Research Group
Distributed Systems

Christian-Albrechts-Universität zu Kiel

Technische Fakultät

Blockchain & Smart Contracts

Olaf Landsiedel

Exams

- First examination period
  - Exam date: Monday, 03. February 2020, 10:30 - 12:30
  - Registration deadline: Monday, 27. January 2020
  - Exam Review: Wednesday, 19. February 2020, 12:00 - 14:00
- Second examination period
  - Exam date: Thursday, 26. March 2020, 10:30 - 12:30
  - Registration deadline: Thursday, 19. March 2020
  - Exam Review: Tuesday, 14. April 2020, 12:00 - 14:00
- iLearn
  - Details, rooms, registration links, …

Last time?

- More on Blockchain
  - Mining

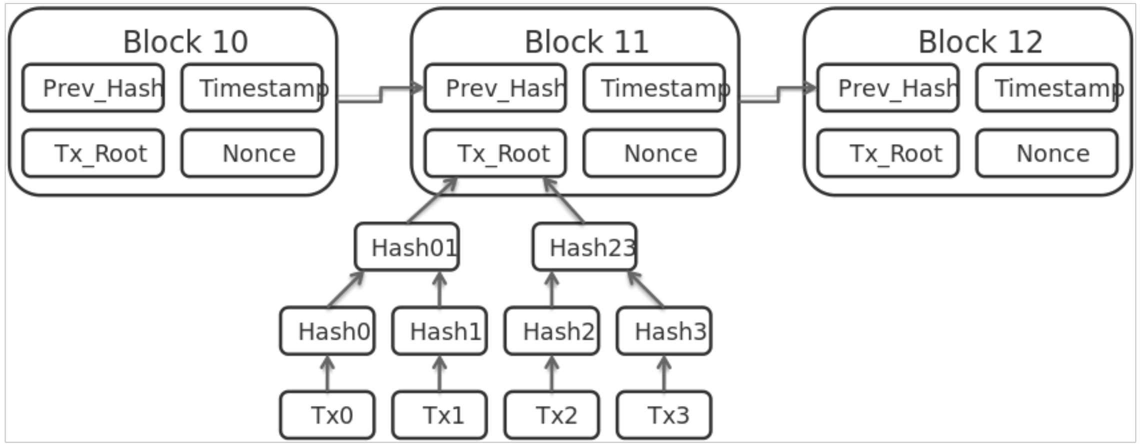
Recap

- What is a blockchain?
  - Blockchain is a decentralized and distributed ledger
    - that records all the transactions that occur on the network
- What is a ledger
  - a book in which things are regularly recorded, especially business activities and money received or paid (Cambridge Dictionary)
    - Deutsch: Konto, Kontobuch, Bestandsbuch

Recap

- Blockchain = Distributed Ledger + Consensus
- A list of transactions between accounts (a ledger) are stored on distributed “nodes”.
- New transactions are periodically added into a block.
- Nodes use an agreed upon protocol to reach consensus on when a new block is appended to the chain of previous blocks.

Merkle Trees

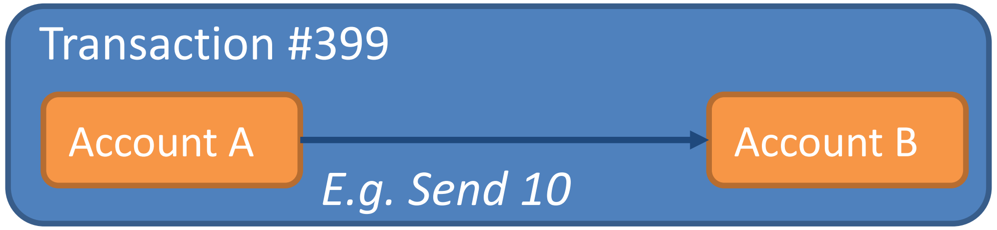


- Why use Merkle Trees
  - Efficient storage
  - Efficient to verify, especially if one entry is broken
    - Tree traversal instead of checking all entries

