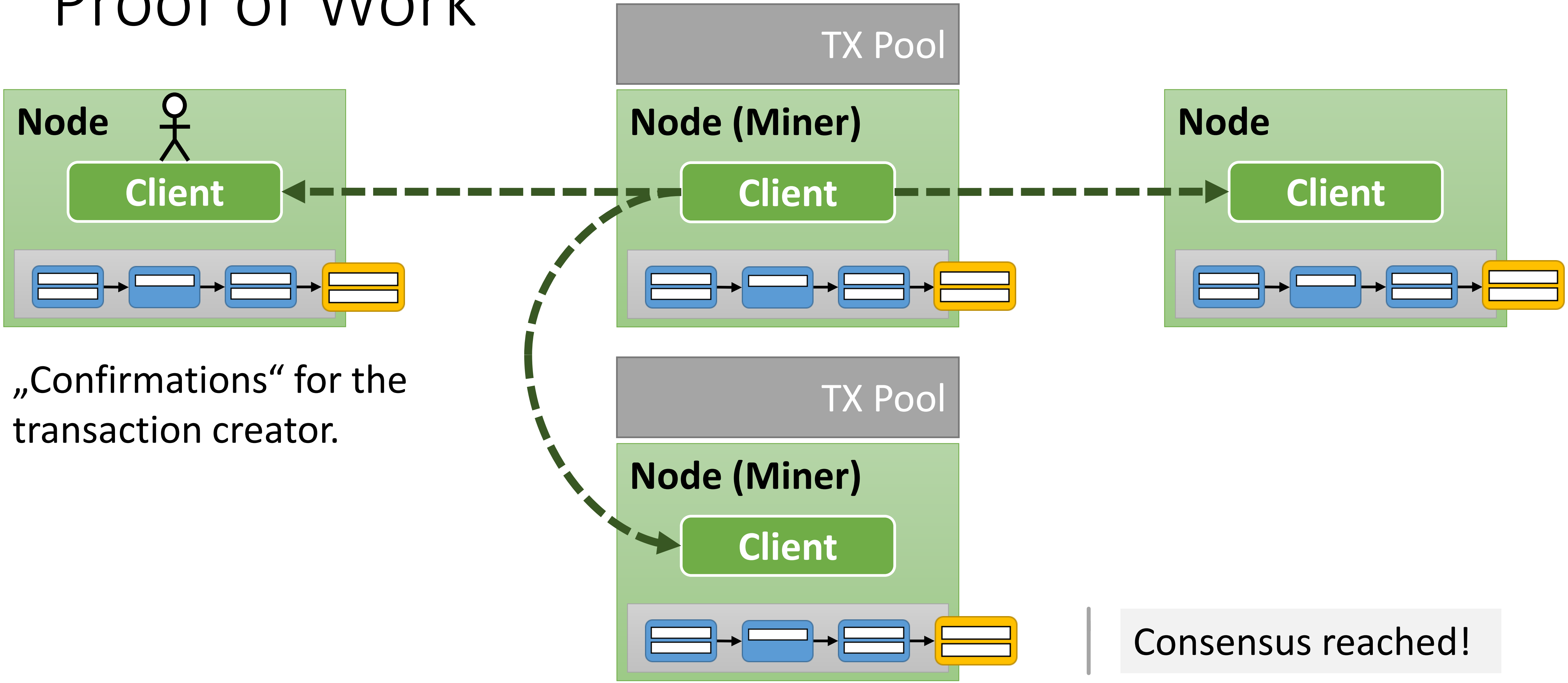


Proof of Work



Incoming block higher than last block!

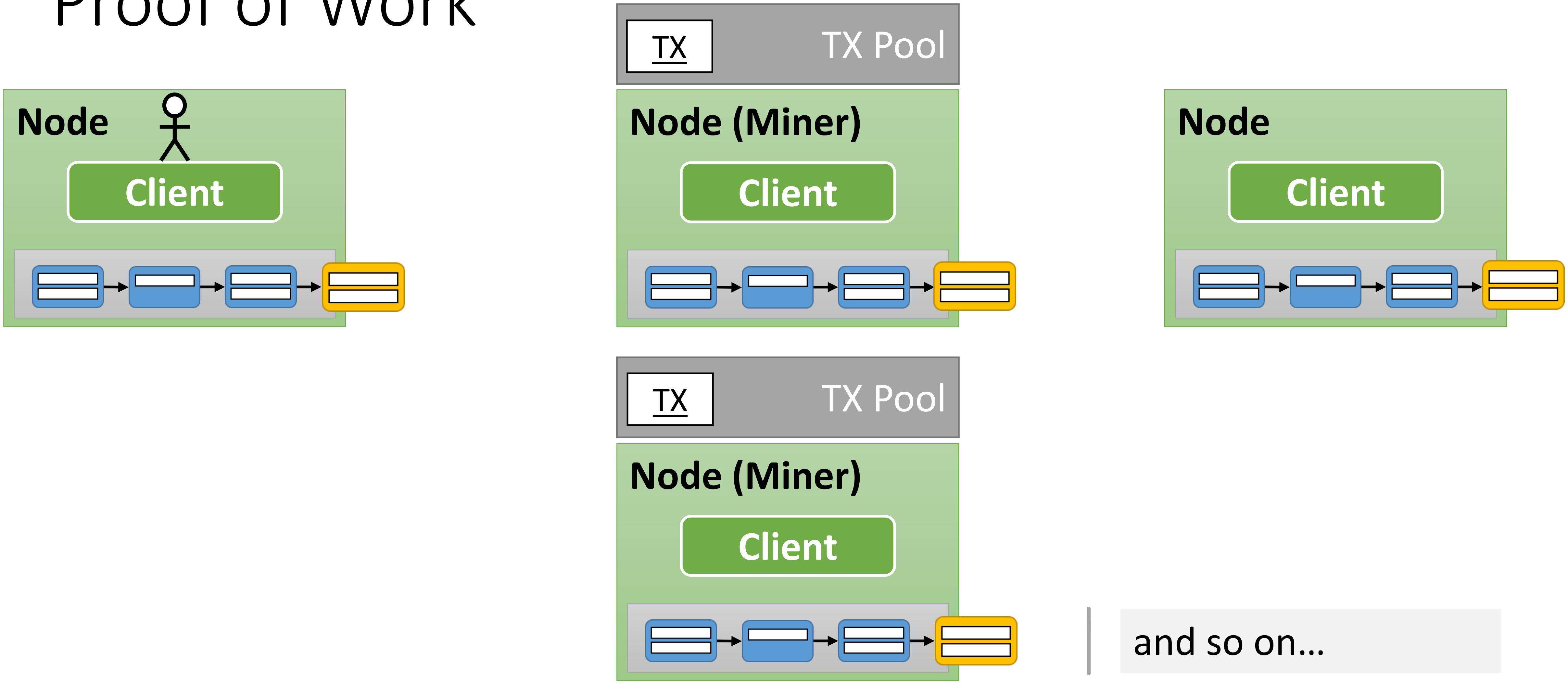
Proof of Work



„Confirmations“ for the transaction creator.

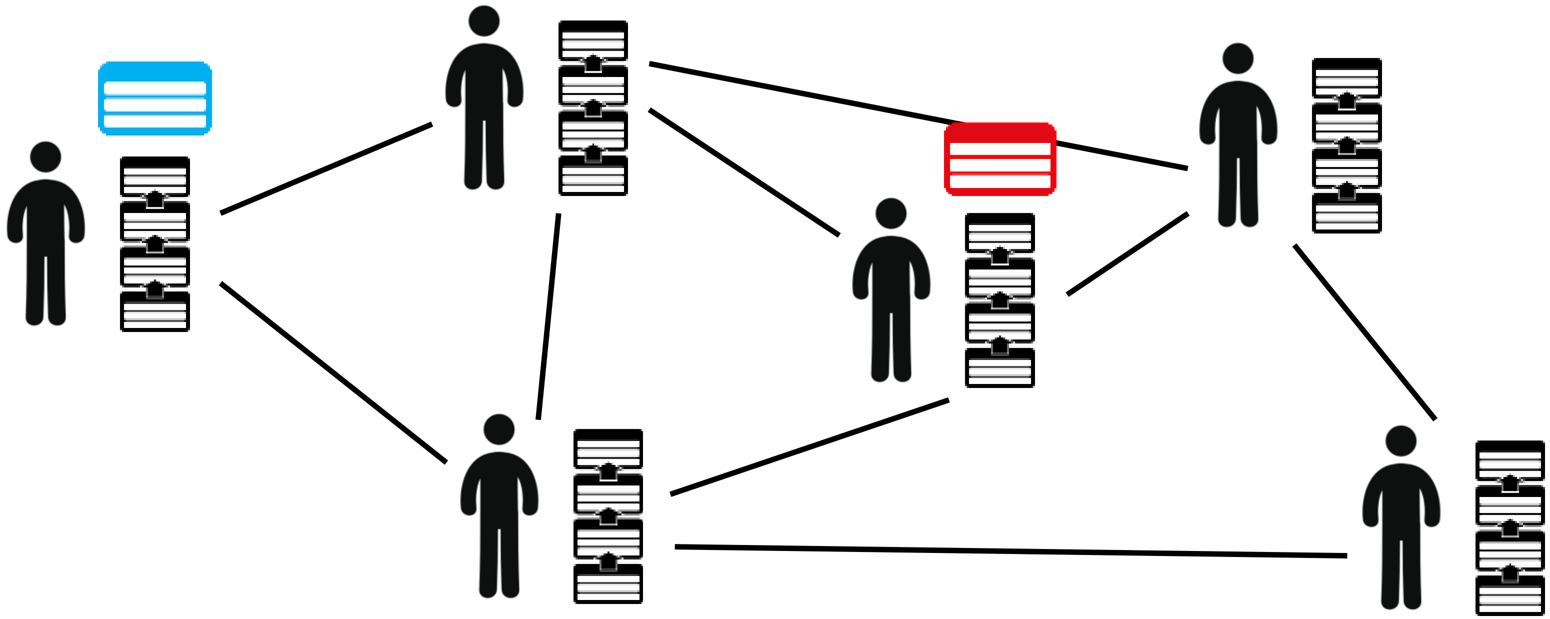
Consensus reached!

Proof of Work

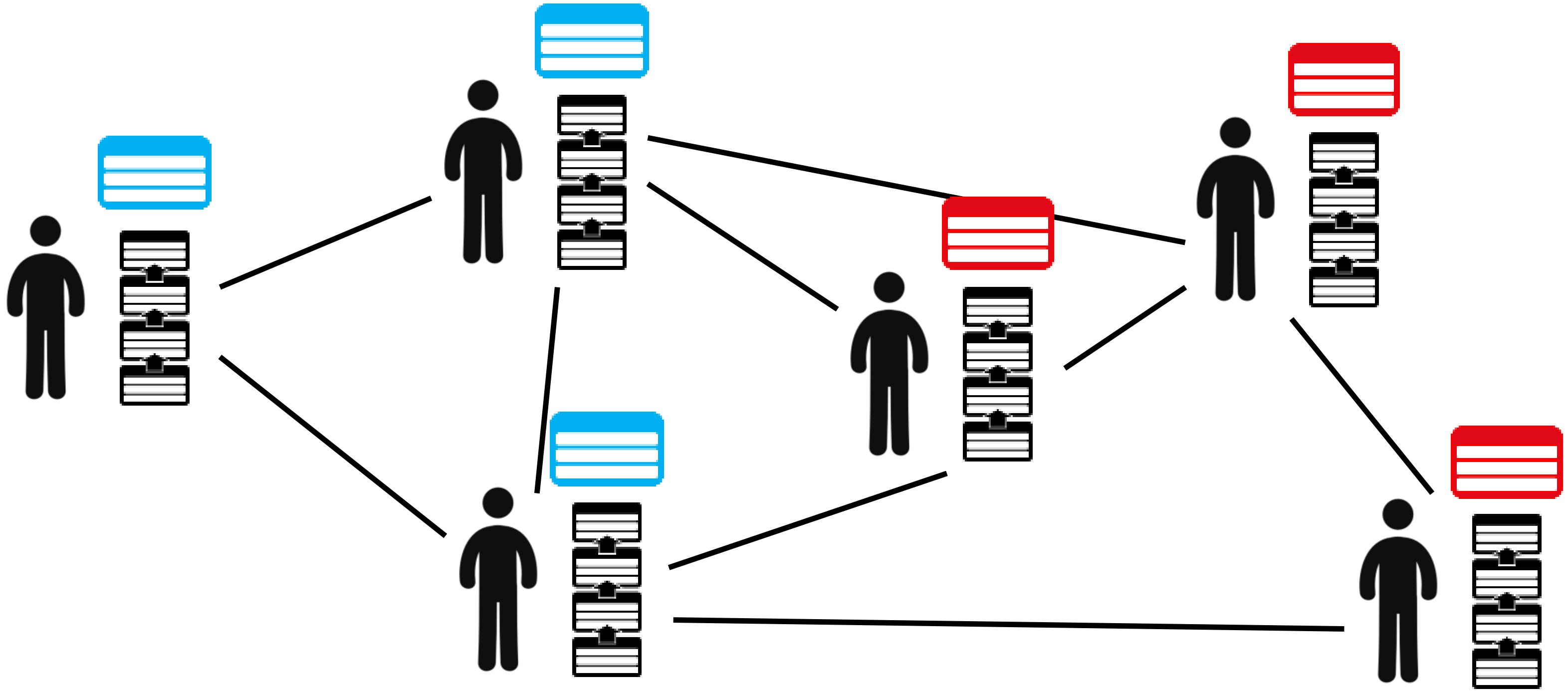


and so on…

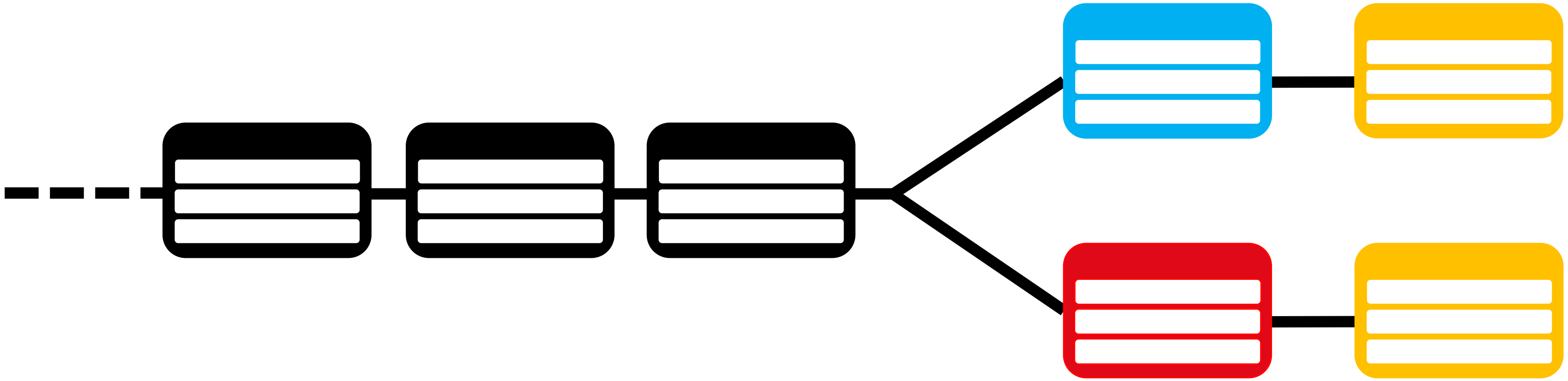
What about: Two winners at the same time?



Network Split

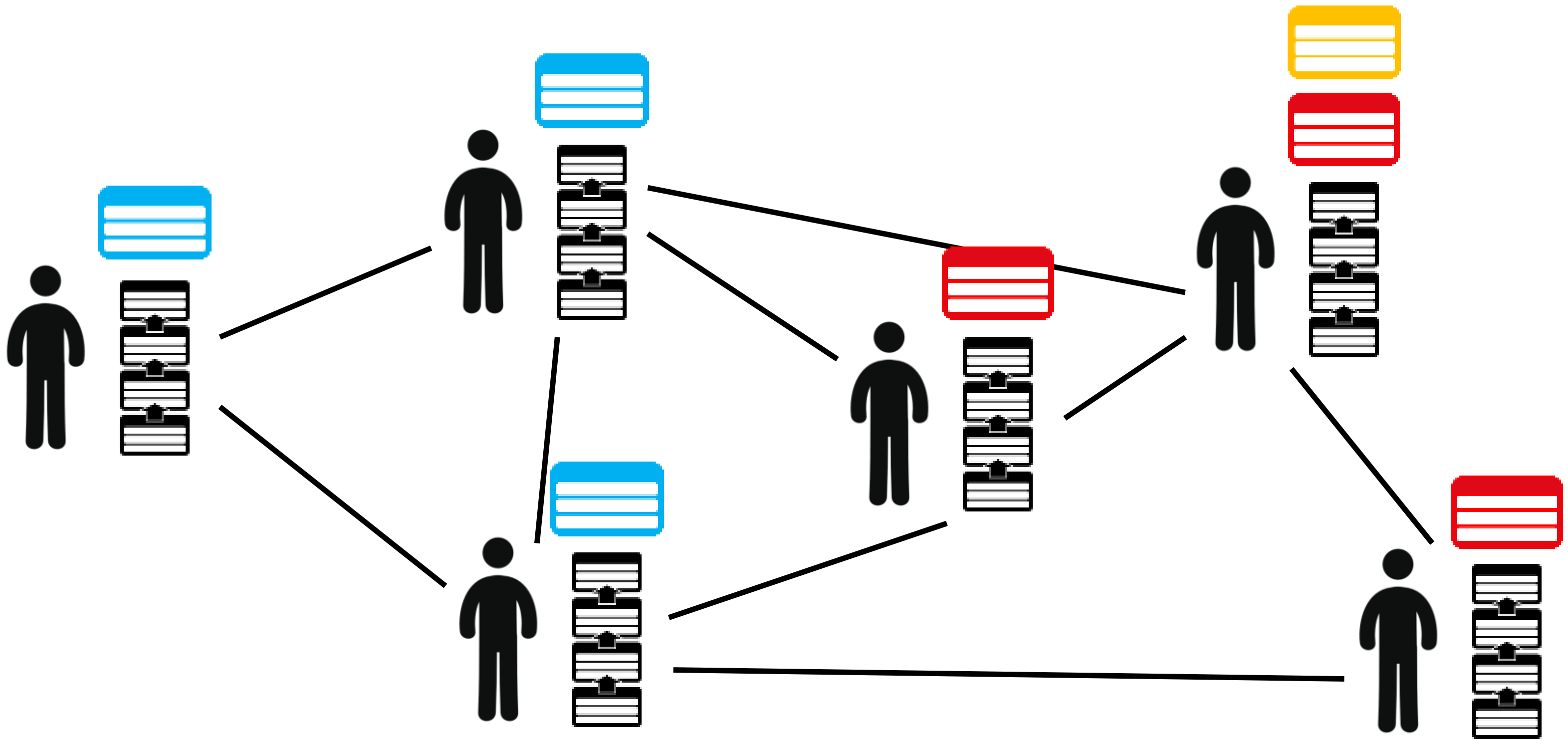


We call this: **a Fork**

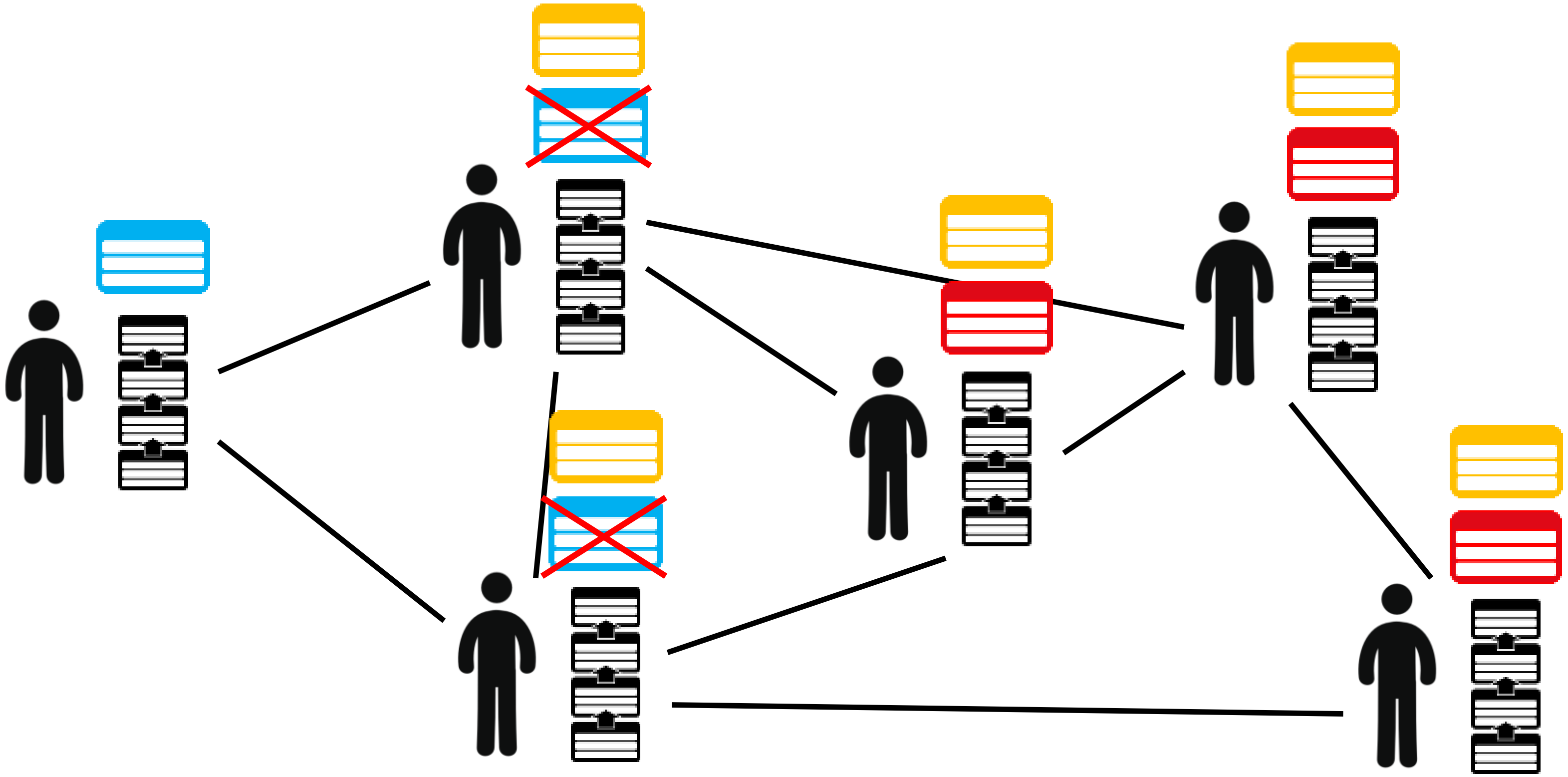


Solving a fork: The longest chain always wins!

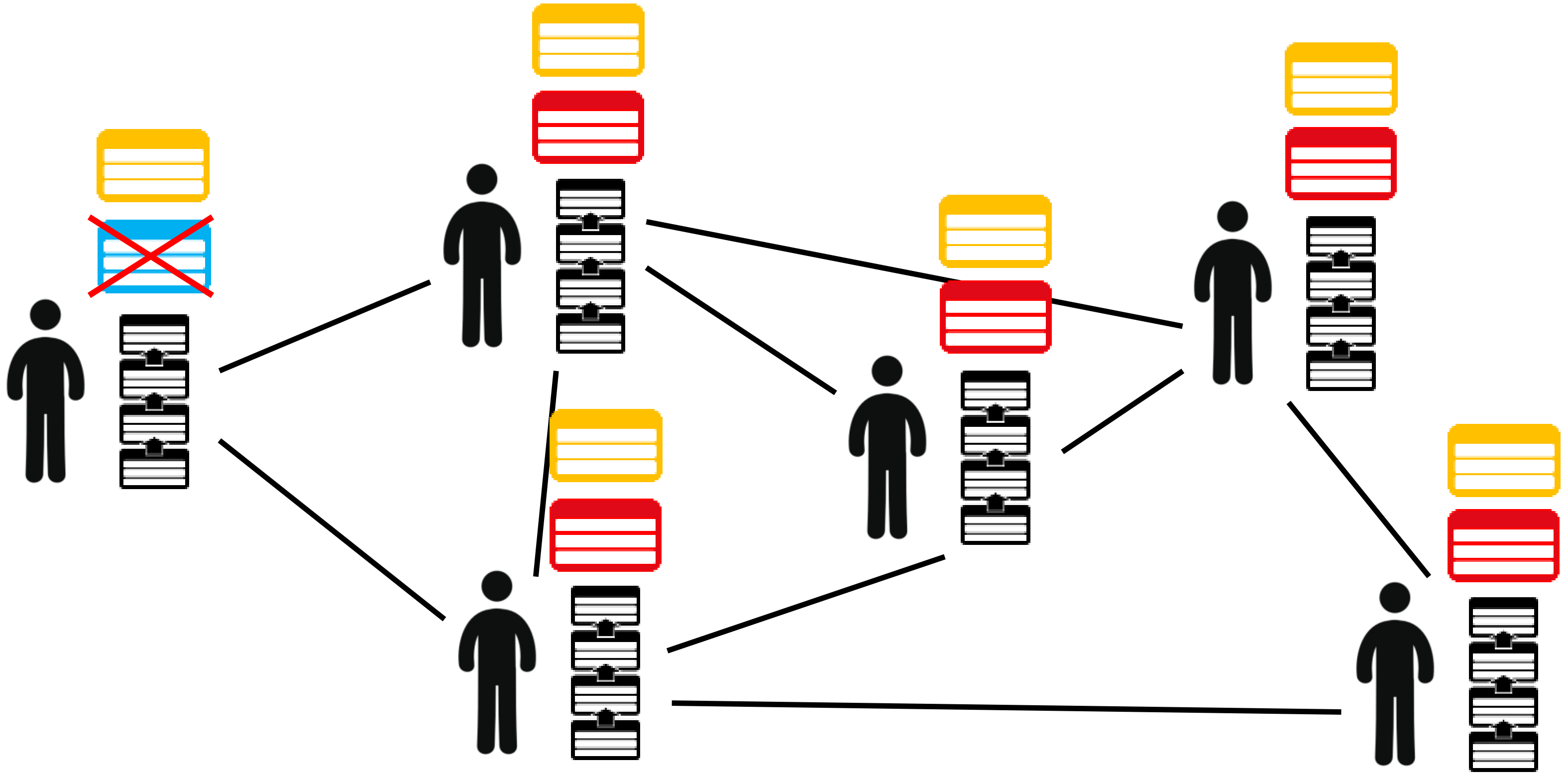
Solving the fork…



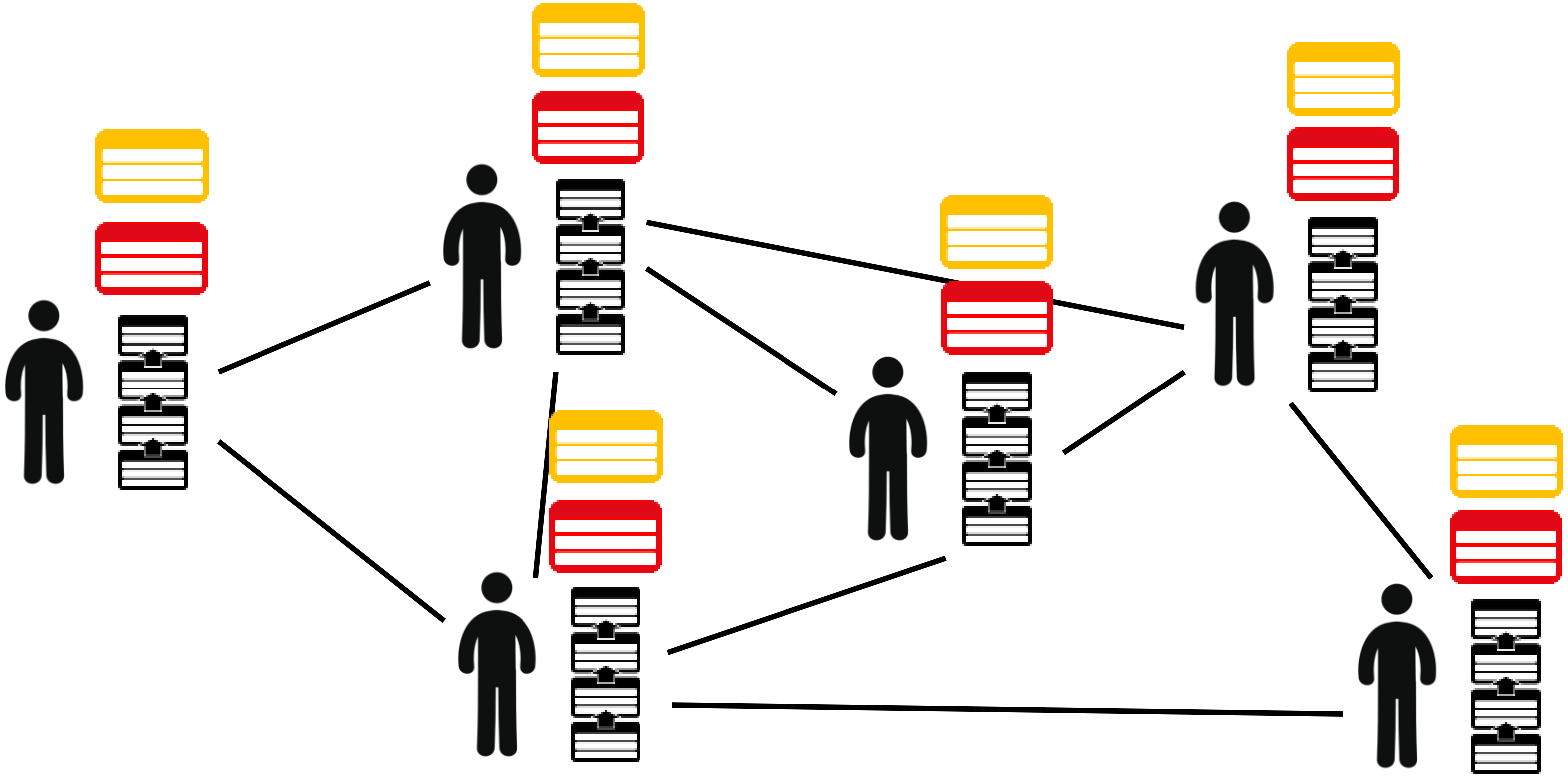
Solving the fork…



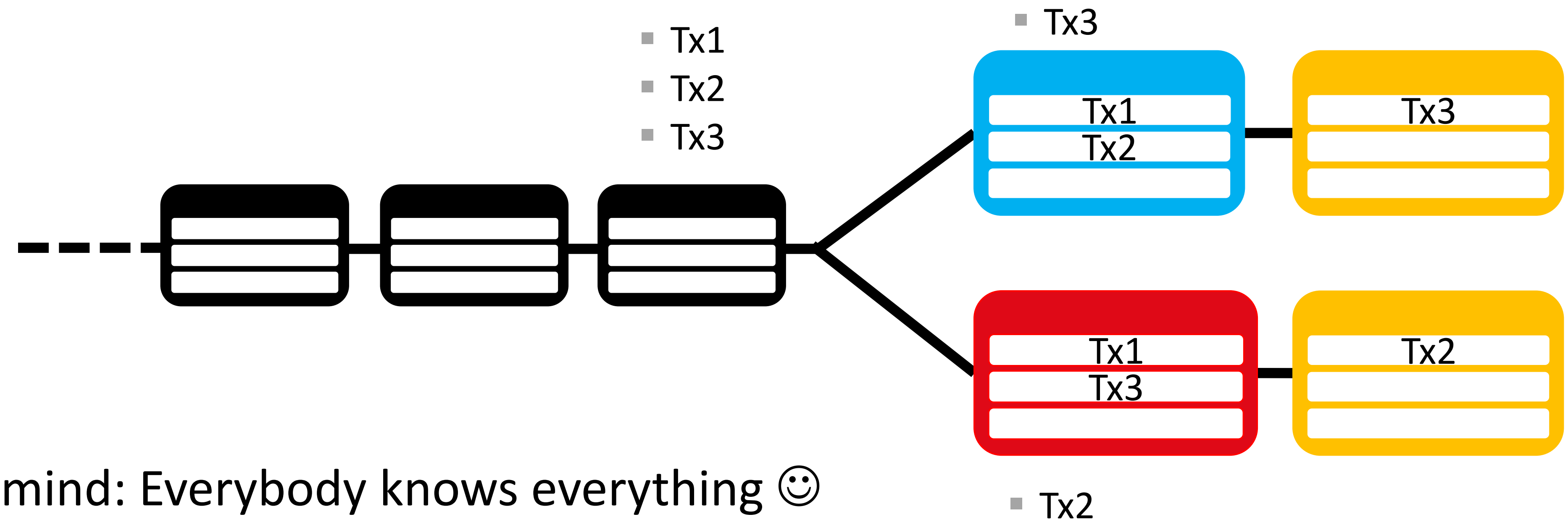
Solving the fork…



Solving the fork…



What about the transactions?

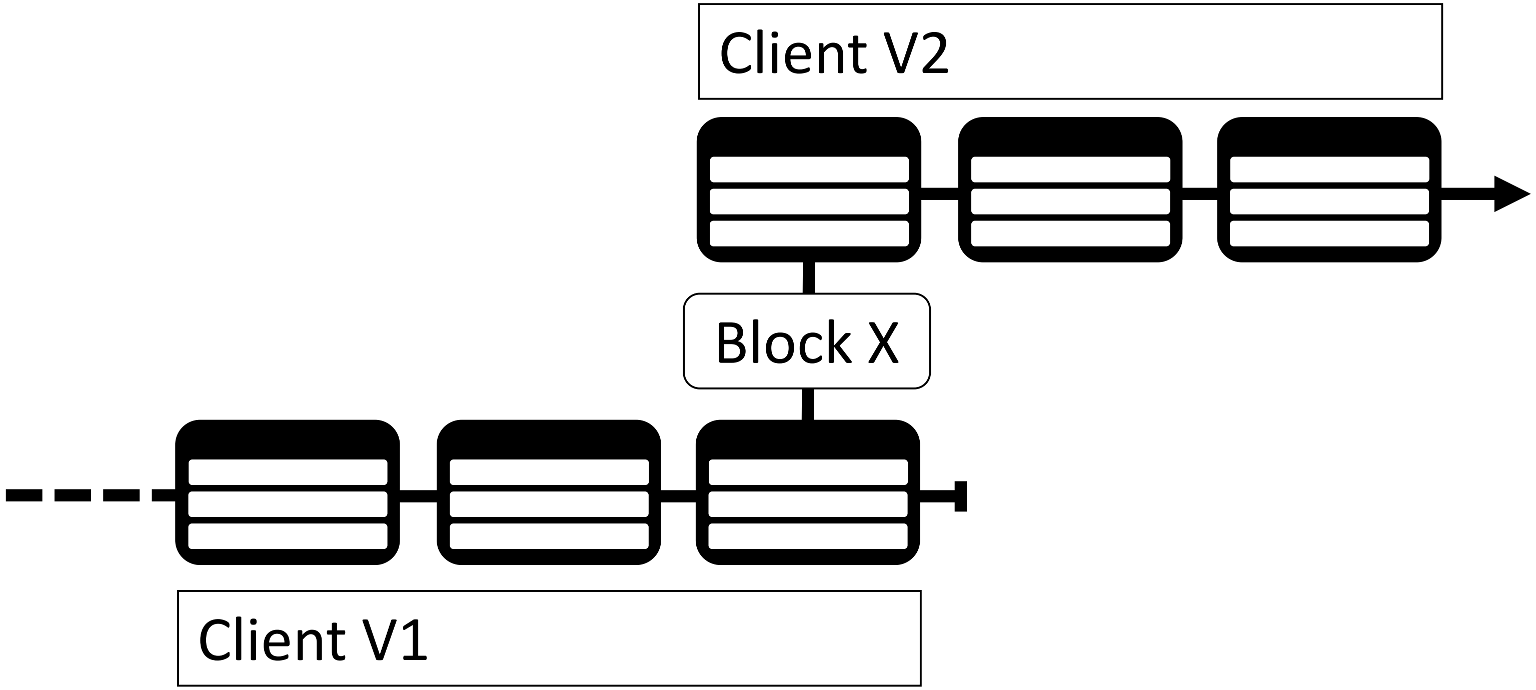


Keep in mind: Everybody knows everything ☺

It can happen that transactions are confirmed later.

It can happen, that transactions are considered invalid!

Hard Fork: Rolling out new client versions



Hard Fork → New incompatibel features (e.g. protocol changes).

Agreement on version → all miners switch at block X

Proof of…

Proof of Work

- Solve a “cryptographic riddle” brute-force
- Difficult to solve – easy to validate (you can imagine a Sudoku)
- Solving takes time, recalculation is virtually impossible

Proof of Stake

- Choose a “truth giver” according to his “stake”
  - e.g. amount of cryptocurrency
  - Democratic …?

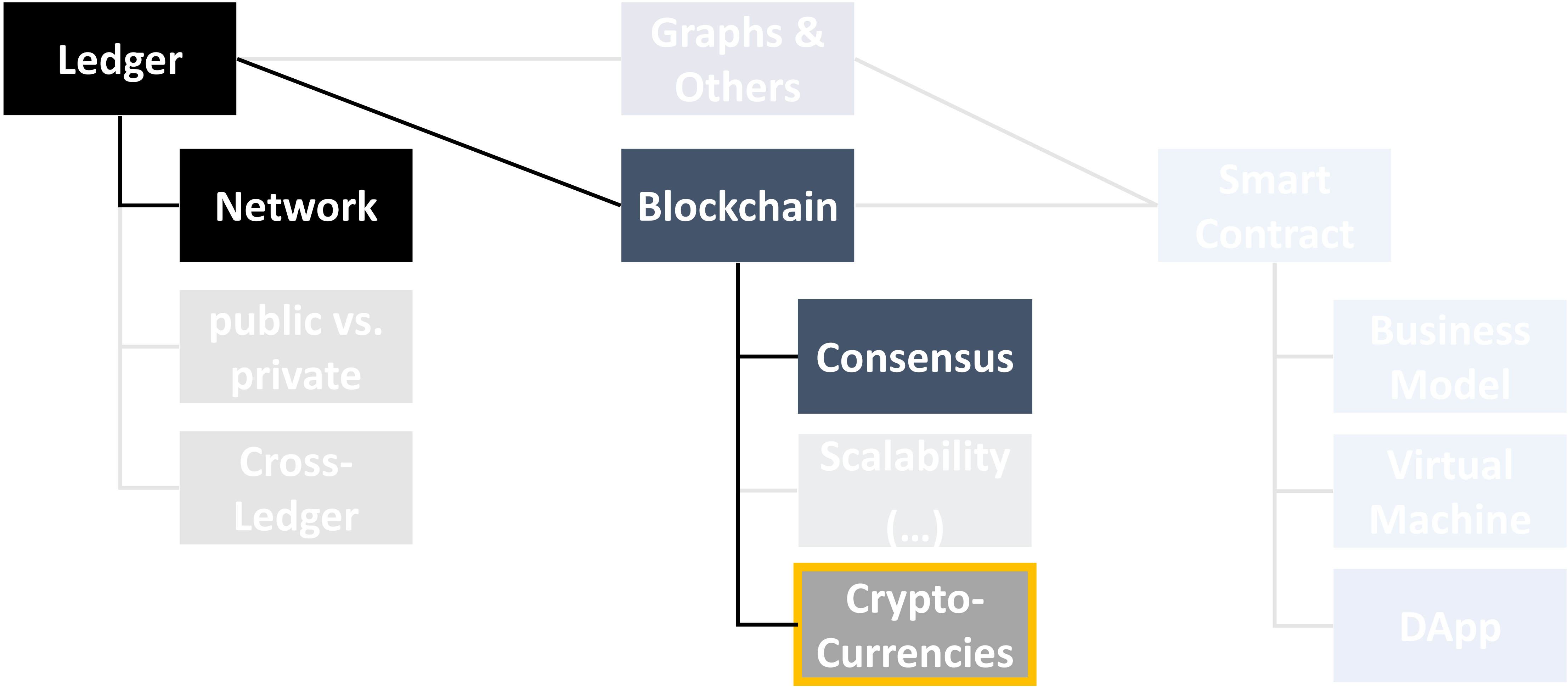
Proof of Elapse Time

- Proof through special hardware
- Certification process for hardware owners
- Some selection process

Proof of Authority

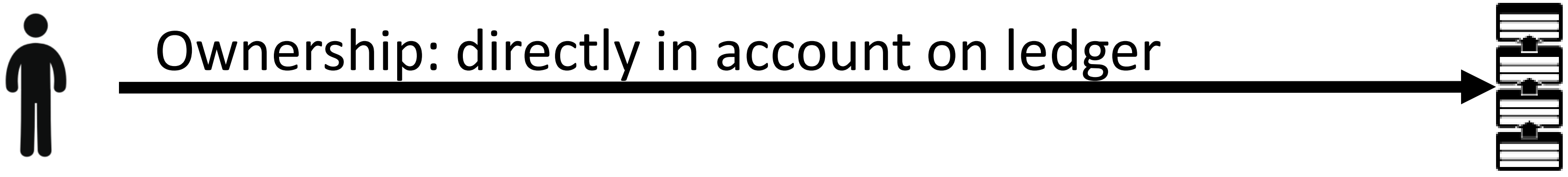
- Only certain nodes have assets
- They serve as “coin faucets”
- To get coins one has to reveal identity
- Used to secure test-networks

Let‘s focus on…



Coins vs. Tokens

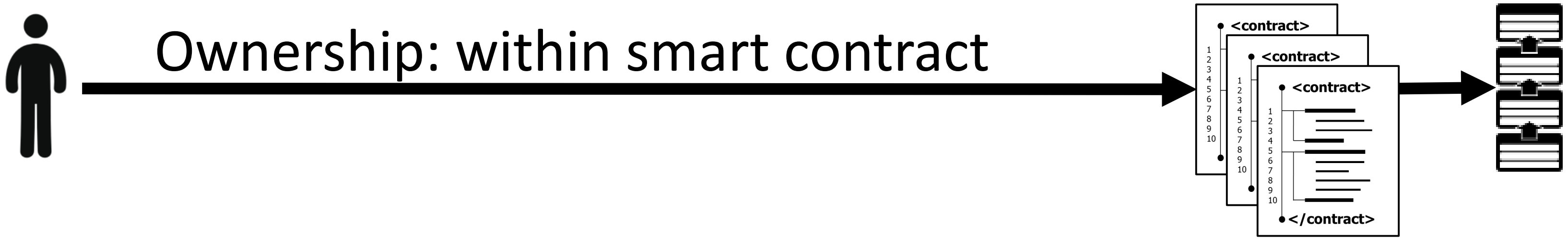
- **Coin:** Money Creation through consensus process (ledger as base)



Ownership: directly in account on ledger

Ledger
(Coin)

- **Token:** Money Creation through generation (smart contract as base)



Ownership: within smart contract

Ledger
(Coin)

- **ICO** (Initial Coin Offer) vs. **Token Sale**
  - Problem: Coins and Tokens are not distinguished clearly

Game Theory & Crypto Economics

Game Theory

- Game Theory Problems (Chicken, …)
- (Nash-)Equilibrium → Testing “experimental economics methods”
- Incentives

Crypto Economics

- The “nature of” P2P currency systems
- “Ledger parameters” → What can work?
- Security Models & Attack Scenarios
- Governance & Legal
- Government backed cryptocurrencies

Alt Coins and Side-Chains

Alt Coin

- Alternative Coin
- Basically everything after Bitcoin
- Opinionated term….