Transaction

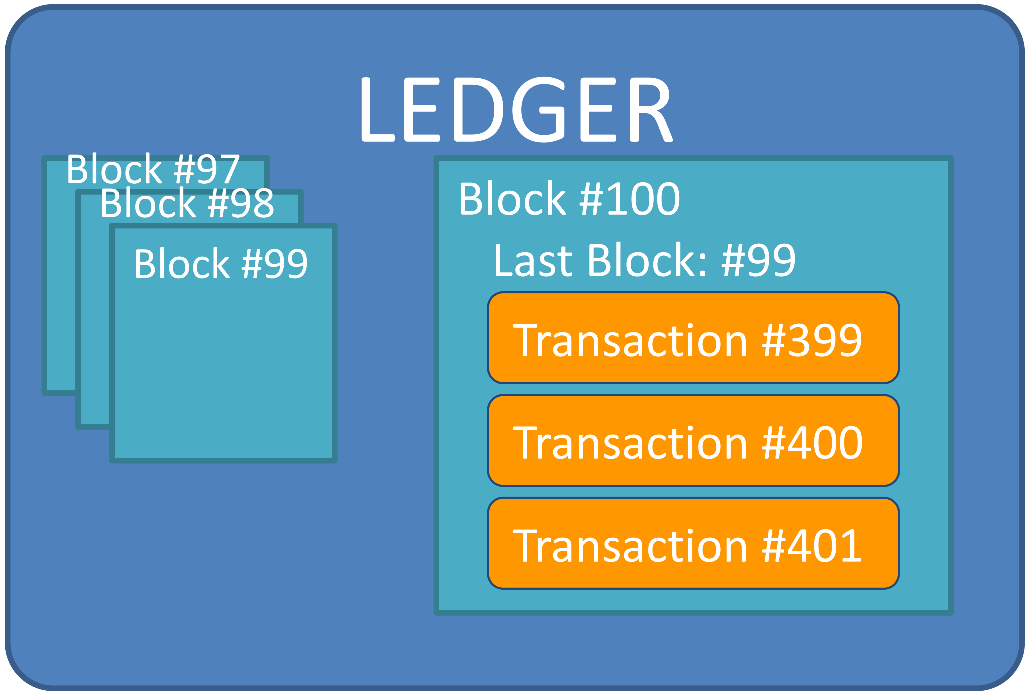
An **example transaction** could be:

Account A will send 10 tokens to Account B

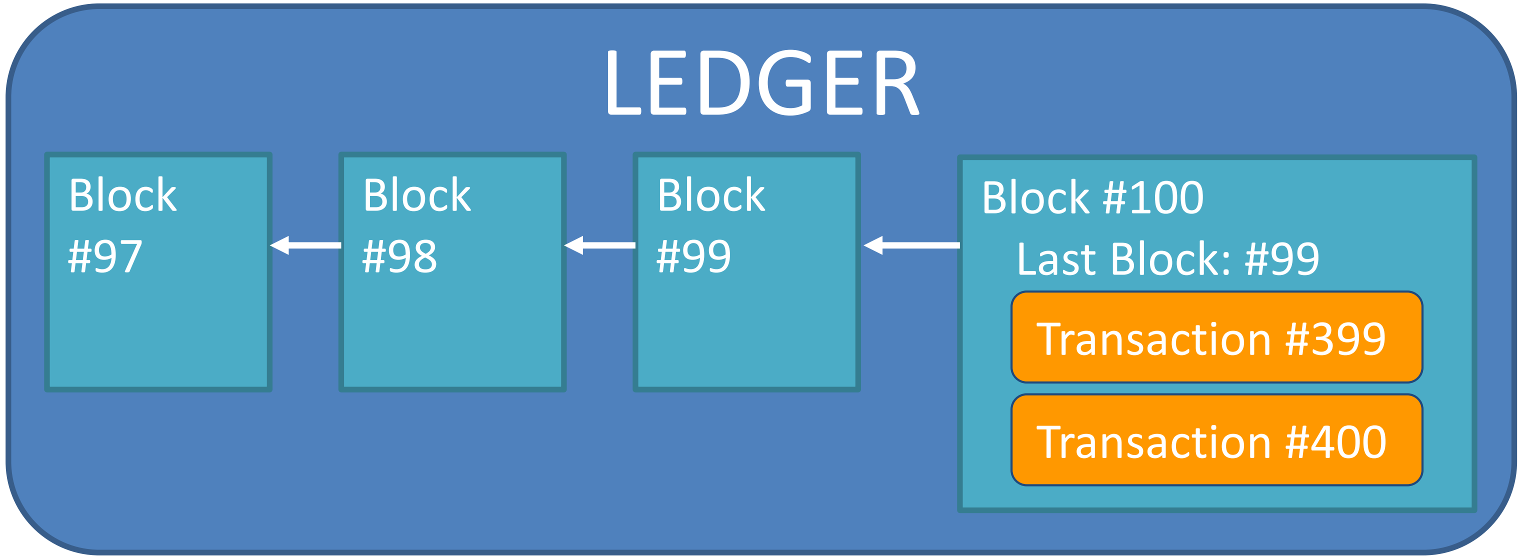


Ledger and Blocks

- Ledger
  - List of transactions, grouped into blocks
  - Blocks are generated on a time interval (e.g. every 5 minutes)



Blocks are Chained Together



- The ledger is a chain of blocks
- Each block is created with a pointer to the previous block
  - creating a blockchain

Blockchain

- Blockchain can do much more than money
  - Run run legal contracts
    - As code (and not legal text)
    - Including all conditions and cancelation details
  - B2C
    - Rental of apartments
    - Utility bills (electricity, water, …)
    - …
  - B2B
    - Logistic
    - Supply chain
    - …

Today

- Smart Contracts
- Ethereum
- Hyperledger

SMART CONTRACTS

Smart Contract

- A smart contract
  - Is code
  - Gets executed / triggered by events
    - Example: Incoming transaction to the address of the contract
  - Modifies the blockchain
    - Does transactions:
      - of crypto-currency
      - Or other things, example: ownership of an item

- Are executed by the blockchain network
  - Removes need for trusted third party
    - Trust the distributed blockchain instead
  - No one can violate the contract
  - No one can question the contract
    - It is public visible code

Example Smart Contract I

- "Saving account"
  - People transfer money to an account
  - Every month an interest is added
  - People can take their money with interest out of the saving account