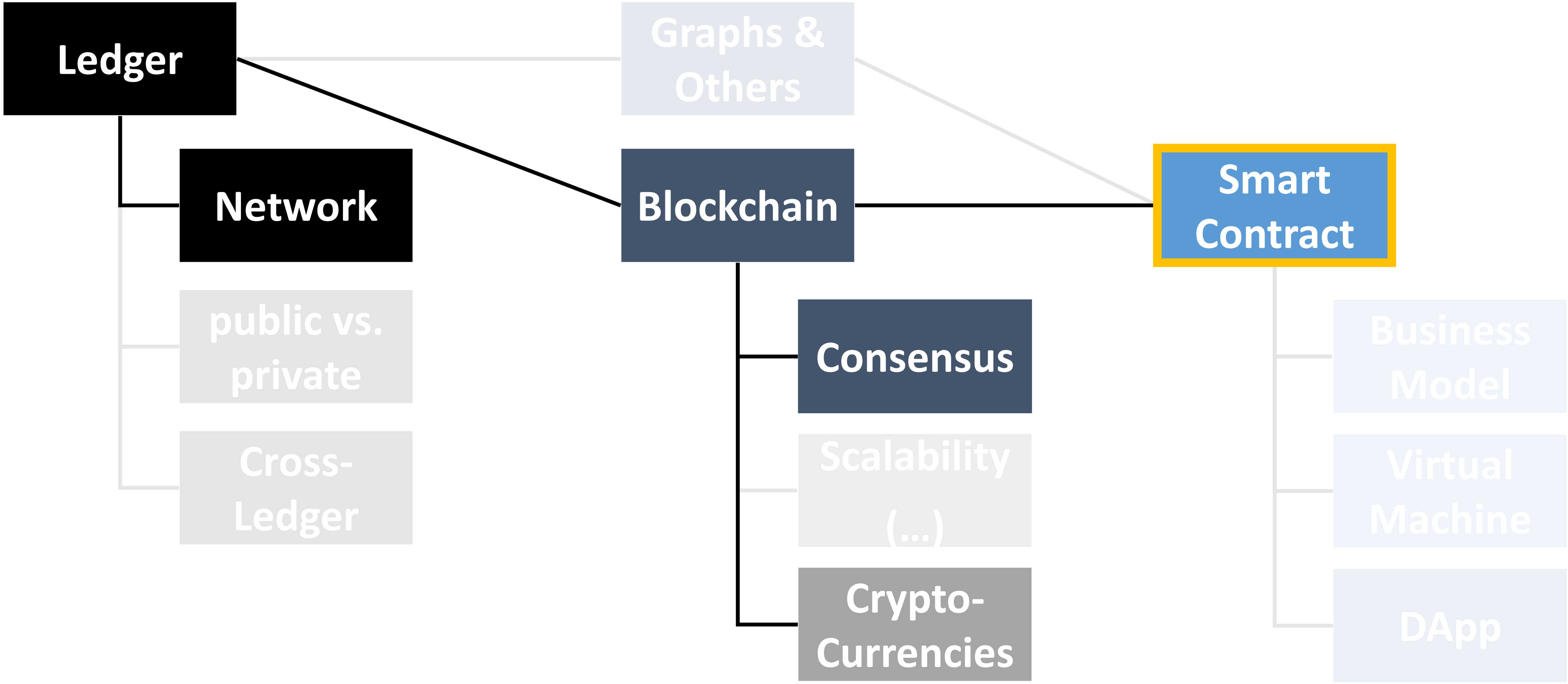
Side Chain

- Detached, independent ledger attached to another
- Asset transfer ~cross ledger through “lock-accounts”

Smart Contracts

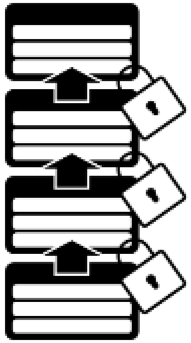
SC in Theory, SC development, SC deployment & communication, DApp

Let‘s focus on…



From Cryptocurrency to Smart Contract Platform

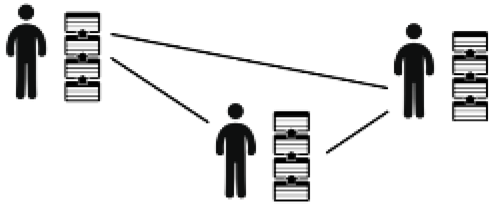
Cryptographic TRX list

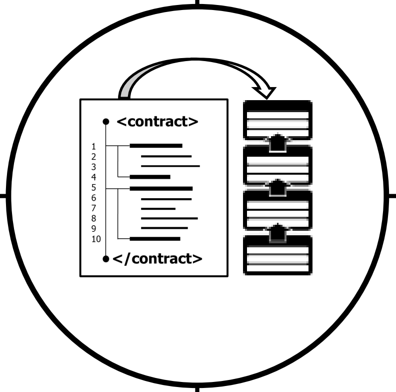


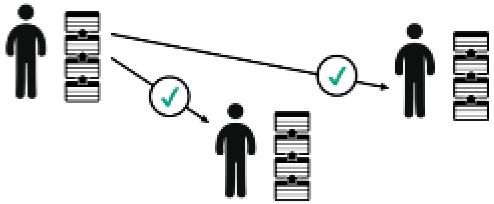
- Cryptographically secured ledger for the management of transactions and accounts

Peer-to-peer architecture

- Decentralized network of equal nodes







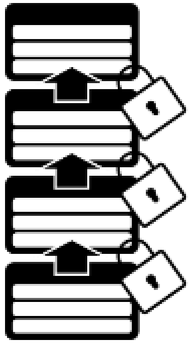
- Mechanism to agree on current state of system based on PO(…)

Consensus

?

From Cryptocurrency to Smart Contract Platform

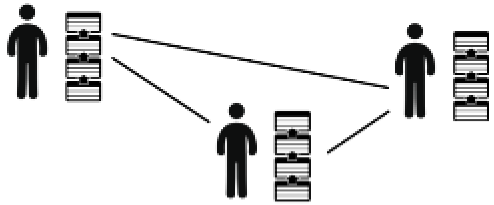
Cryptographic TRX list

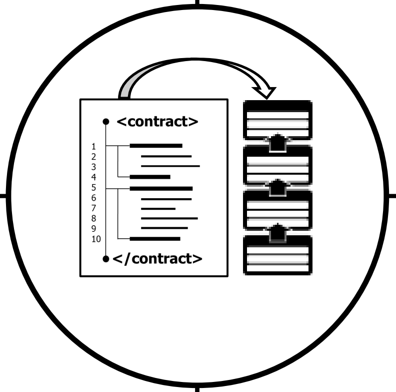


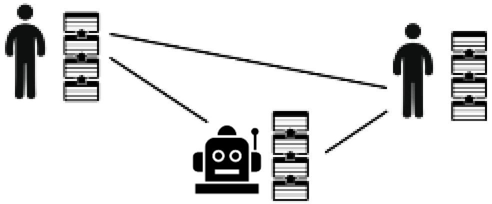
- Cryptographically secured ledger for the management of transactions and accounts

Peer-to-peer architecture

- Decentralized network of equal nodes

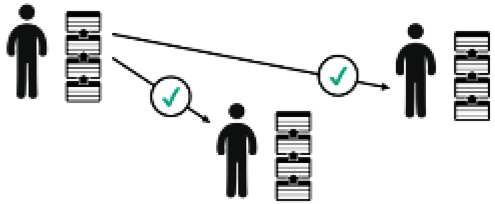






- Execute Smart Contract Bytecode
- Bytecode stored on Blockchain

Virtual Machine



- Mechanism to agree on current state of system based on PO(…)

Consensus

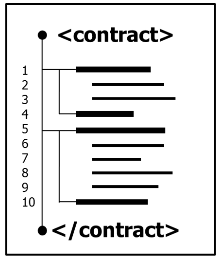
Smart Contracts in a Nutshell

“Transaction Service-Interface”

- Alter “data” on the “blockchain”
  State change through interface
- Interface: Methods & Parameters

Fairness and Transparency

- Contract Design → Fairness
- Bytecode openly available
- Every state change (data change) openly available



Contract Structure

- Definition of the contract
- Functionality of the contract
- Compare to: Class
- Bytecode on chain: Contract Creation
- No changes after creation

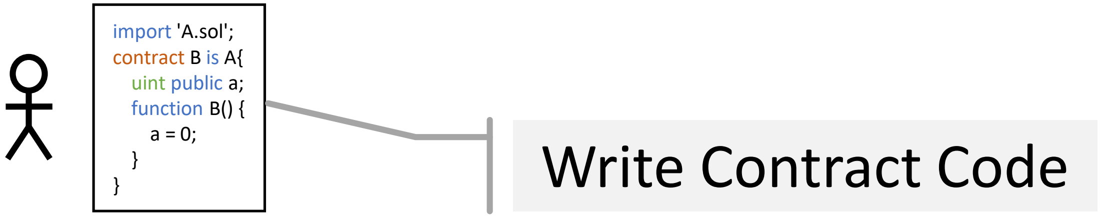
Contract State

- Alter variable values within the contract through transactions
- After contract creation: Send TX to method at contract address

Let‘s focus on Ethereum…

After all: It is one of (or the) most advanced smart contract platform out there.

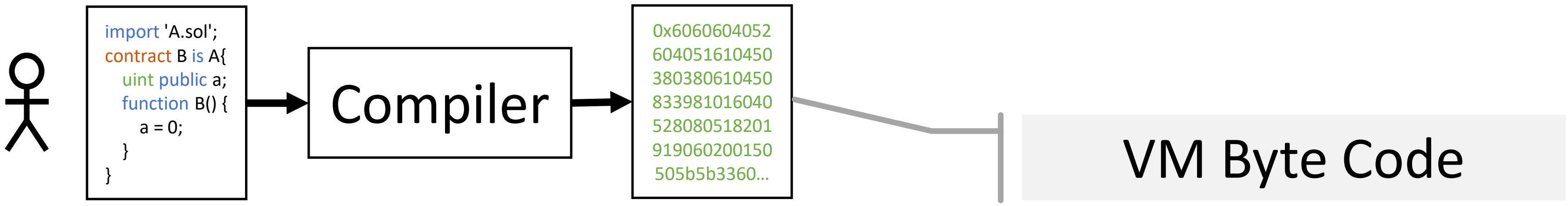
Contract Development



```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

Write Contract Code

Contract Development



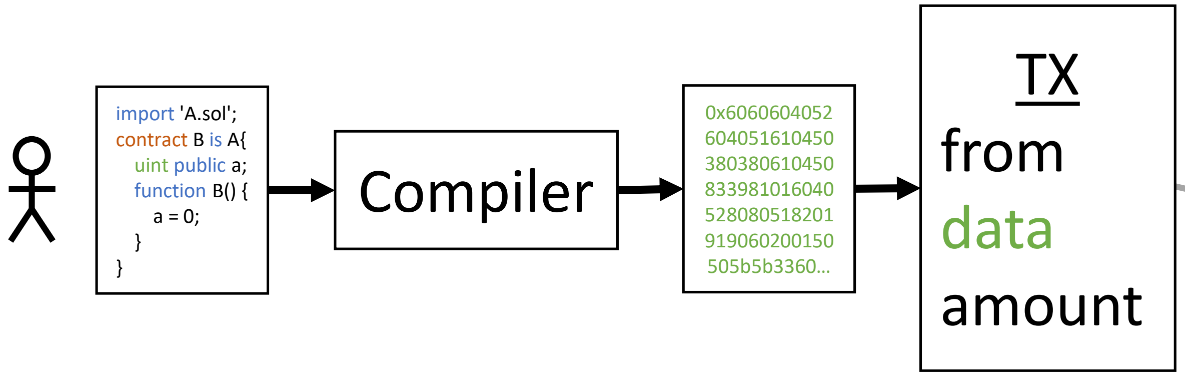
```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

```
0x6060604052
604051610450
380380610450
833981016040
528080518201
919060200150
505b5b3360...
```

VM Byte Code

- Languages that can be compiled to EVM bytecode:
  - **Solidity**
  - Serpent (not as much in use)
  - Viper (not finished?)
- Future: **eWASM** (Ethereum on WebAssembly) aka. the EVM 2.0 project

Contract Creation (Deployment)

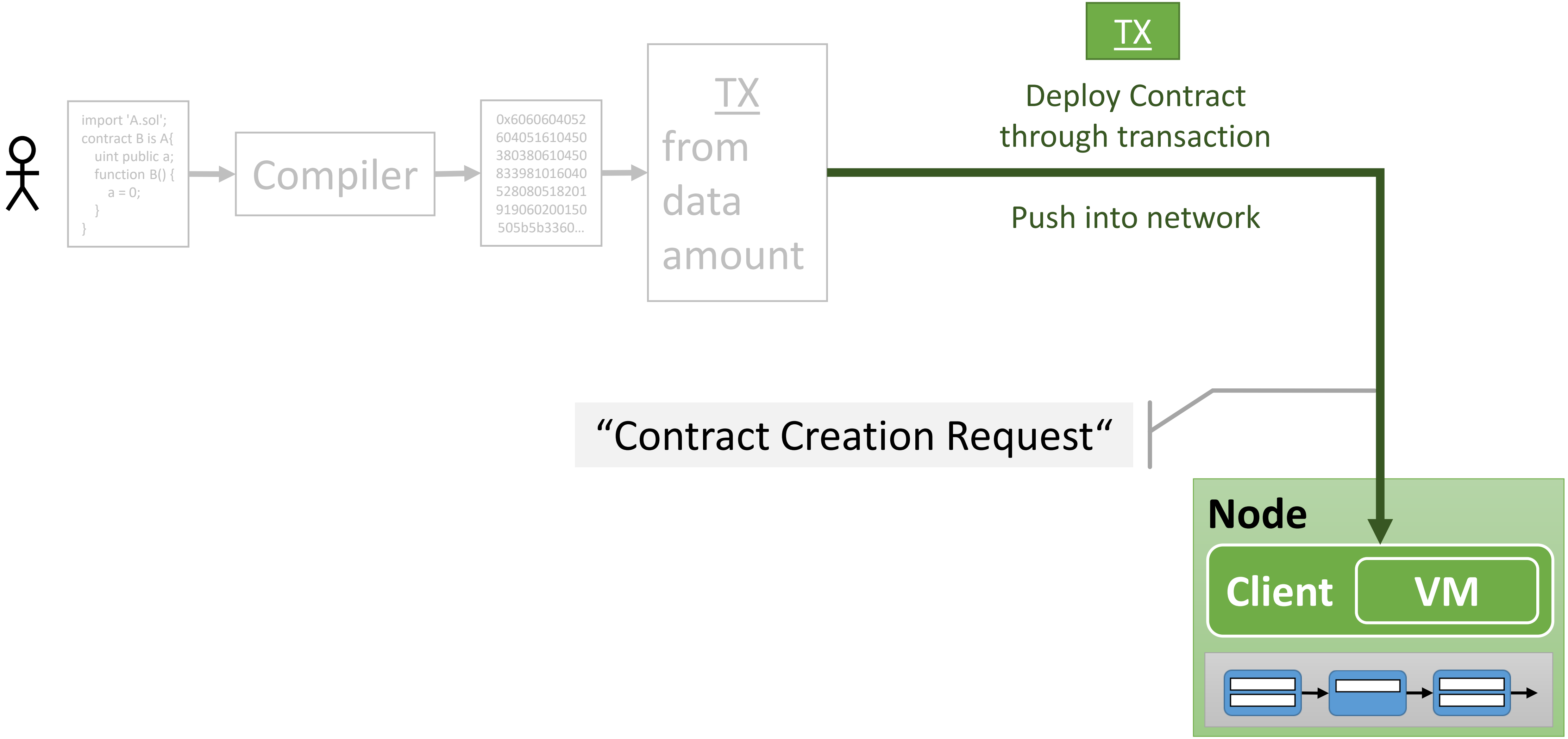


```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

0x6060604052
604051610450
380380610450
833981016040
528080518201
919060200150
505b5b3360...

Deploy Transaction

Contract Creation

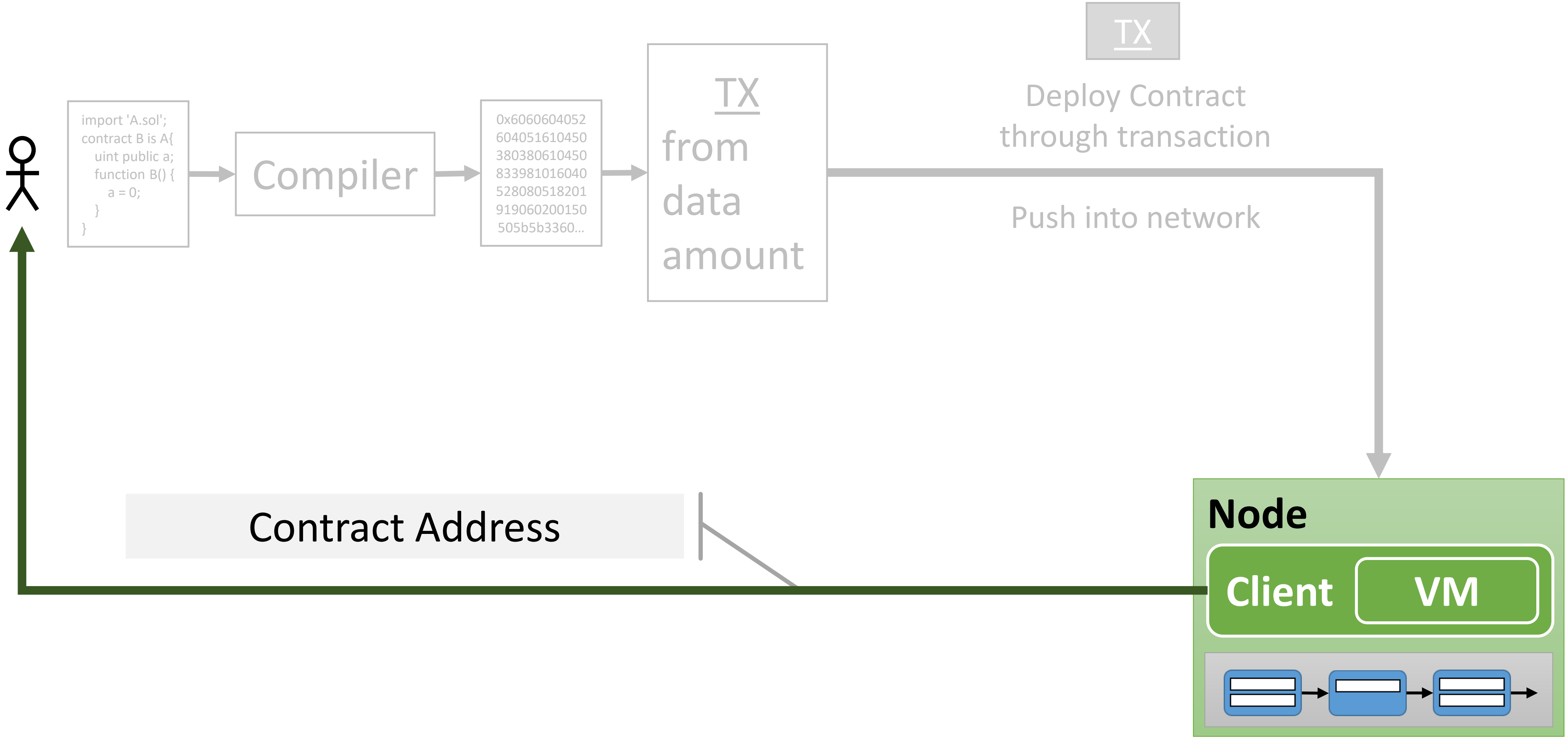


```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

```
0x6060604052
604051610450
380380610450
833981016040
528080518201
919060200150
505b5b3360...
```

“Contract Creation Request“

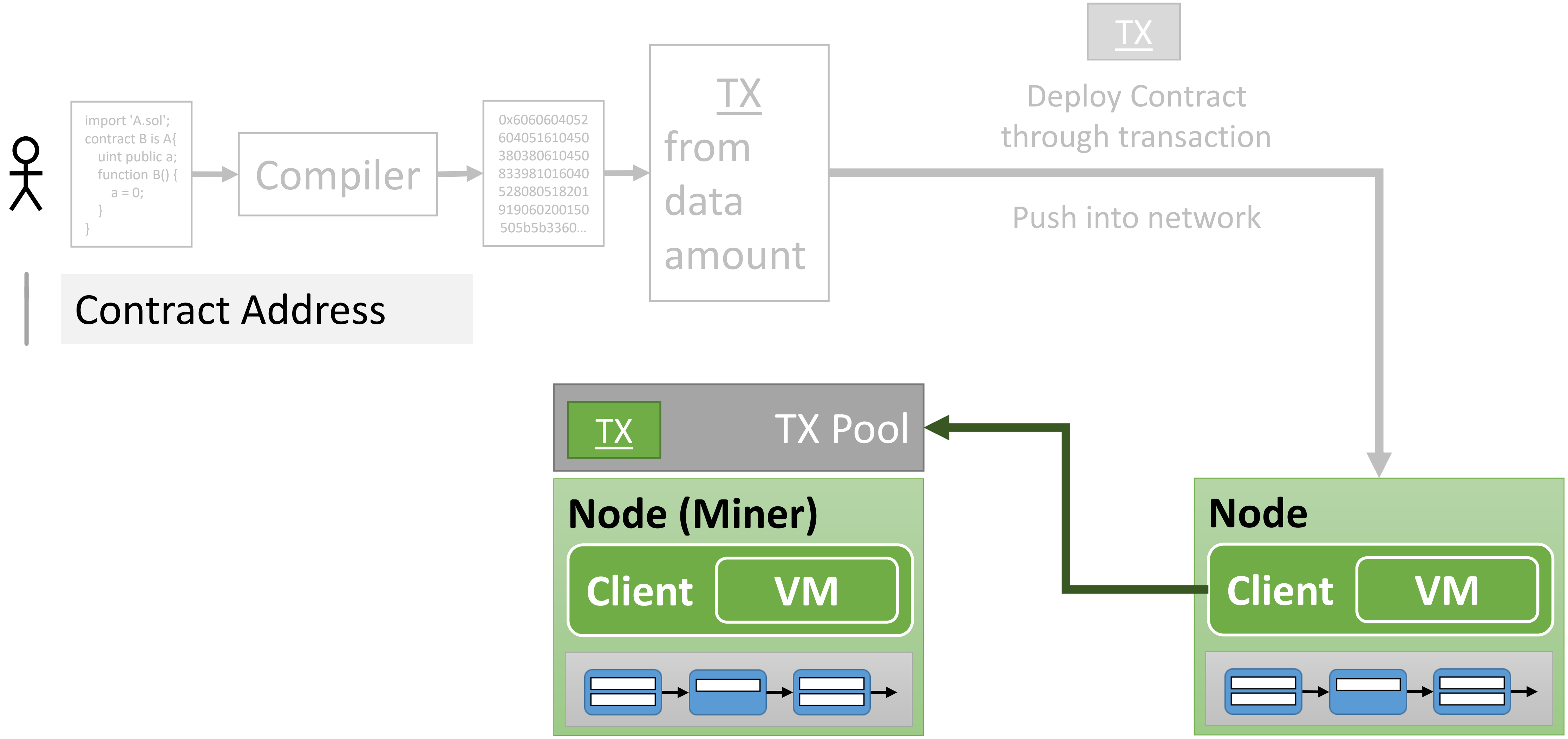
Contract Creation



```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

```
0x6060604052
604051610450
380380610450
833981016040
528080518201
919060200150
505b5b3360...
```

Contract Creation



```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

Compiler

```
0x6060604052
604051610450
380380610450
833981016040
528080518201
919060200150
505b5b3360...
```

TX
from
data
amount

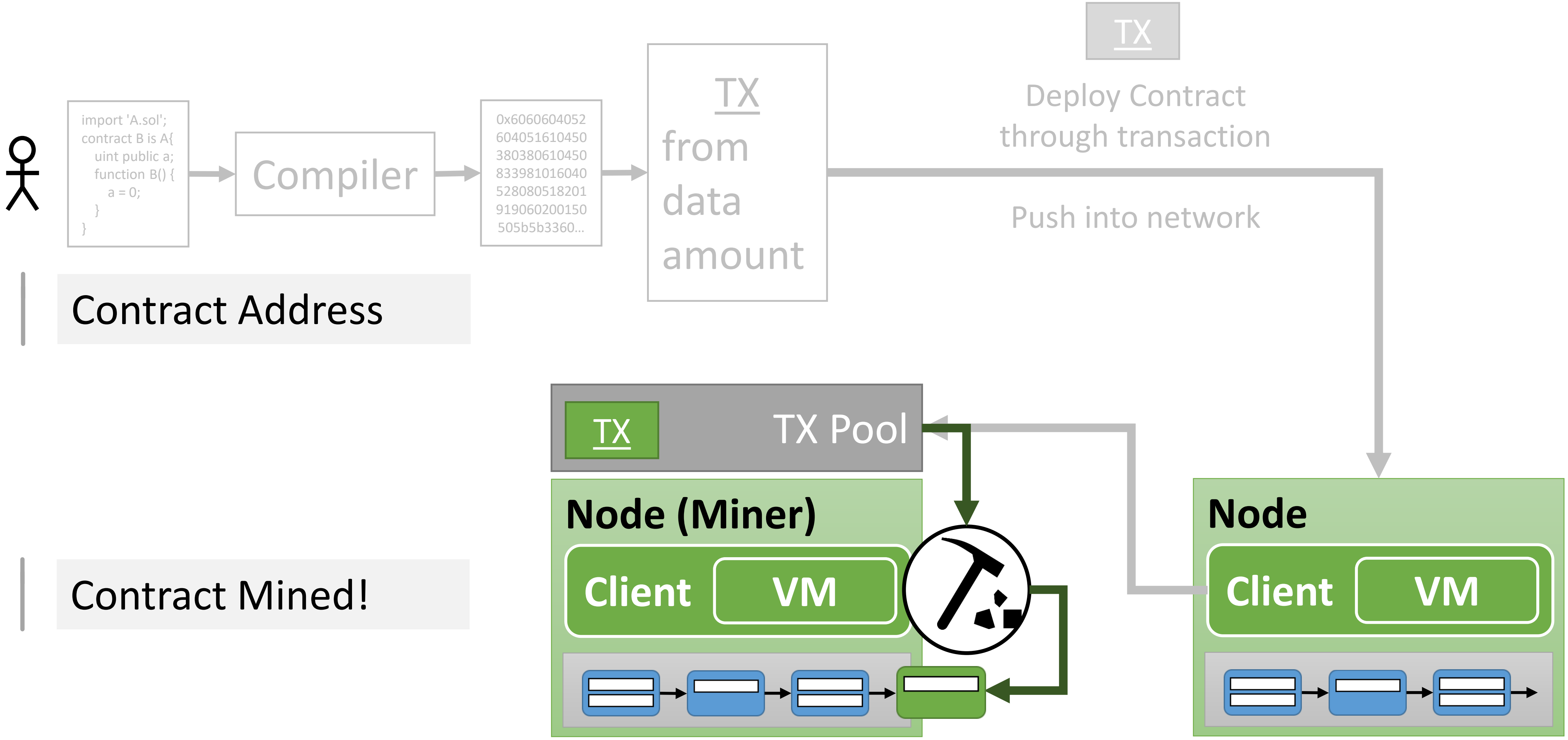
TX

Deploy Contract through transaction

Push into network

Contract Address

Contract Creation

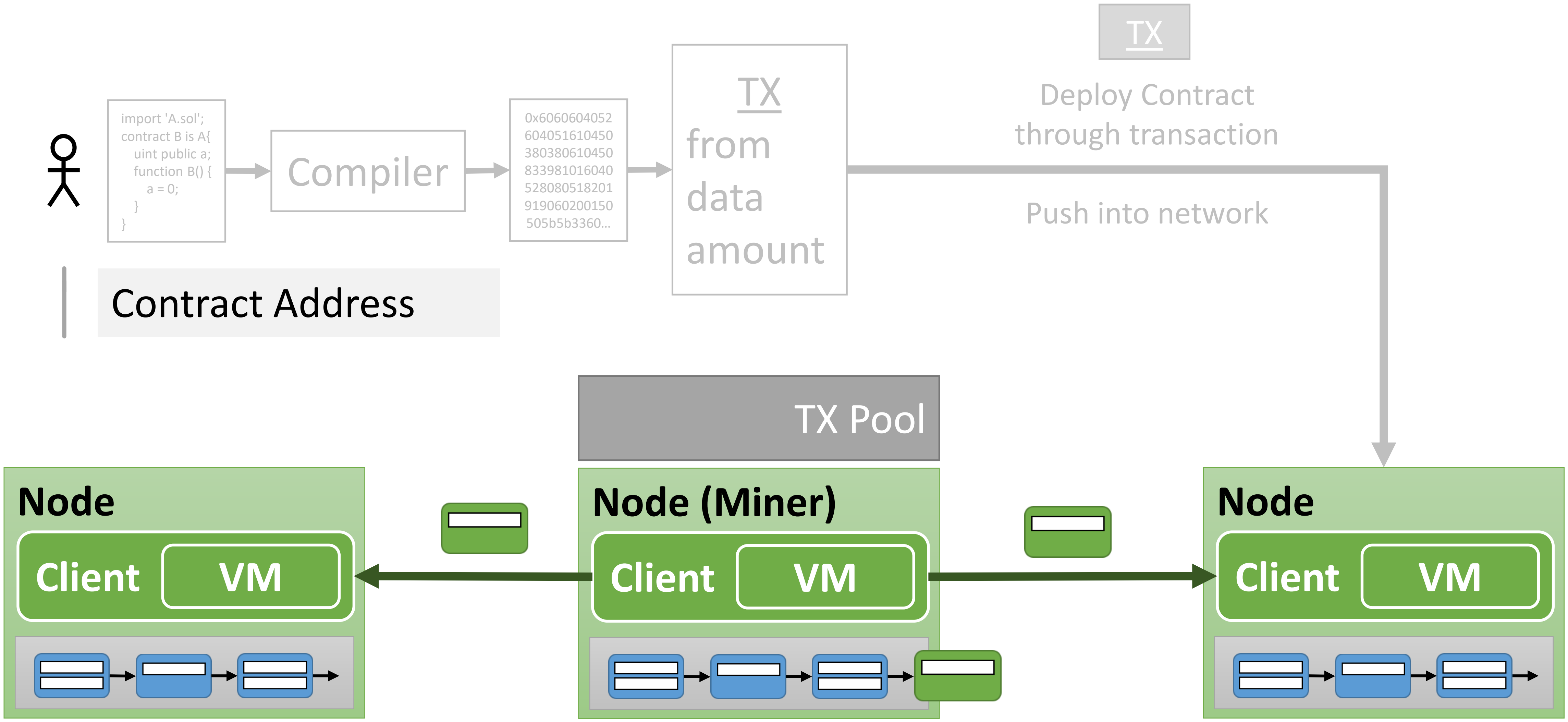


```
import 'A.sol';
contract B is A{
    uint public a;
    function B() {
        a = 0;
    }
}
```

Contract Address

Contract Mined!

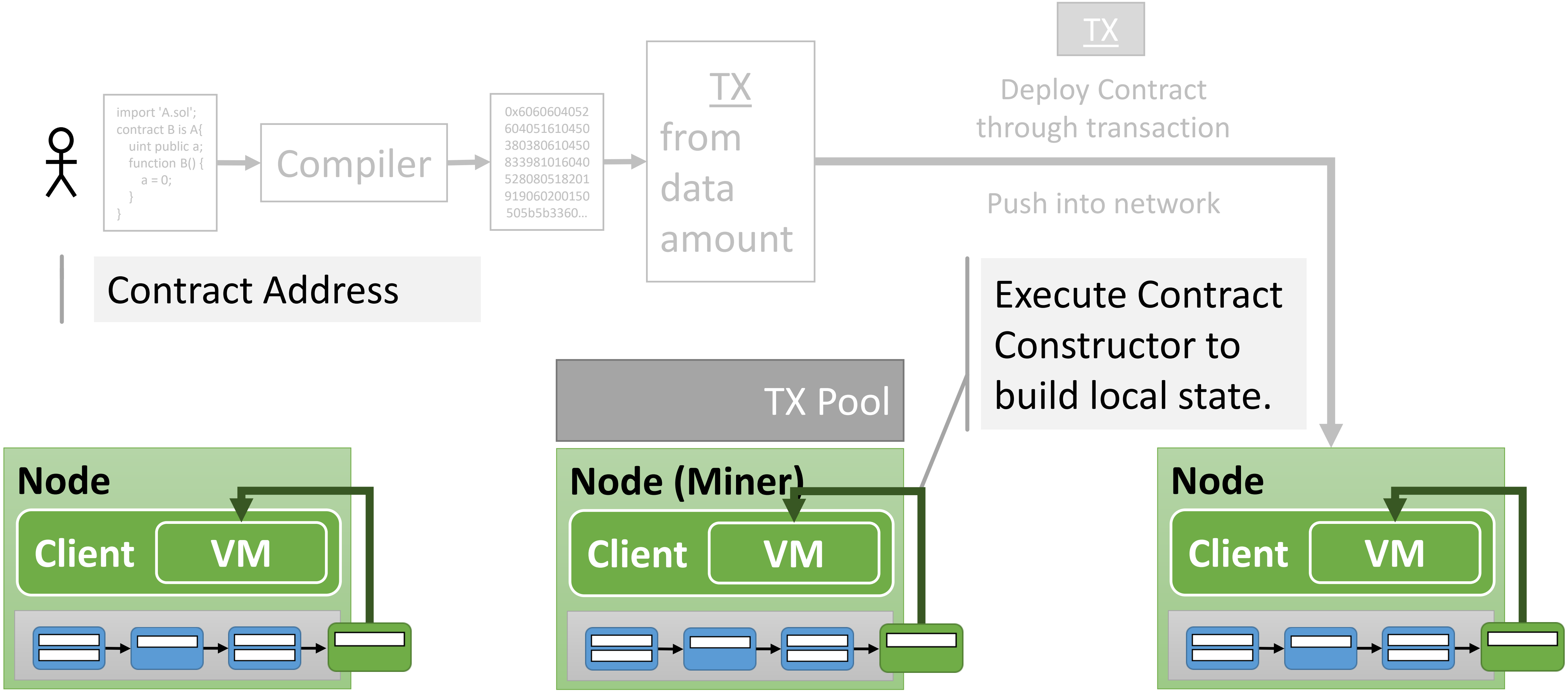
Contract Creation



```
import 'A.sol';
contract B is A{
   uint public a;
   function B() {
      a = 0;
   }
}
```

```
0x6060604052
604051610450
380380610450
833981016040
528080518201
919060200150
505b5b3360...
```

Contract Creation



```
import 'A.sol';
contract B is A{
    uint public a;
    function B() {
        a = 0;
    }
}
```

Contract Address

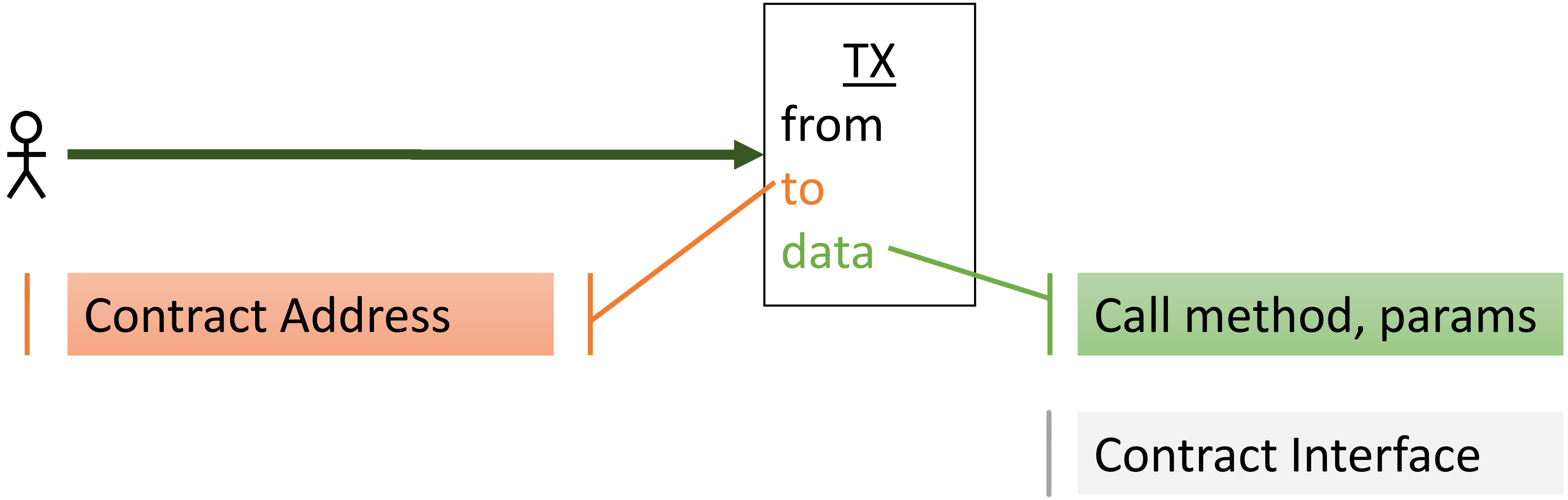
Execute Contract Constructor to build local state.