Example Smart Contract II

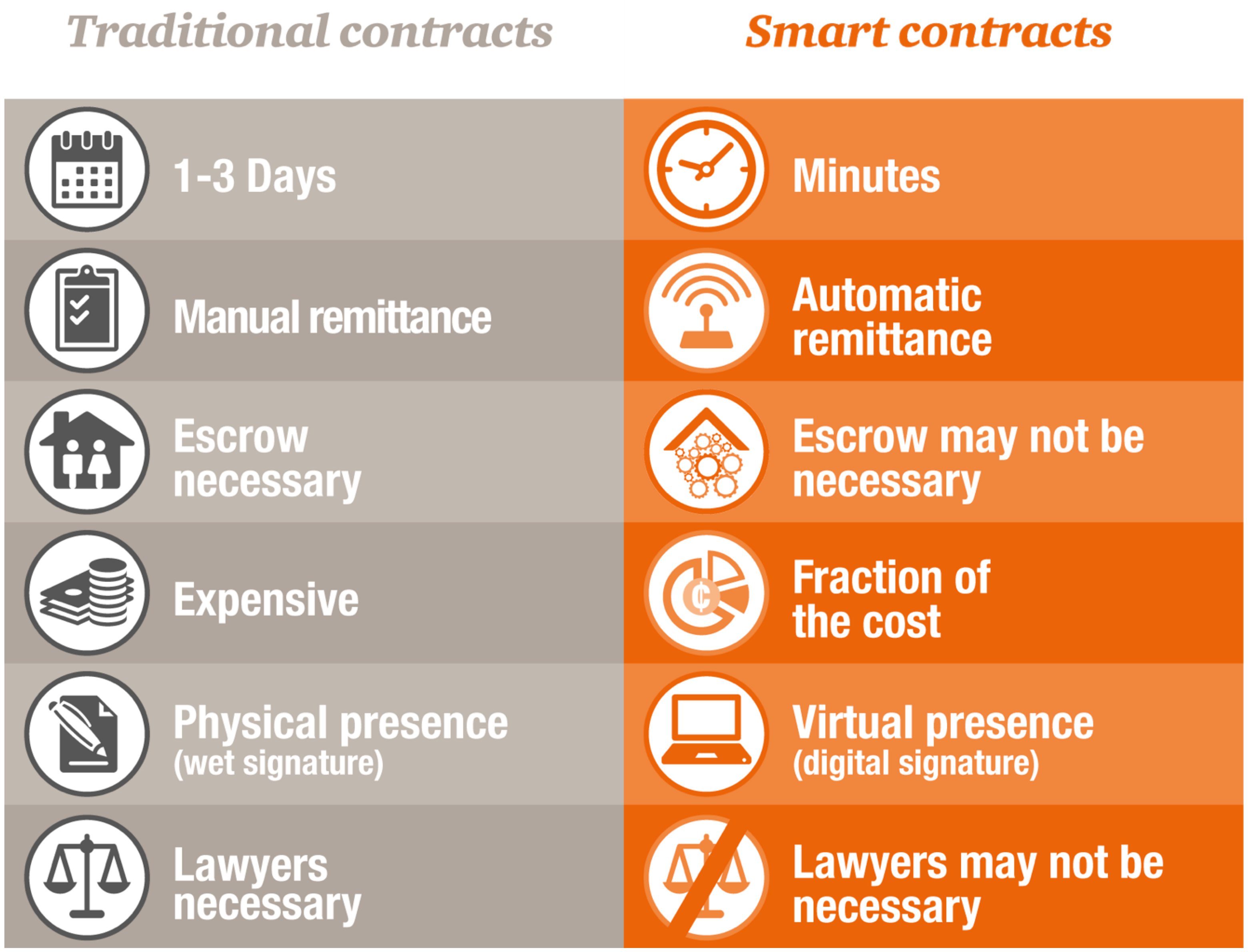
- Ownership of an item, e.g., a car
  - Each new car produced is added to the blockchain
    - Owner: production company
  - During sales
    - Ownership is changed,
      - Cryptographically signed by previous owner and new owner
  - During maintenance
    - All changes and repairs are recorded and connected to the car

Smart Contracts

- Any business relationship can be a smart contract
  - Write is it as code
  - And not legal contract
- But?
  - Writing legal contracts is a skill, requires training
    - To avoid loopholes etc.
  - Writing legal things as code is challenging, too
    - To avoid loopholes etc.
    - Bugs etc.

Smart Contracts

- First proposed by Nick Szabo in 1996
  - Long before Blockchain, Bitcoin, …
- An agreement between two or more parties that is stored on the blockchain
  - Setting the agreement in stone
- Disintermediating the legal system
  - We trust the blockchain
- The contract has an ID and so do the parties involved



Traditional contracts	Smart contracts
1-3 Days	Minutes
Manual remittance	Automatic remittance
Escrow necessary	Escrow may not be necessary
Expensive	Fraction of the cost
Physical presence (wet signature)	Virtual presence (digital signature)
Lawyers necessary	Lawyers may not be necessary

https://www.devteam.space/blog/10-uses-for-smart-contracts/

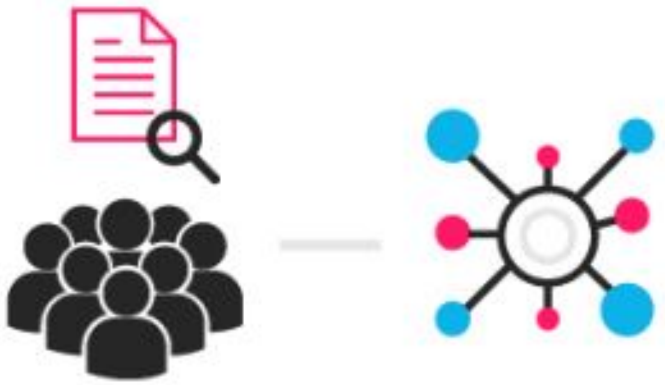
1

An option contact between parties is written as code into the blockchain. The individuals involved are anonymous, but the contact is the public ledger.

2

A triggering event like an expiration date and strike price is hit and the contract executes itself according to the coded terms.