Contract Communication

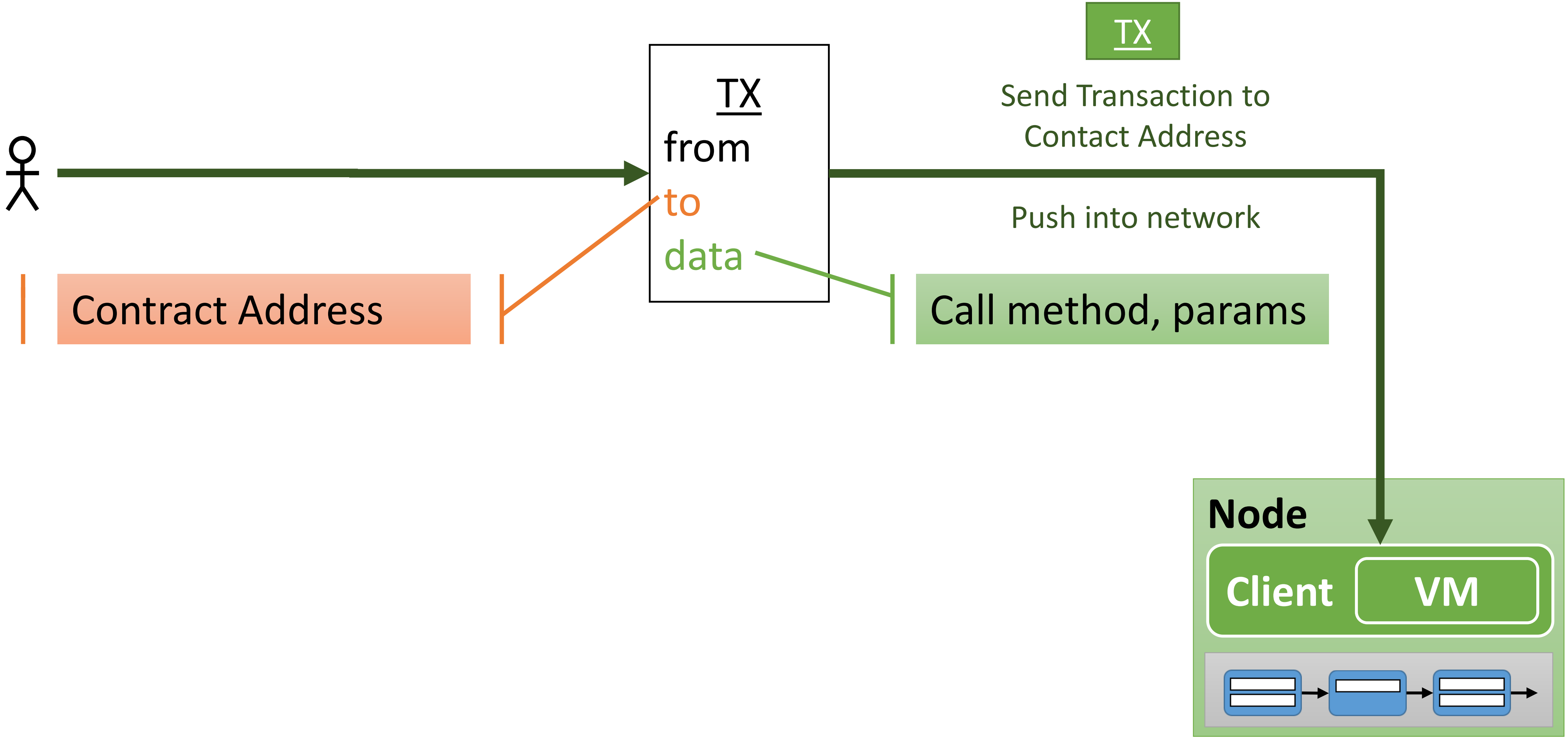
Contract Address

Contract Interface

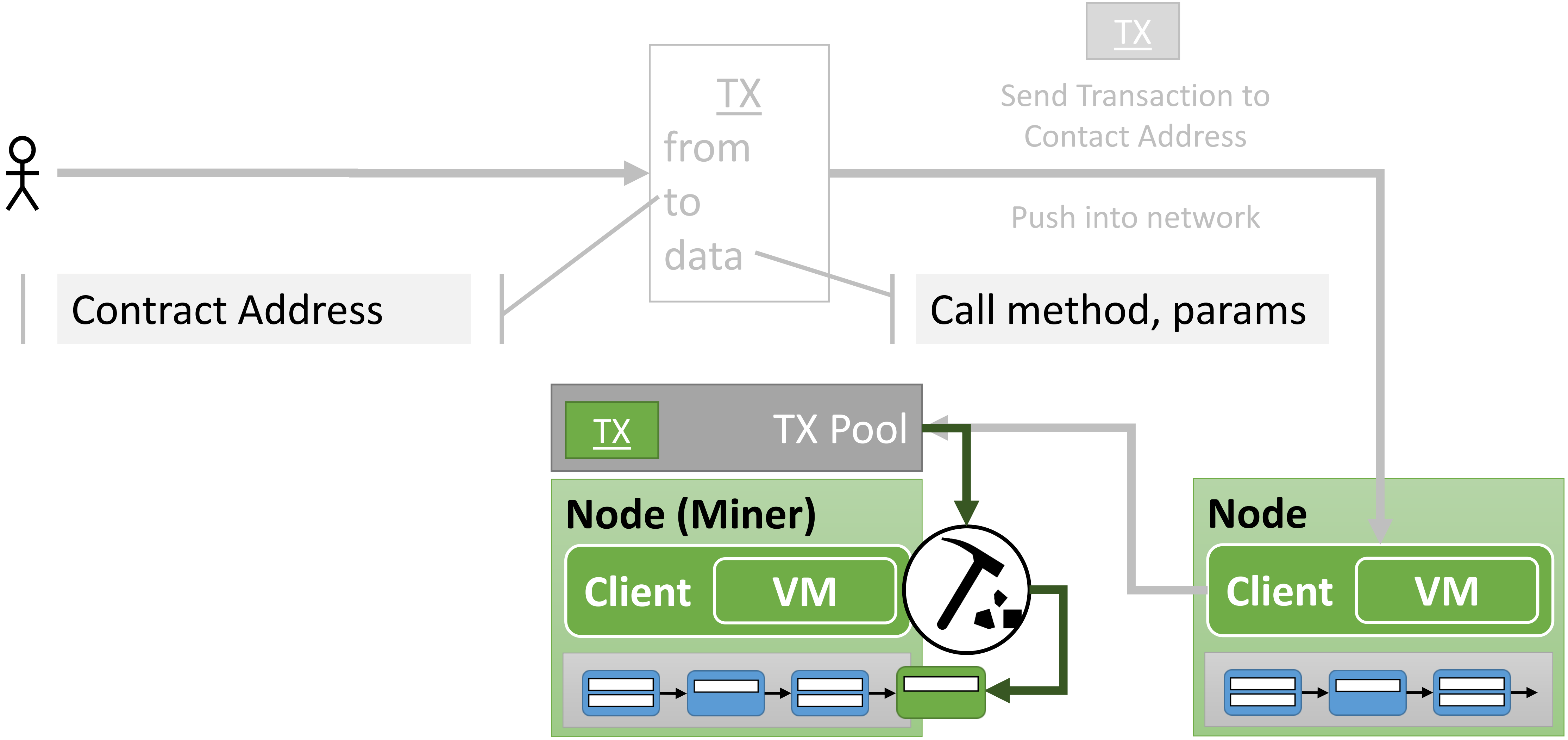
Contract Communication



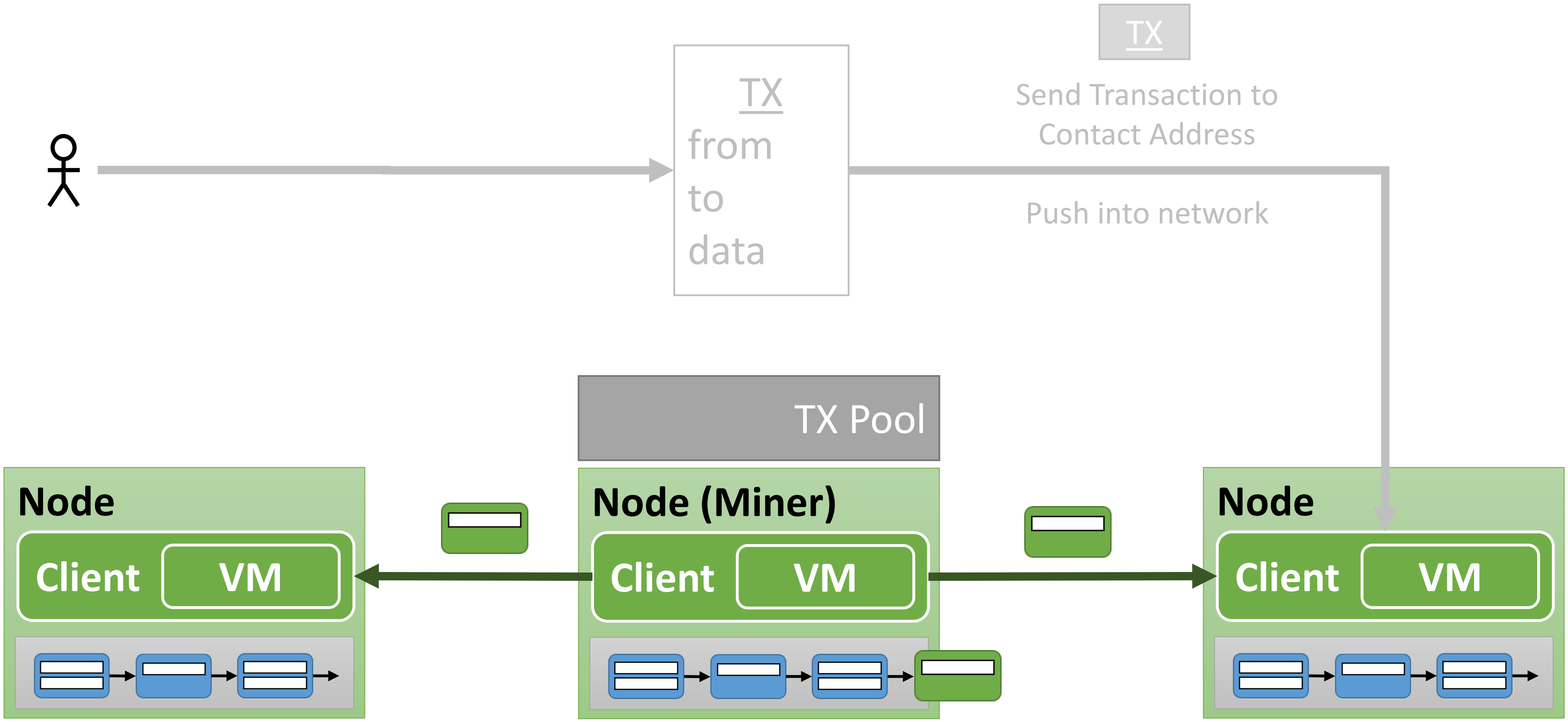
Contract Communication



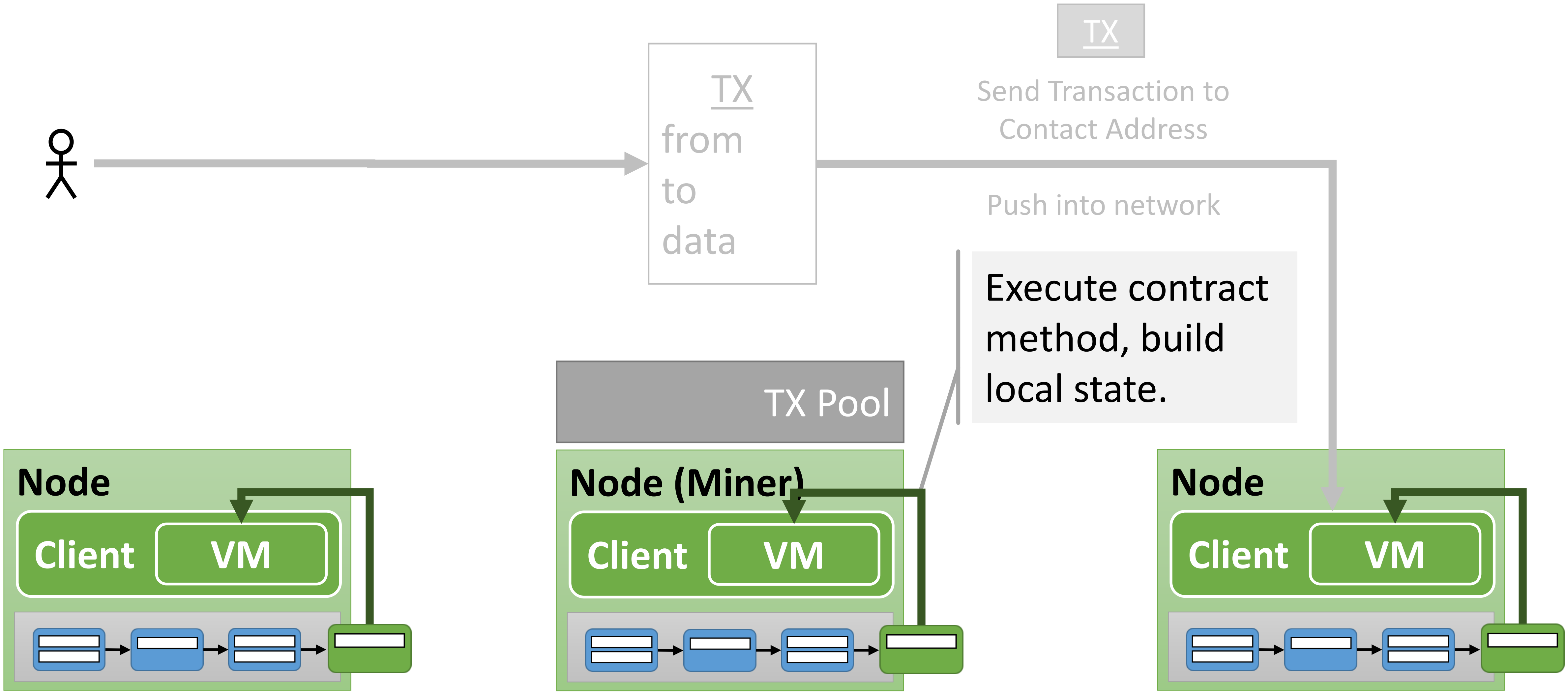
Contract Communication



Contract Communication

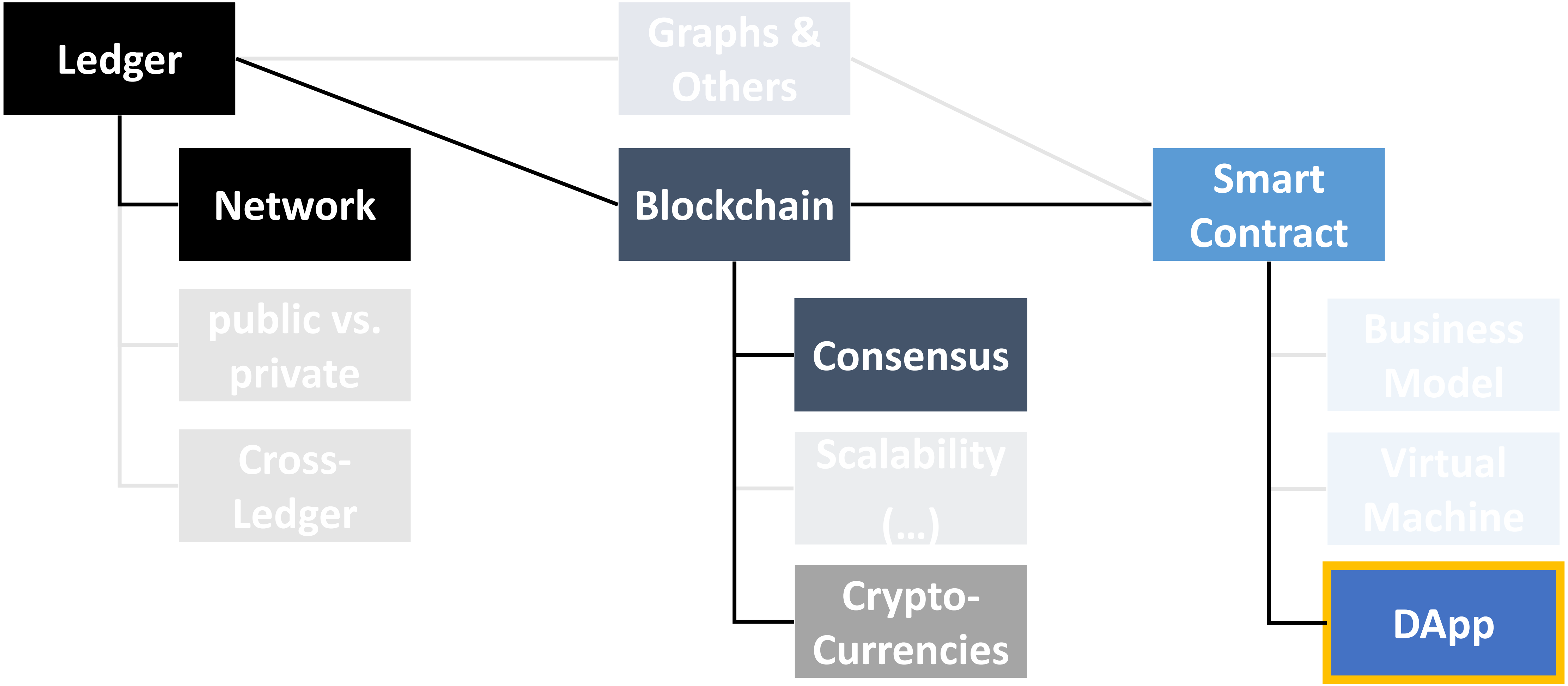


Contract Communication



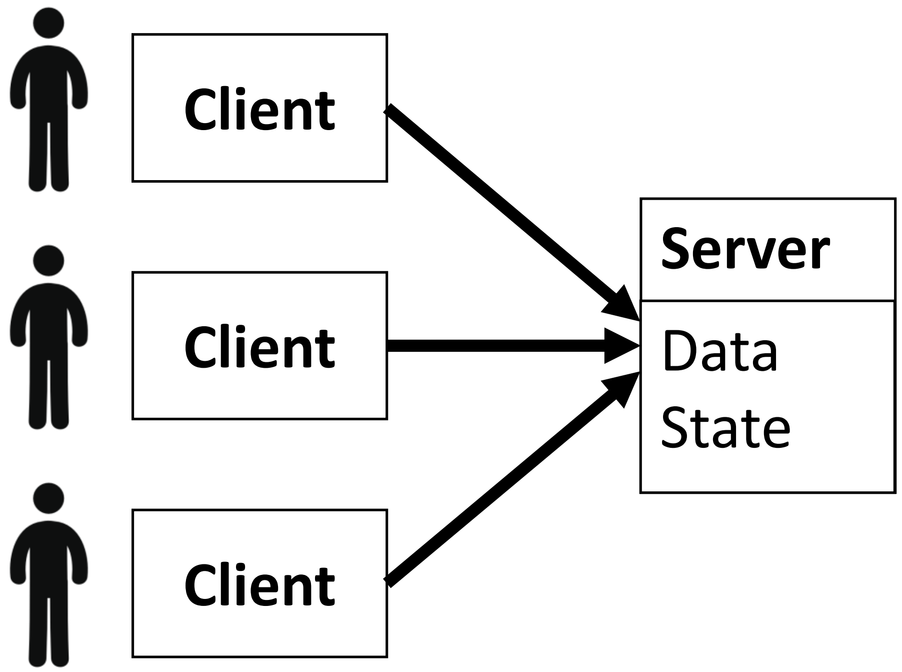
Execute contract method, build local state.

Let‘s focus on…



From Central to Decentral

Central



Client

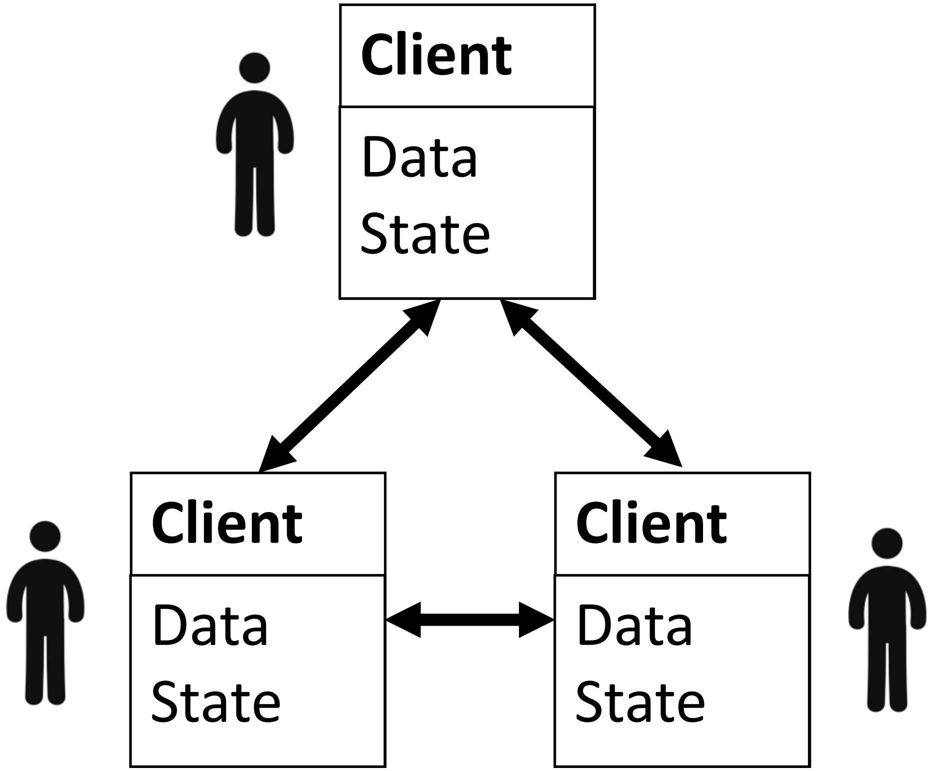
Client

Client

Server

Data
State

Decentral



Client

Data
State

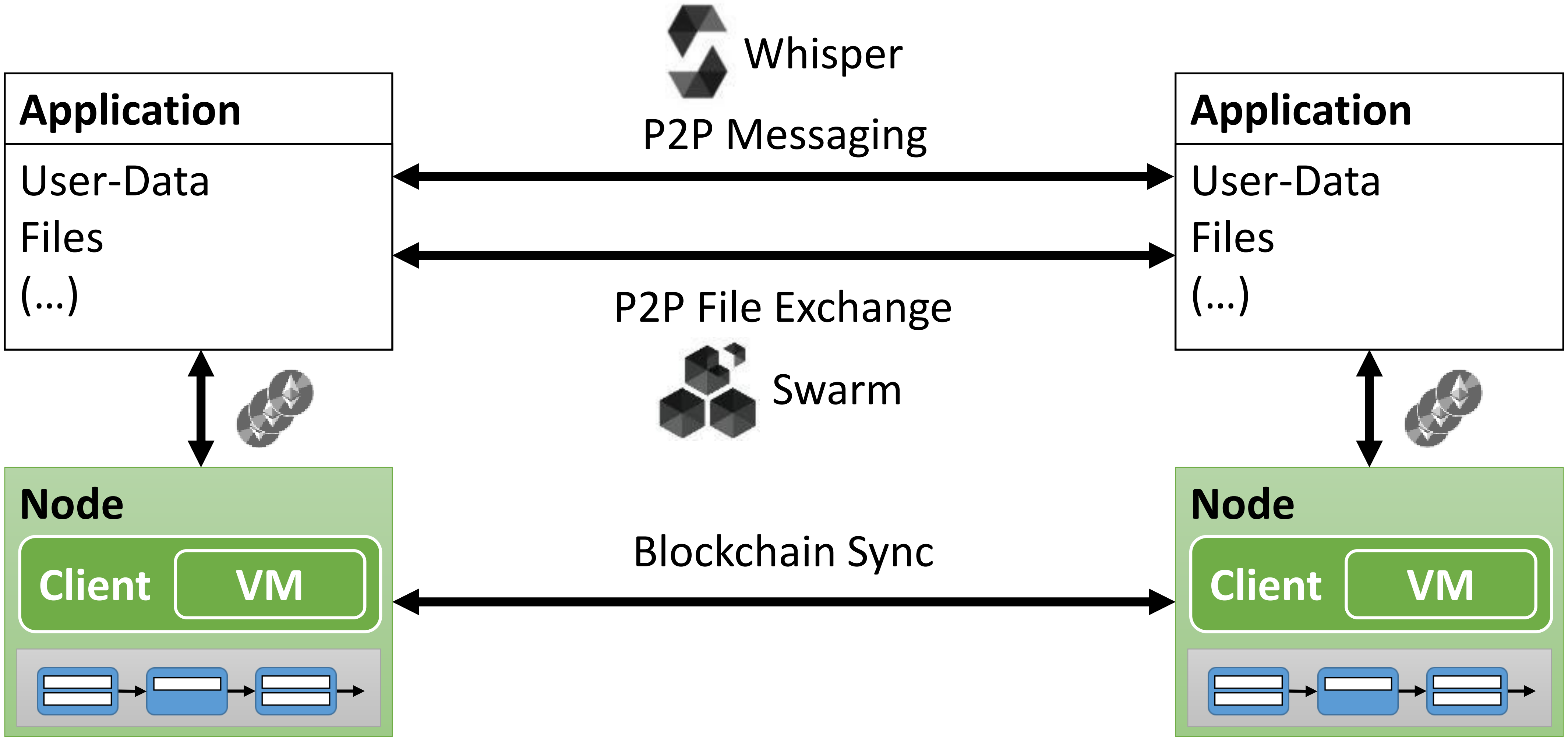
Client

Data
State

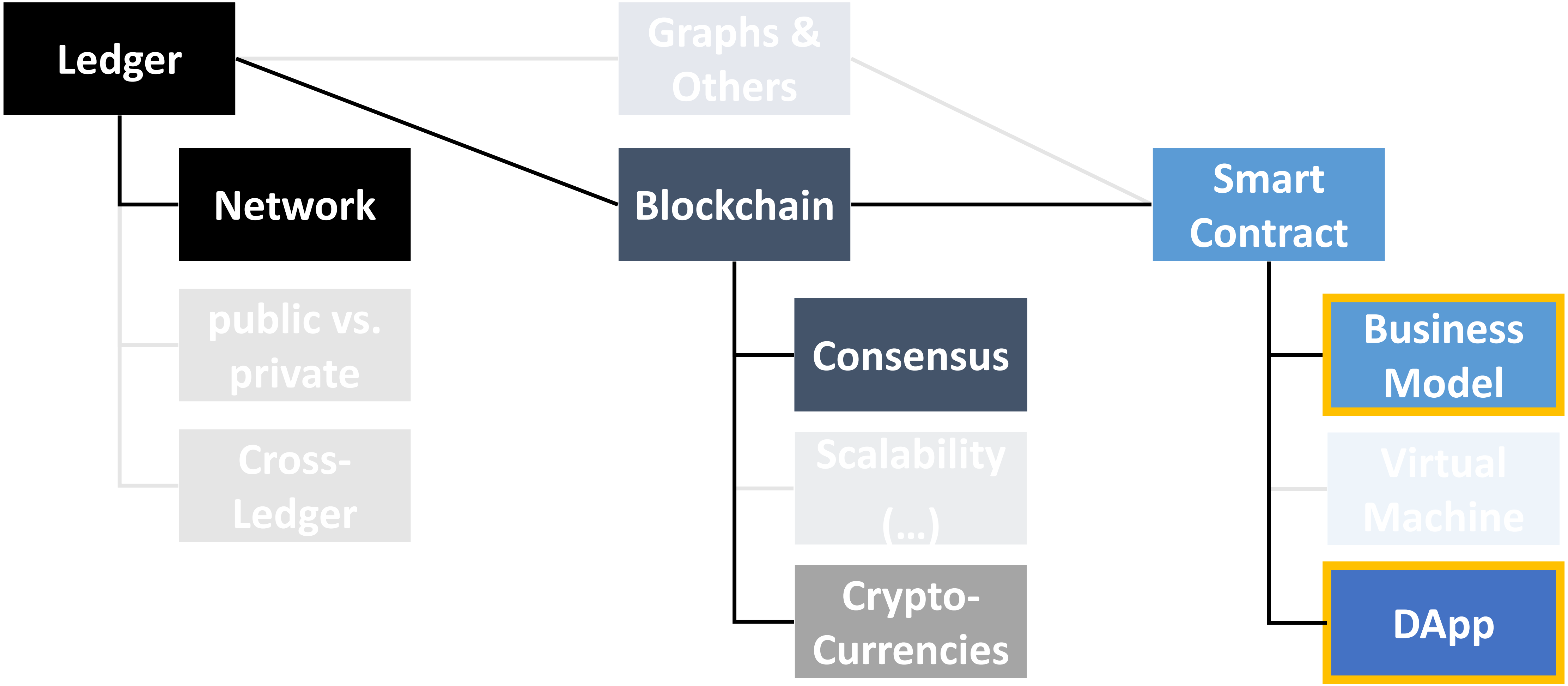
Client

Data
State

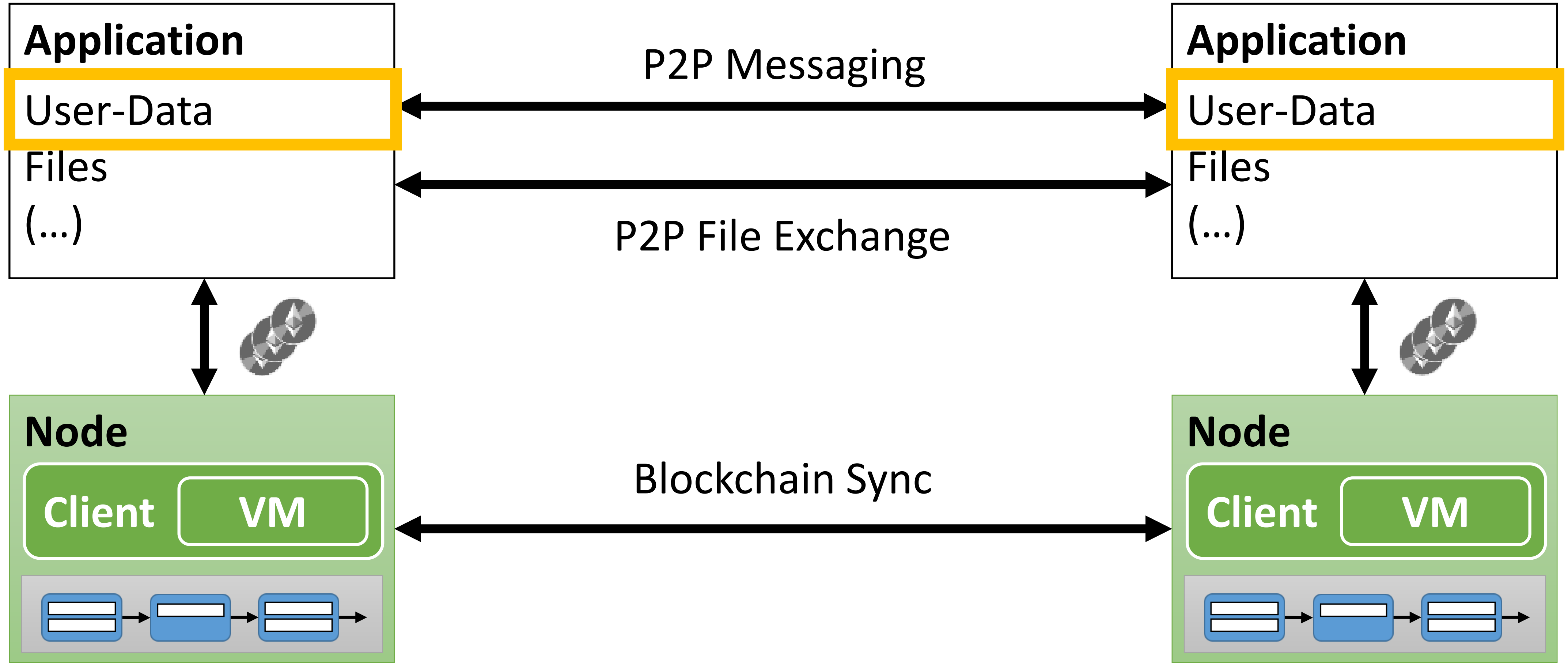
What is a real decentral application?



Let‘s focus on…



Business model change



Application
User-Data
Files
(...)

P2P Messaging

P2P File Exchange

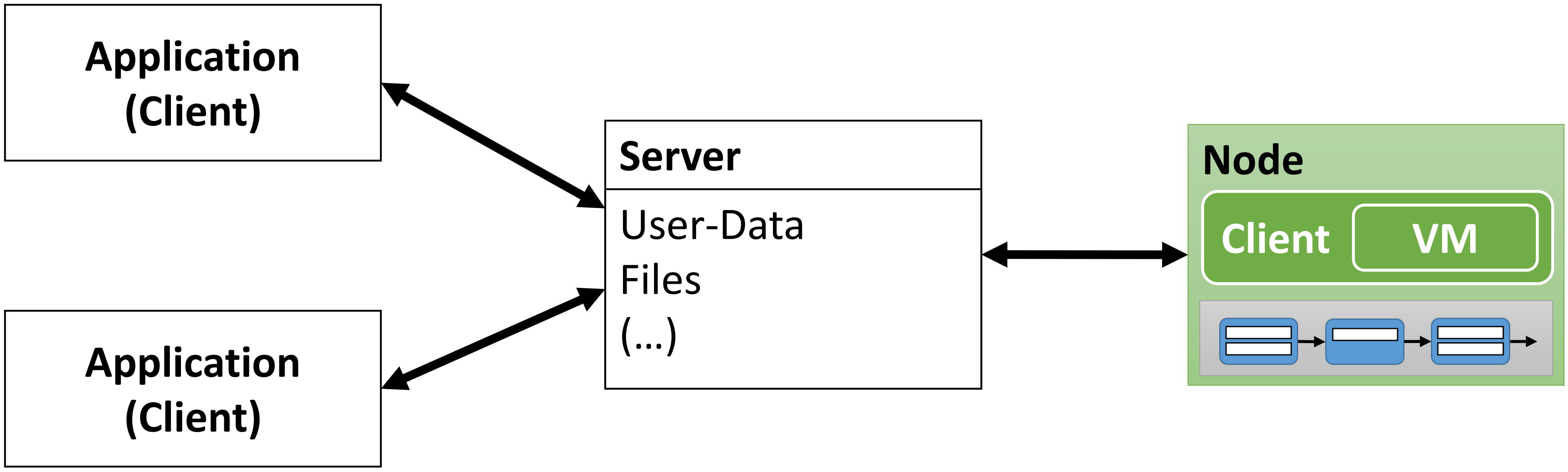
Application
User-Data
Files
(...)

Node
Client
VM

Blockchain Sync

Node
Client
VM

What people often build today…



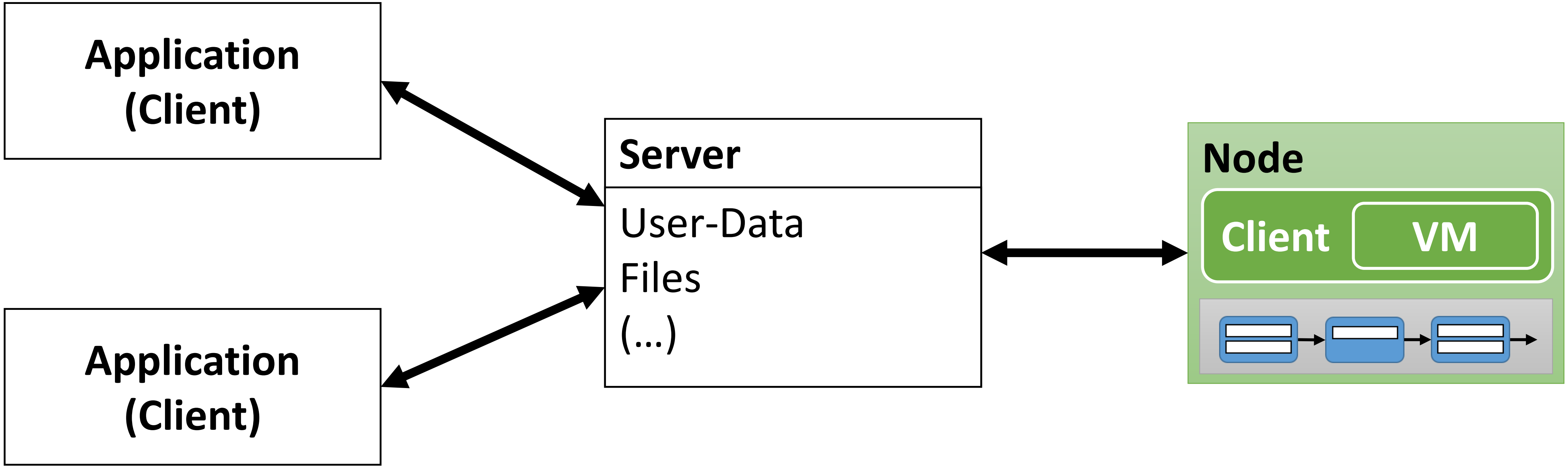
This is not necessarily wrong…

For e.g. certification & transparency it is OK, but it is not the „big revolution“

Where do you keep the account?

Who manages the account – and what implications does that have?

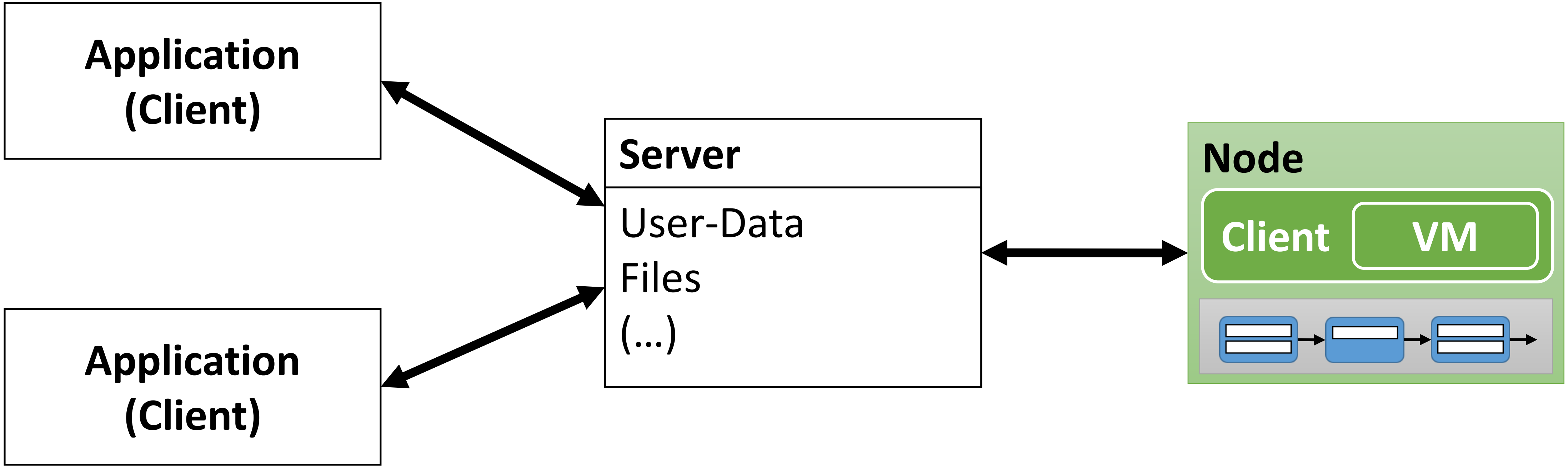
Accounts on client...



User manages accounts:

- Signing on client necessary
- Integration in Client difficult
- Can users manage accounts?

Accounts on server...



Manage Accounts for user:

- Server signs with client private-key+password
- Server = „Beacon of trust“
- Real benefit questionable...

Why we cannot just go DApp yet…

- The technology & tools are not there yet
- Restrictions to one ledger/technology/currency
- Usability ☹
- No (good) mobile versions of clients
- Reliability & Risk
- Volatility (speculation with cryptocurrencies)
- For most industries: No acceptance of the (end-)user

The future of DApp…

- Messaging, Distributed File Exchange
- Web Assembly - https://www.w3.org/wasm/
- eWASM - https://github.com/ewasm

Ethereum

History, Status, testnet, Outlook

History

www.ethereum.org

Proposed	2013
First Release	2015
Current State	Beta
Cryptocurrency	Ether

Most advanced smart contract platform to date

Olympic	→	Frontier	→	Homestead	→	Metropolis	→	Serenity
05/2015		07/2015		03/2016				

Within these releases there are numerous client updates (hard and soft forks).

Clients

- Most popular

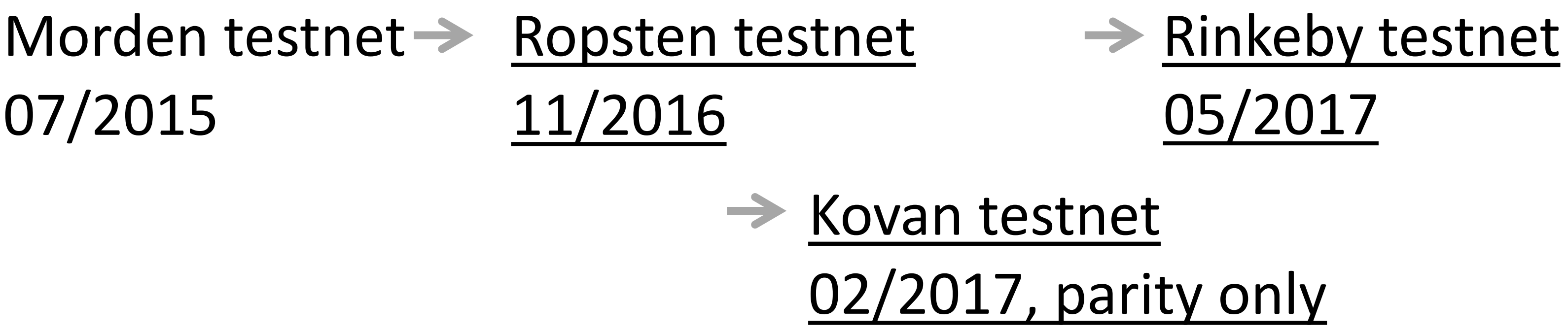
geth	go	Ethereum Foundation
parity	rust	Ethcore
eth	C++	„Ethereum Community“
pyethapp	Python	Pyethereum

- Other

EthereumJ	Java
ethereumH	Haskell
ruby-eth	Ruby

Network

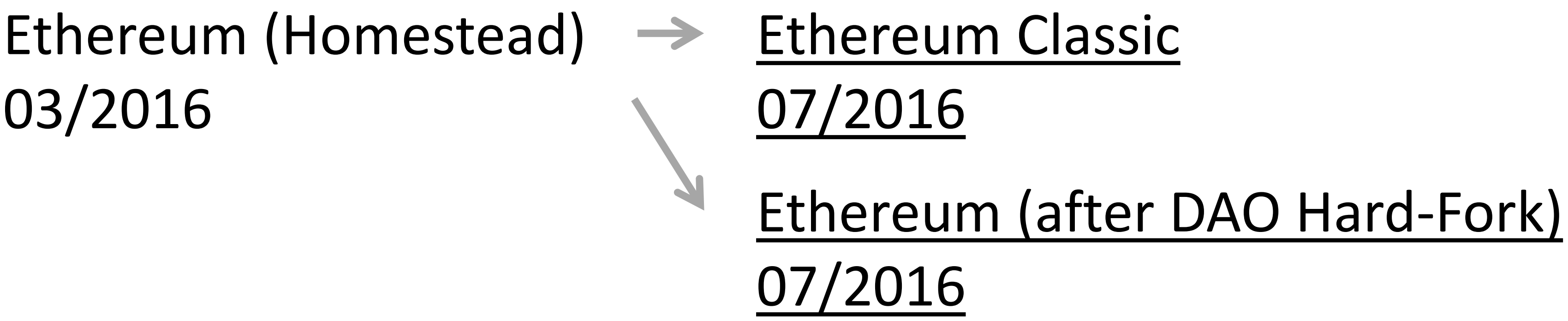
- Testnet



Morden testnet 07/2015 → Ropsten testnet 11/2016 → Rinkeby testnet 05/2017

→ Kovan testnet 02/2017, parity only

- Production



Ethereum (Homestead) 03/2016 → Ethereum Classic 07/2016

→ Ethereum (after DAO Hard-Fork) 07/2016

Smart Contract Platforms

Selected Platforms that enable Smart Contracts

Ethereum

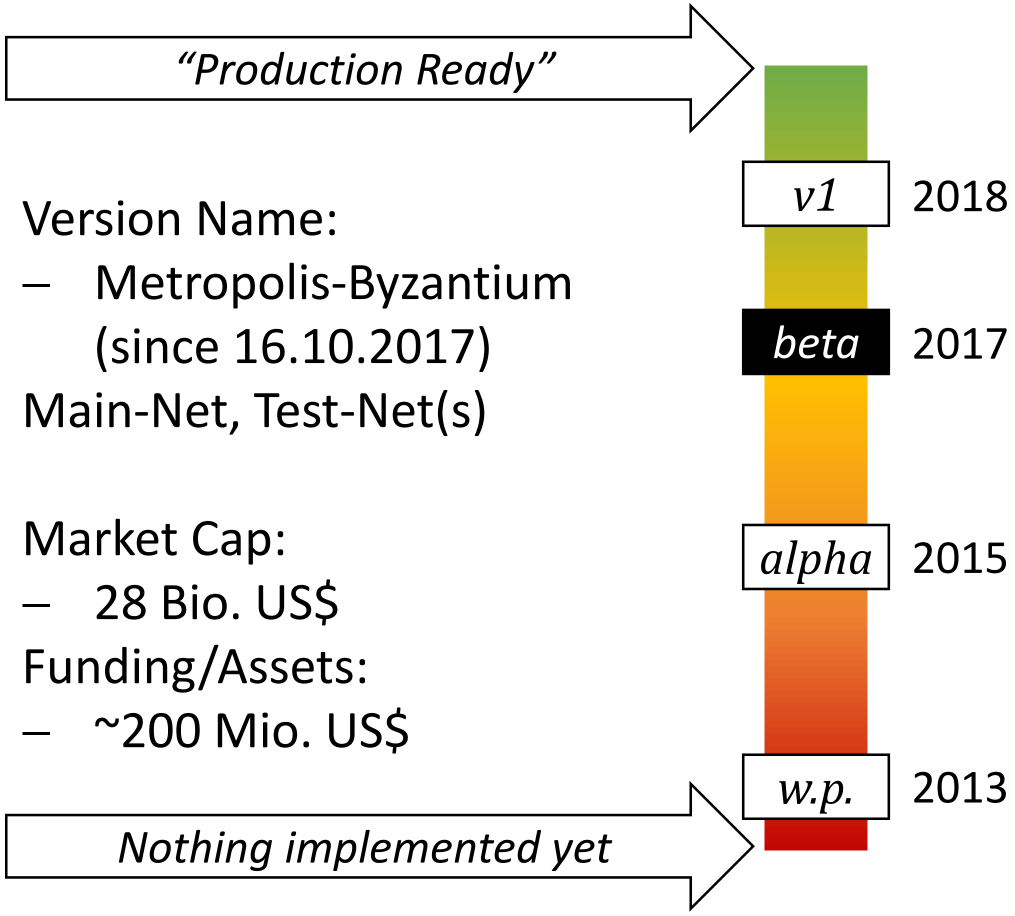
Key Feature

- Currency: ETH
- Smart Contracts
- EVM
- Proof of Work/Stake (Casper)
- Wisper (Messaging)
- Swarm (Distributed File Exchg.)

- Ethereum Foundation
  - Vitalik Buterin
- Ethcore
  - Gavin Wood
- Ecosystem: geth, parity, eth, (…)

Key People / Community

Status



Version Name:

- Metropolis-Byzantium (since 16.10.2017)

Main-Net, Test-Net(s)

Market Cap:

- 28 Bio. US$

Funding/Assets:

- ~200 Mio. US$

NEM (“New Economy Movement”)

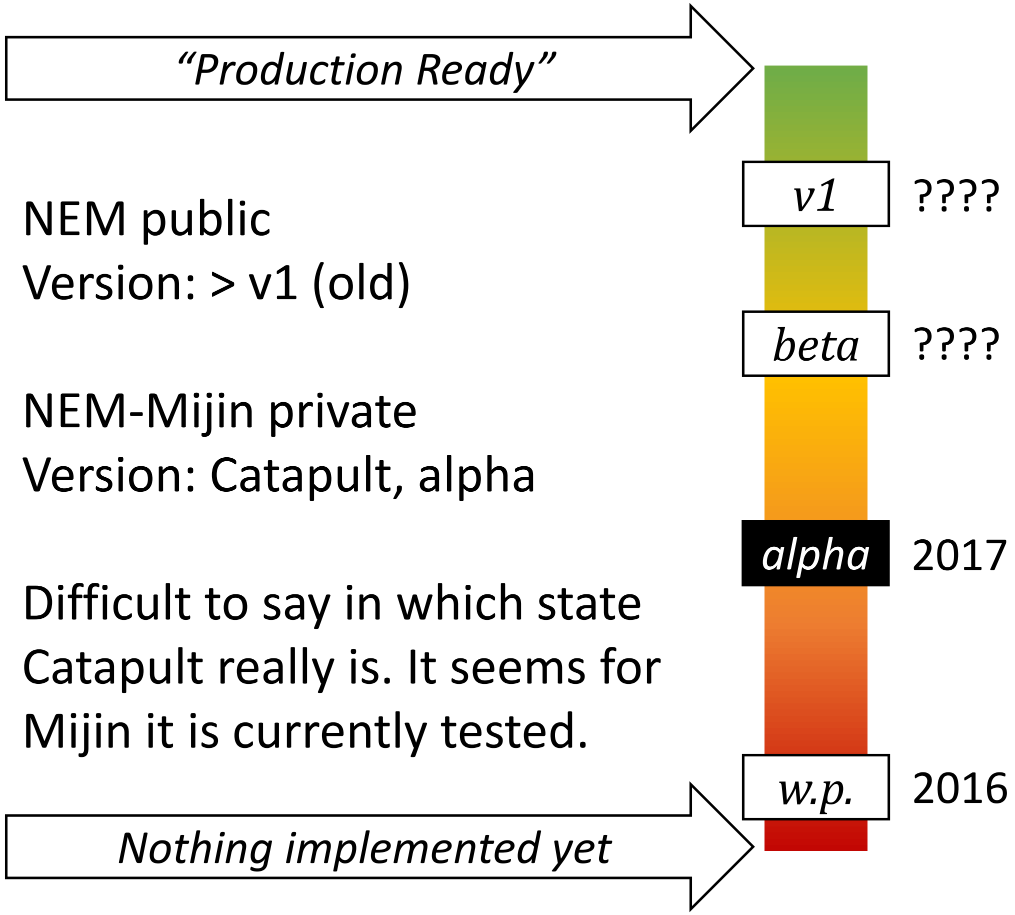
Key Feature

- Currency: XEM
- Smart “Assets”
- Java (Catapult in C++)
- Messaging (and more)
- Proof of Importance
- Focus on “enterprise” with Mijin

Key People / Community

- Dragonfly Fintech
- NEM Foundation
- Nem.io (non profit organization)

Status



NEM public
Version: > v1 (old)

NEM-Mijin private
Version: Catapult, alpha

Difficult to say in which state Catapult really is. It seems for Mijin it is currently tested.

NEO (Antshare), the "Chinese Ethereum"

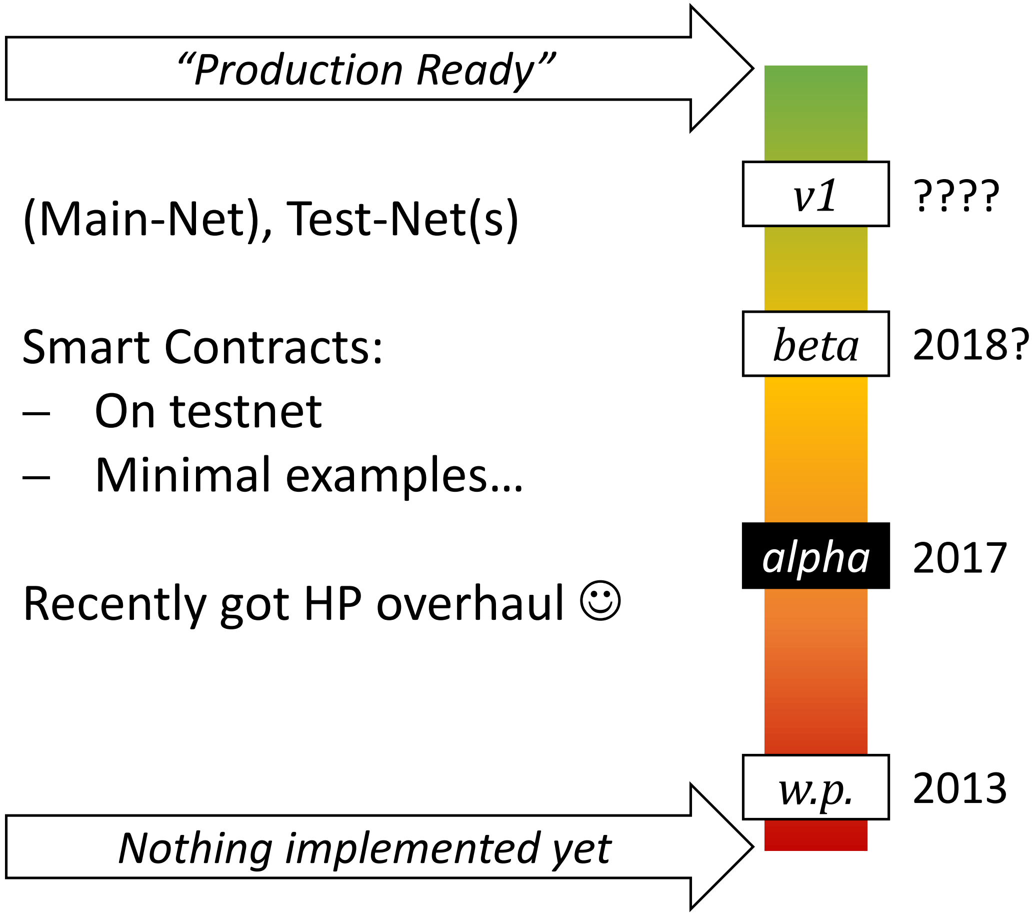
Key Feature

- Currency: NEO
- Smart Contracts (multi-language)
- Identity Management
- Delegated Byzantine Fault Tolerance (Proof of Stake)
- "Legal Ready"

- NEO Project (NEO)
- City of Zion (NEON)
- Various apps, monitors (community)

Key People / Community

Status



"Production Ready"

v1	????
beta	2018?
alpha	2017
w.p.	2013

Nothing implemented yet

(Main-Net), Test-Net(s)

Smart Contracts:
- On testnet
- Minimal examples…

Recently got HP overhaul ☺

EOS

Key Feature

- Currency: EOS
- Delegated Proof of Stake (DPOS)
- Transactions as Proof of Stake (?)
- Messaging, Distributed File Exchg.
- WASM
- VM Integration (Smart Contracts)

- block.one

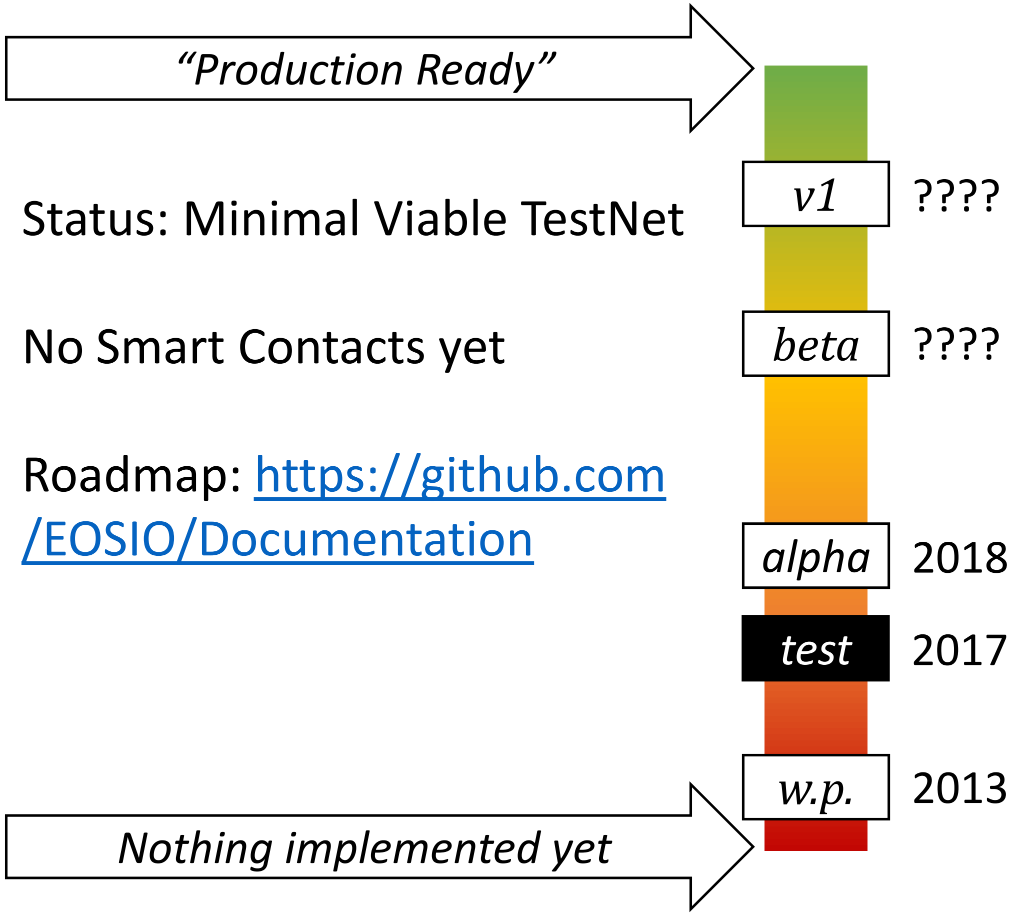
Key People / Community

Status

Status: Minimal Viable TestNet

No Smart Contacts yet

Roadmap: https://github.com/EOSIO/Documentation



Crypti (now Lisk?)

Key Feature

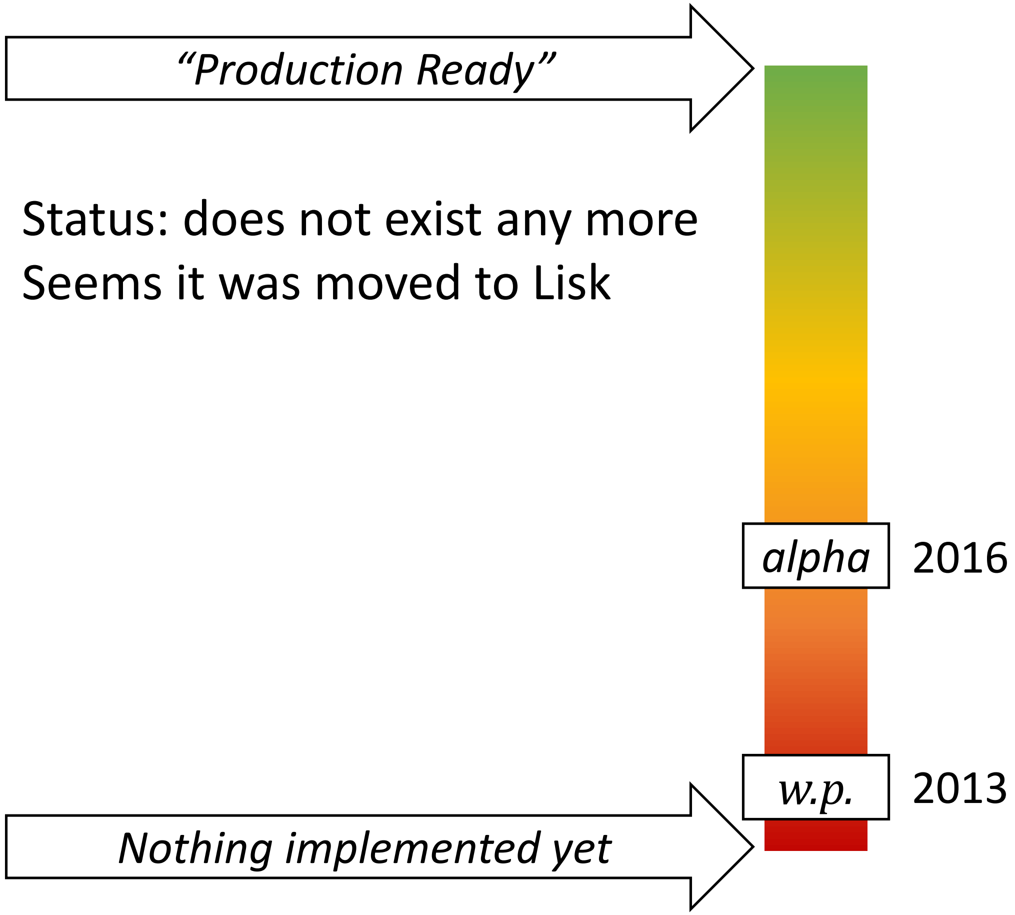
- Currency: XCR (LSK)
- Nothing too special
- DApps/DApp development

Key People / Community

- Crypti Foundation
- Lisk

Status

Status: does not exist any more
Seems it was moved to Lisk



More Smart Contract Platforms…

Ethereum based/on-top-of

- Ethereum Classic
- Ubiq
- Shift
- Soil (just Smart Contracts?)
- Expanses

Bitcoin based/on-top-of

- Counterparty
- Omni
- Rootstock RSK
- Qtum

Crypti based

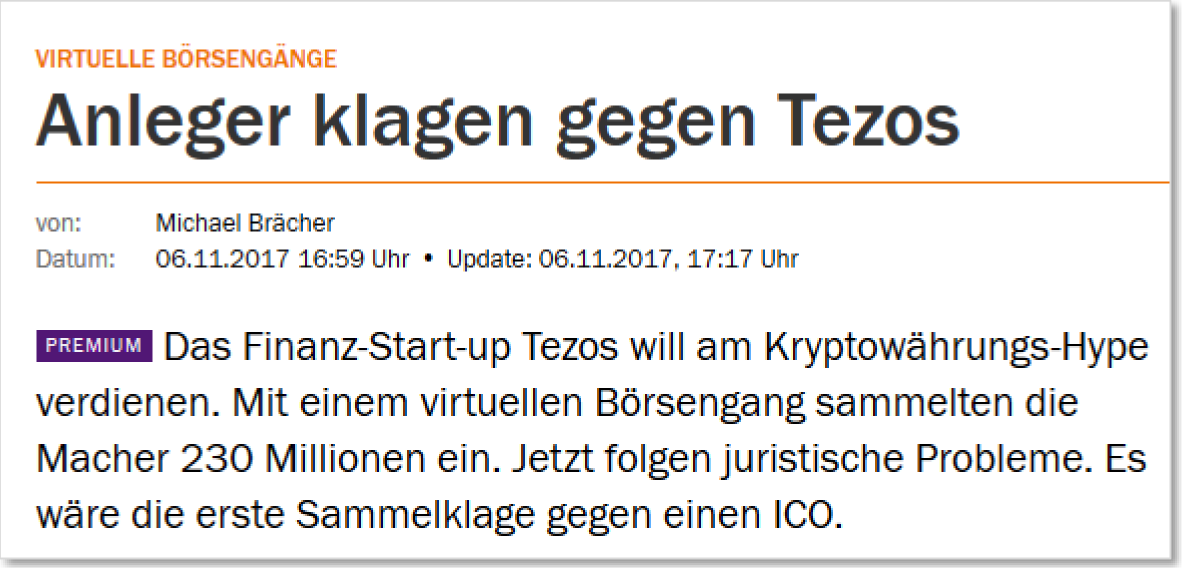
- Lisk
- Ark
- Rise

Other

- Aeternity
- Agoras Tauchain
- BOSCoin
- Burst
- NXT
- Tezos
- Waves

The other side of the coin…

- Raised in 232 US$ through ICO
  - raised BTC and ETH; token sale
  - https://www.tezos.com/faq

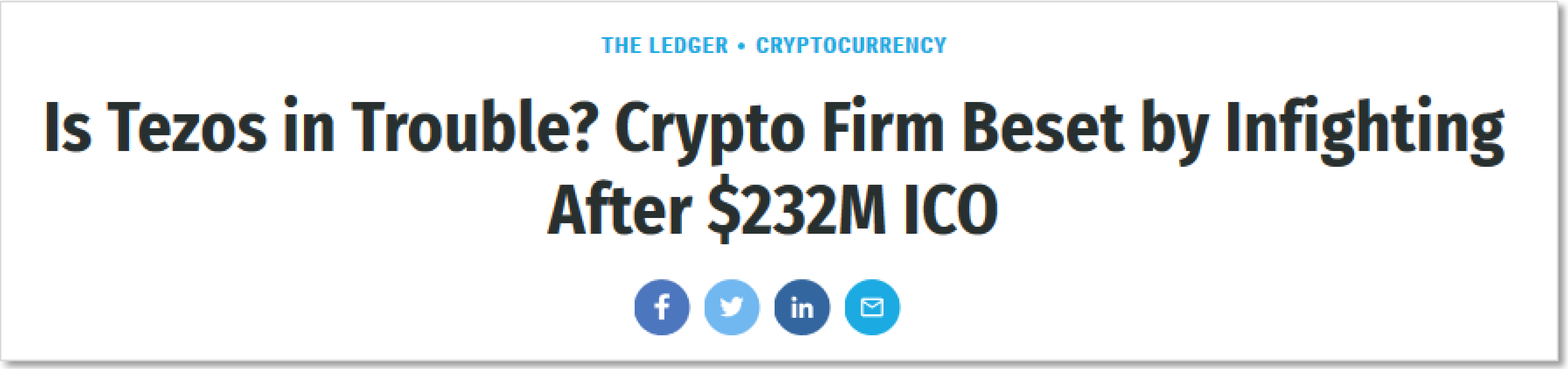


VIRTUELLE BÖRSENGÄNGE

Anleger klagen gegen Tezos

von: Michael Brächer
Datum: 06.11.2017 16:59 Uhr • Update: 06.11.2017, 17:17 Uhr

PREMIUM Das Finanz-Start-up Tezos will am Kryptowährungs-Hype verdienen. Mit einem virtuellen Börsengang sammelten die Macher 230 Millionen ein. Jetzt folgen juristische Probleme. Es wäre die erste Sammelklage gegen einen ICO.



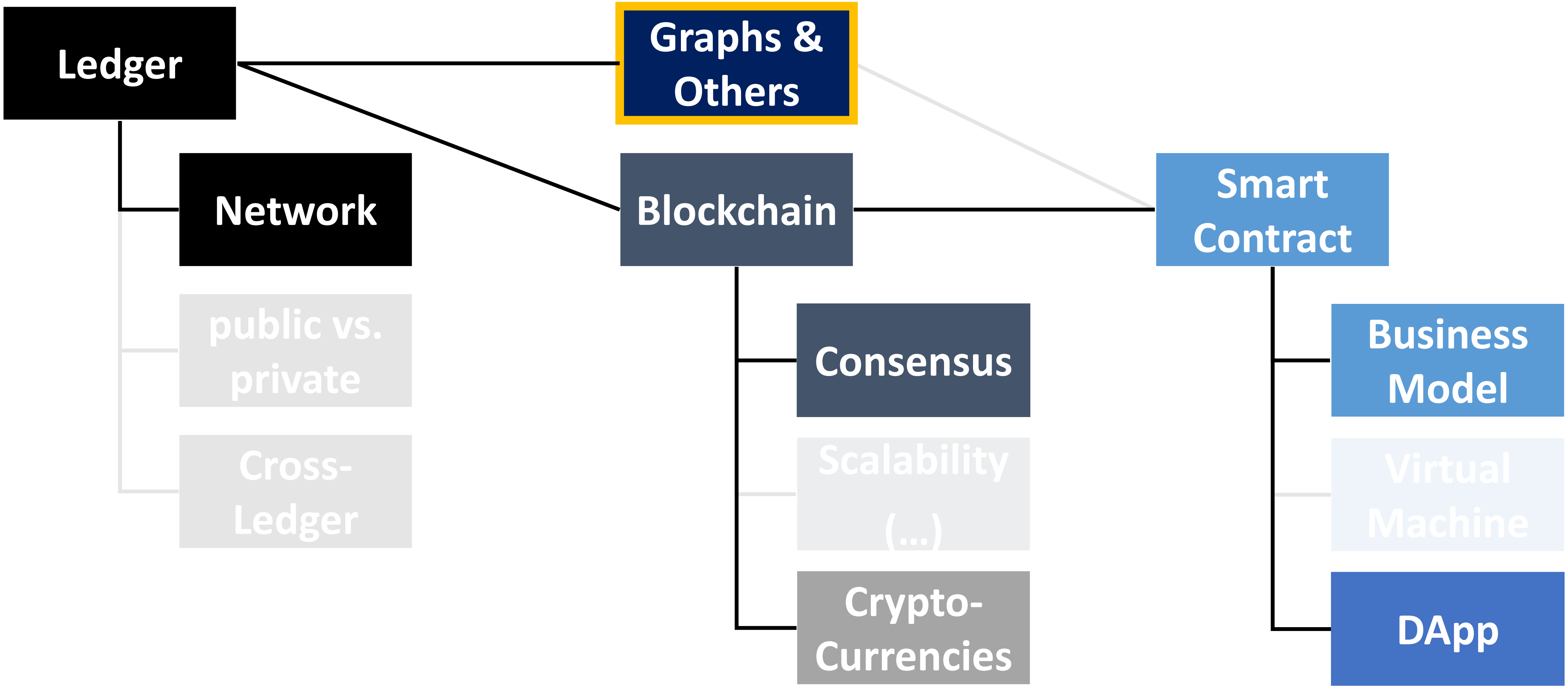
THE LEDGER • CRYPTOCURRENCY

Is Tezos in Trouble? Crypto Firm Beset by Infighting After $232M ICO

Why does it have to be a “chain”?

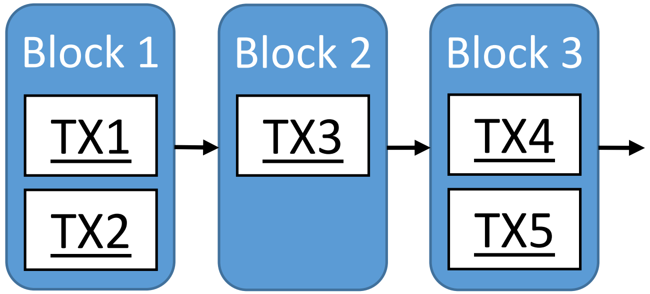
And now for something completely different…

Let‘s focus on…



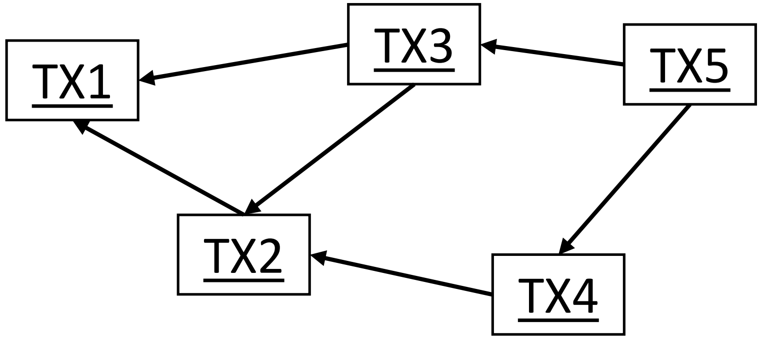
Blockchain „3.0“ – The Tangle

Blockchain



- Linked-List (of blocks)
- Each List-Node contains transactions
- Blocks are created by dedicated network-nodes (miners)

Tangle



- Graph (of transactions)
- Each new transaction confirms two already existing transaction
- Everybody “confirms”
- “Even more decentral”
- “Even more eventually consistent”

IOTA, “the Tangle”

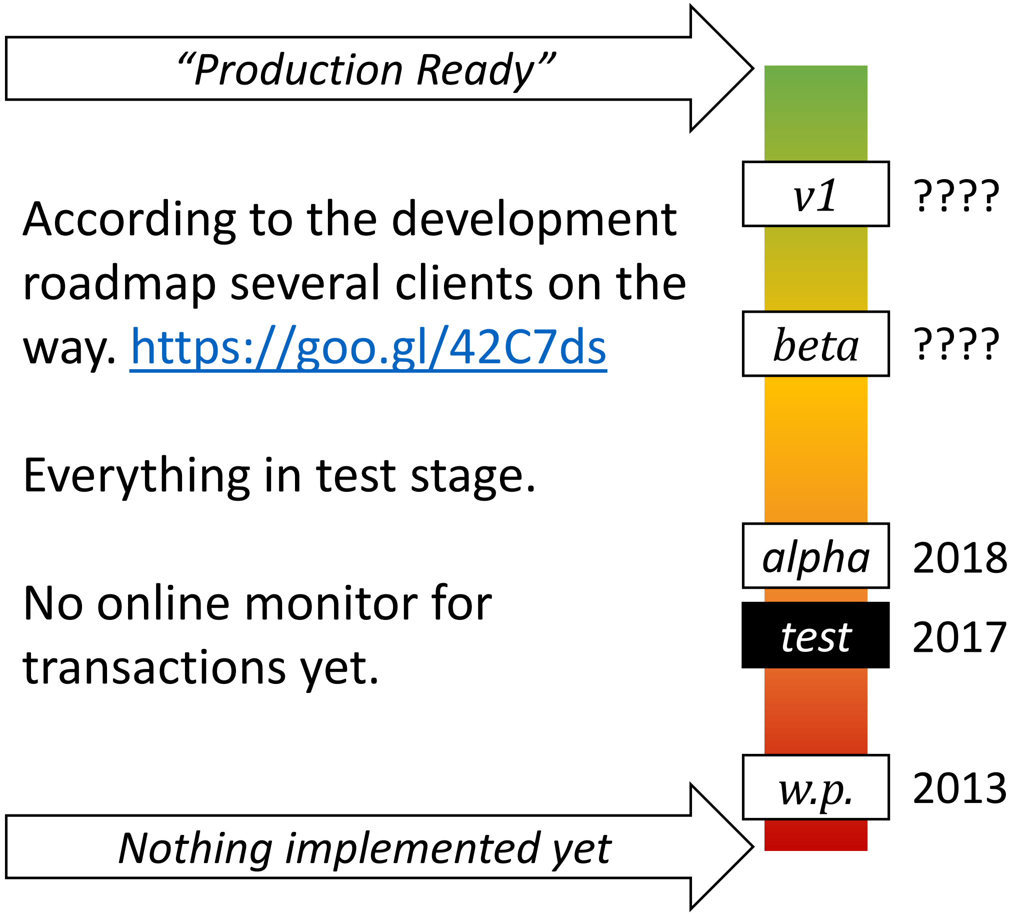
Key Feature

- Currency: MIOTA
- DAG (Directed Asyclic Graph)
- New transactions confirm existing
- Smart Contracts
- For the “Internet of Things”

Key People / Community

- IOTA Foundation
  - David Sønstebø (founder)

Status



According to the development roadmap several clients on the way. https://goo.gl/42C7ds

Everything in test stage.

No online monitor for transactions yet.

Smart Contract Products

Use-Cases, What can one do?, Selected Existing Products

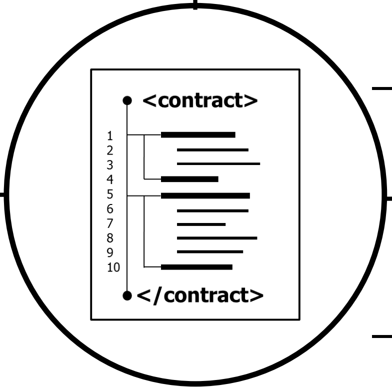
Use-Case Characteristics

Transparency & Decentralization

- Everything is visible
- Public validation
- Public vs. Private – sense of obscurity?
- Everybody, anything, any time
- Unstoppable

Fairness & Trust

- Multiple parties establish trust
- Who is allowed to do what?
- When is who allowed to do what?
- Data can only go in (no outside calls)



Collaboration & Motivation

- Incentives for stakeholders
- Costs for services
- Benefit should be given

Efficiency & Maintenance

- One cannot argue with a contract
- Mistakes are unchangeable
- Higher complexity = Higher cost
- Complexity restrictions

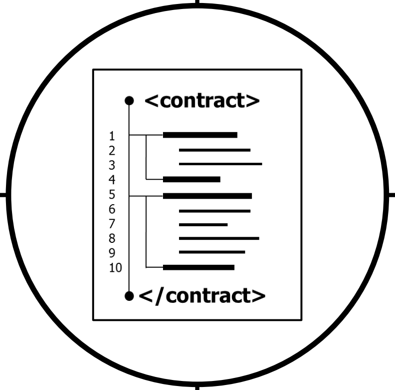
Smart Contract Challenges

Usability

- End-Users and:
  - The technology (processes)
  - Accounts
  - Assets
- Lack of (good) user interfaces

Regulation

- Identity
- Law Situation
- Governance



- Restriction to Ledger
- Lack of standards

Risk

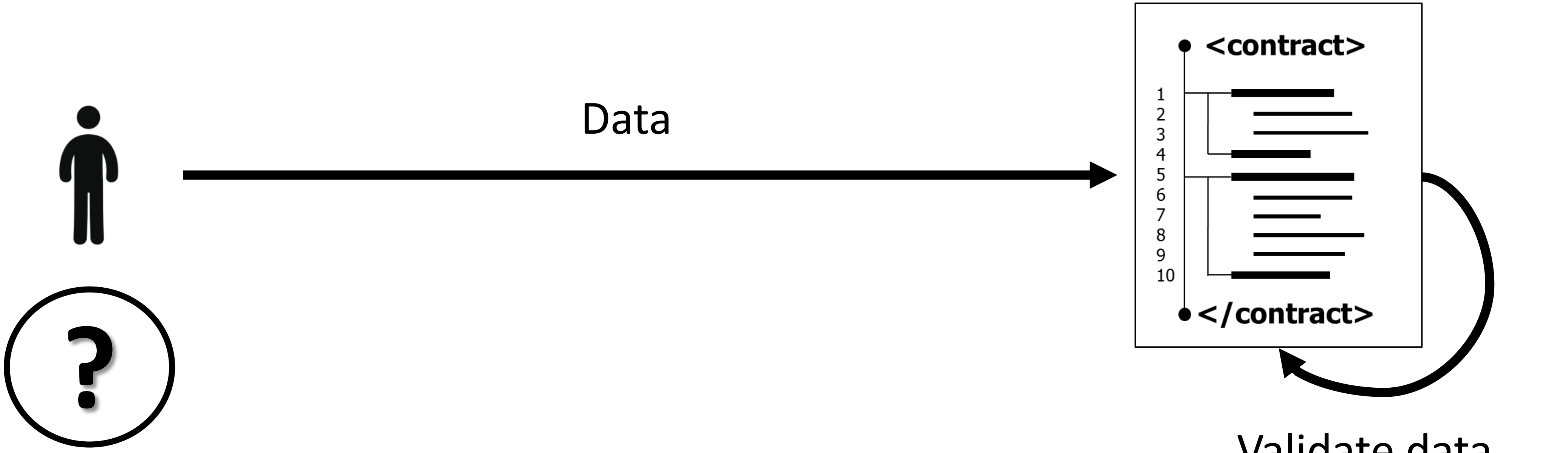
- Volatility of the currencies
- Volatility of the software

Volatility

Smart Contract Service Providers

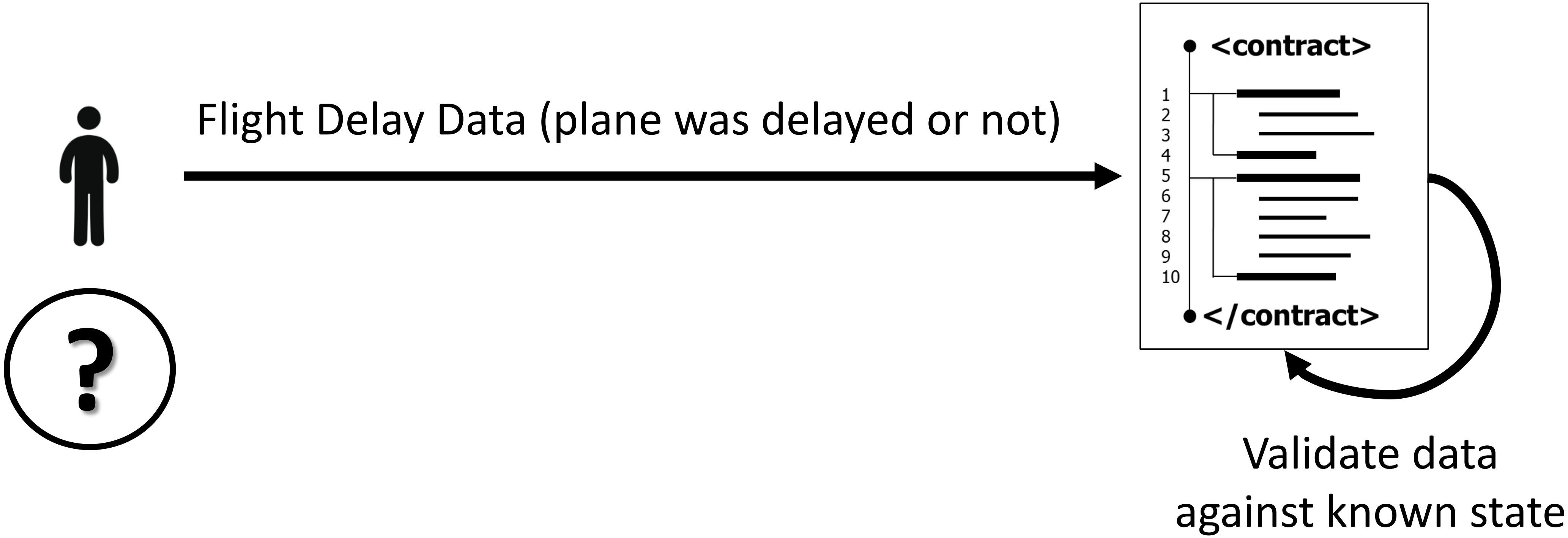
Products and Services ontop of the technology

Smart Contracts and Data



Validate data
against known state

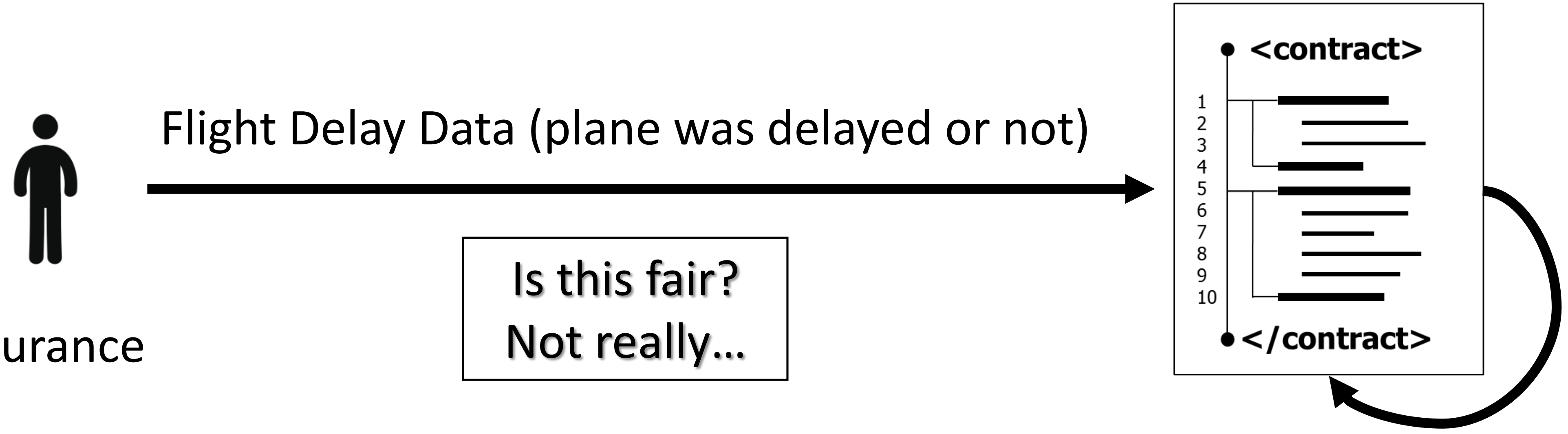
Example: Flight Delay Insurance



Flight Delay Data (plane was delayed or not)

Validate data against known state

Example: Flight Delay Insurance



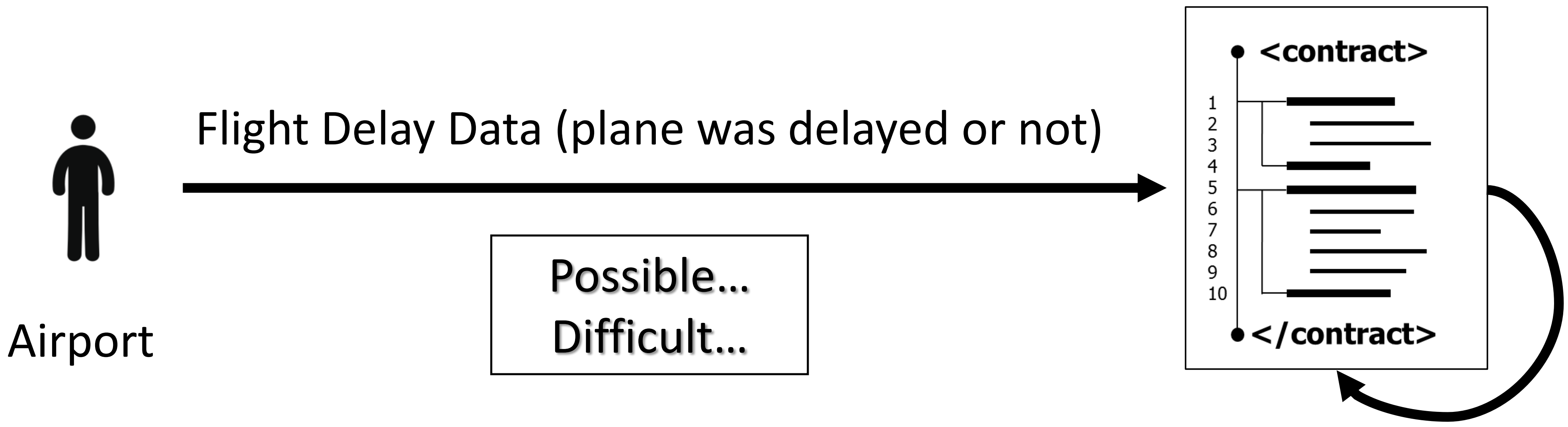
Flight Delay Data (plane was delayed or not)

Insurance

Is this fair?
Not really…

Validate data
against known state

Example: Flight Delay Insurance



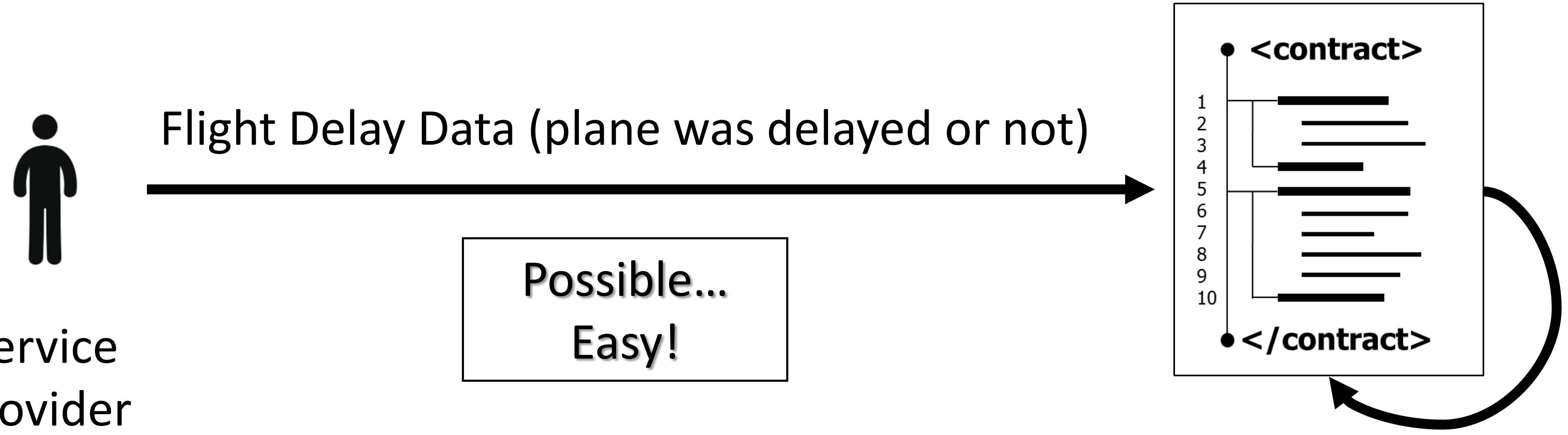
Flight Delay Data (plane was delayed or not)

Airport

Possible…
Difficult…

Validate data
against known state

Example: Flight Delay Insurance



Flight Delay Data (plane was delayed or not)

Service
Provider

Possible…
Easy!

Validate data
against known state

Oraclize

Use Case

- Identity
- Data Provider Services (SC)
- DApp Prototypes
- Governance

Technology

- Services ontop of Ethereum (mostly Smart Contract based)
- Rootstock (Bitcoin sidechain)
- Bitcoin research

Oraclize Team
Not transparent who is behind

http://www.oraclize.it

Team / Links

Several products in different status.
Most important:

- Oracle Service
- Work on Identity

prod 2017

alpha 2017

Status

Gaming

Smart Contract Products in the online gaming industry

First Blood

Use Case

- eSports
- In-platform tokens as stake
- Smart Contract serves as escrow
- Not much decision logic in SC
- Players can serve as witness nodes (fraud detection)

Technology

- Ethereum Smart Contracts
- Online Platform
- MetaMask
- Interfaces to:
  - Dota 2
  - Steam

Team / Links

First Blood Team
Community driven

https://firstblood.io
https://github.com/firstbloodio

Status

Contract: Extended token contract

Version: Beta, update 13

beta 2017

alpha 2017

Ownage

Use Case

- Trade Game Content
- “Proof of Ownership”

Technology

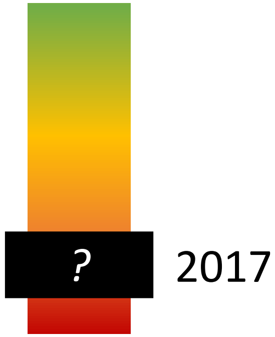
- Ethereum Smart Contracts

Team / Links

http://ownage.io/
https://github.com/ownage-ltd

Status

Version: Nothing operational



Supply Chain

Smart Contract Products in the supply chain industry

Modum

Use Case

- Supply Chain
- Shipment of Pharmaceutical Goods
- Sensors hold private key
- Sensors register at smart contract
- Contract validates shipment status when sensors send data

Technology

- Ethereum Smart Contracts (current)
  - IOTA (future)
- Most likely some server in between
- Smartphones (without node)
- Sensors

Team / Links

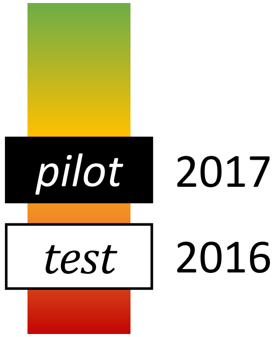
Modum Team
Partners

https://modum.io
https://github.com/modum-io

Status

Status: Pilot Projects
Contract: v1.1

43 companies
1000+ shipments



Gambling

Smart Contract Products in the gambling industry

Quanta

Use Case

- Smart Contract Lottery
- RanDao+
- Planning to get real lottery license
- Have own token “QNT” ?

Technology

- Ethereum Smart Contracts
- RanDAO, RanDAO+
- Quanta Wallet
- Most likely a server connected to wallet

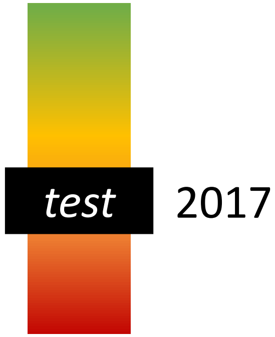
Team / Links

Quanta Team

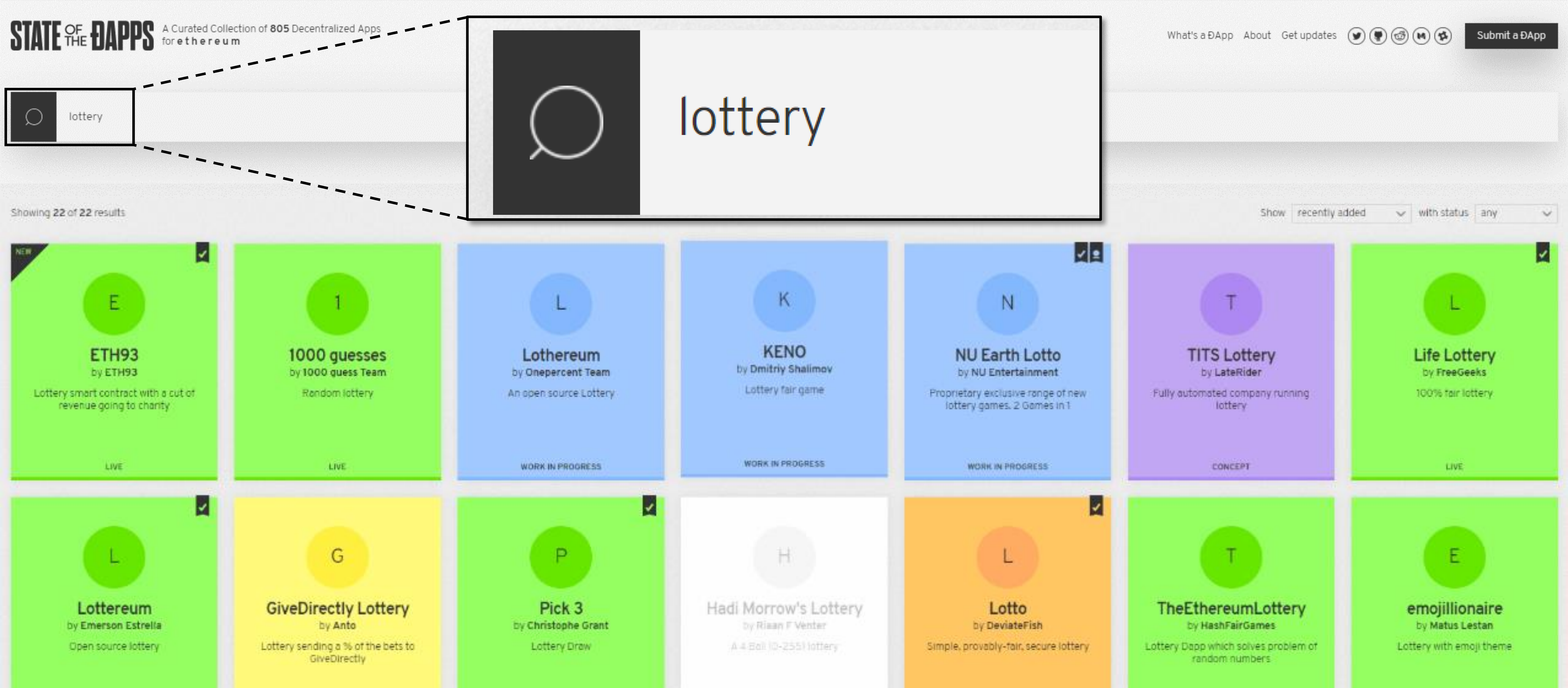
https://www.quanta.im/
https://github.com/tjade273/RanDAOPlus

Status

Status: Wallet operational
Lottery: Test (not in wallet yet)



From playing with technology to product



STATE OF THE ÐAPPS A Curated Collection of 805 Decentralized Apps for ethereum

What's a ÐApp About Get updates Submit a ÐApp

lottery

Showing 22 of 22 results

Show recently added with status any

NEW

E
ETH93
by ETH93
Lottery smart contract with a cut of revenue going to charity
LIVE

1
1000 guesses
by 1000 guess Team
Random lottery
LIVE

L
Lothereum
by Onepercent Team
An open source Lottery
WORK IN PROGRESS

K
KENO
by Dmitriy Shalimov
Lottery fair game
WORK IN PROGRESS

N
NU Earth Lotto
by NU Entertainment
Proprietary exclusive range of new lottery games. 2 Games in 1
WORK IN PROGRESS

T
TITS Lottery
by LateRider
Fully automated company running lottery
CONCEPT

L
Life Lottery
by FreeGeeks
100% fair lottery
LIVE

L
Lottereum
by Emerson Estrella
Open source lottery

G
GiveDirectly Lottery
by Anto
Lottery sending a % of the bets to GiveDirectly

P
Pick 3
by Christophe Grant
Lottery Draw

H
Hadi Morrow's Lottery
by Riaan F Venter
A 4 Ball (0-255) lottery

L
Lotto
by DeviateFish
Simple, provably-fair, secure lottery

T
TheEthereumLottery
by HashFairGames
Lottery Dapp which solves problem of random numbers

E
emojillionaire
by Matus Lestan
Lottery with emoji theme

Insurance

Smart Contract Products in the insurance industry

AXA Fizzy

Use Case

- Flight Delay Insurance

Technology

- Ethereum Smart Contracts
- Webpage
- (iOS, android) – not in store…
- Not much technical insight. Not sure how they get the flight data in.

AXA Insurance

https://fizzy.axa/

Team / Links

Status: Beta, Test

Coverage:
2017: Flights from Paris to US
2018: Worldwide

beta 2017

Status

Energy

Smart Contract Products in the energy industry

Brooklyn Micro-Grid

Use Case

- P2P Electricity Sharing
- Building up micro-grids in densely populated areas
- Share your electricity from (e.g. solar) power with your neighbors

Technology

- Ethereum Smart Contracts (just test)
- App & Platform

Team / Links

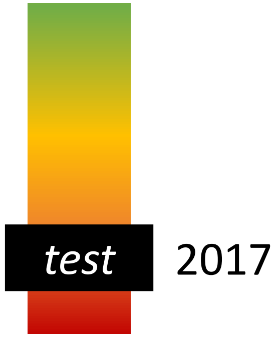
LO3 Energy (prev. TransActive Grid)
Siemens

https://www.brooklyn.energy/
http://lo3energy.com/

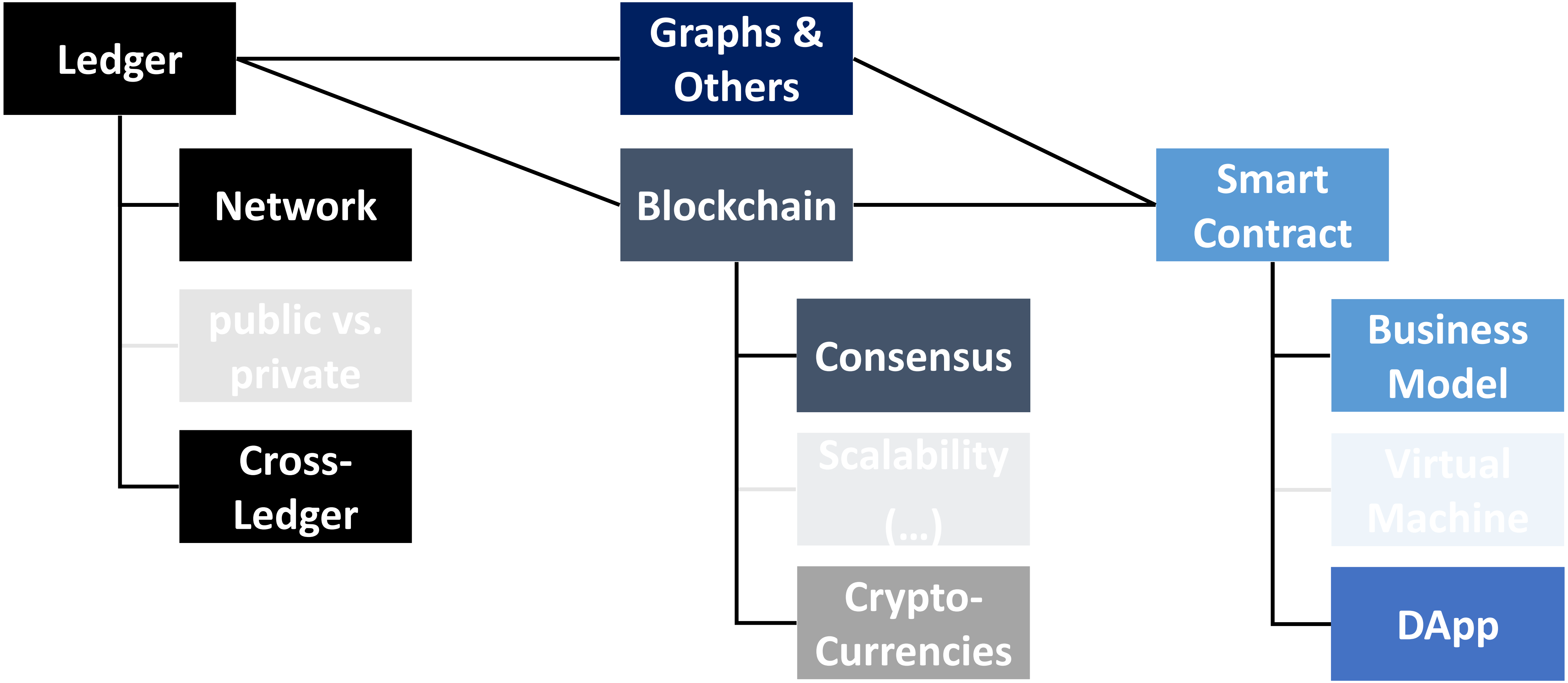
Status

Status: No Smart Contracts in current version

Simple test with Ethereum Smart Contacts done



Questions?


Ledger
Network
public vs. private
Cross-Ledger
Graphs & Others
Blockchain
Consensus
Scalability (...)
Crypto-Currencies
Smart Contract
Business Model
Virtual Machine
DApp