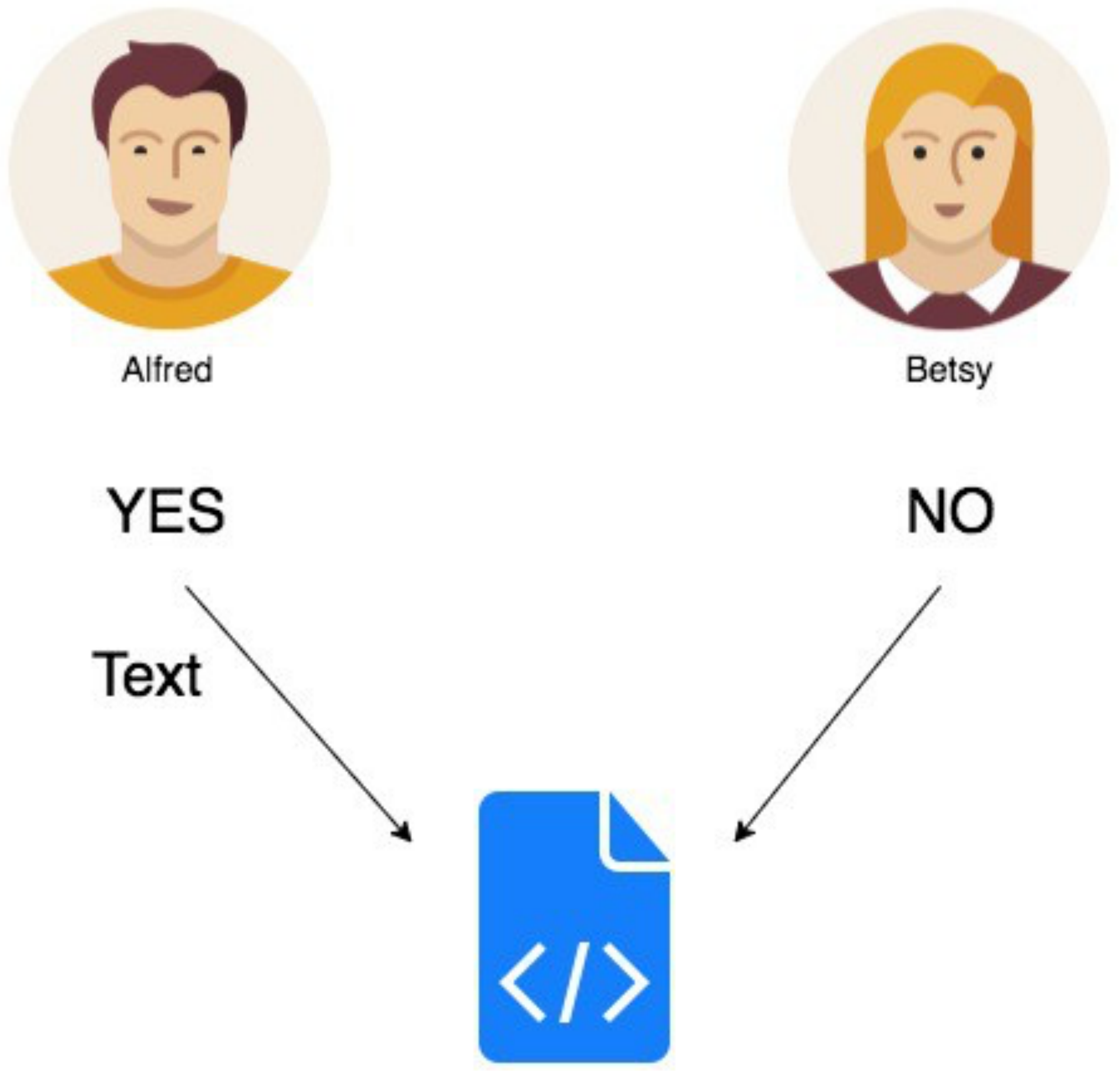
3



Regulators can use the blockchain to understand the activity in the market while maintaining the privacy of individual actors' positions

Smart contract: if … then … else

Departs the train between Utrecht Centraal and Amsterdam Centraal on time?

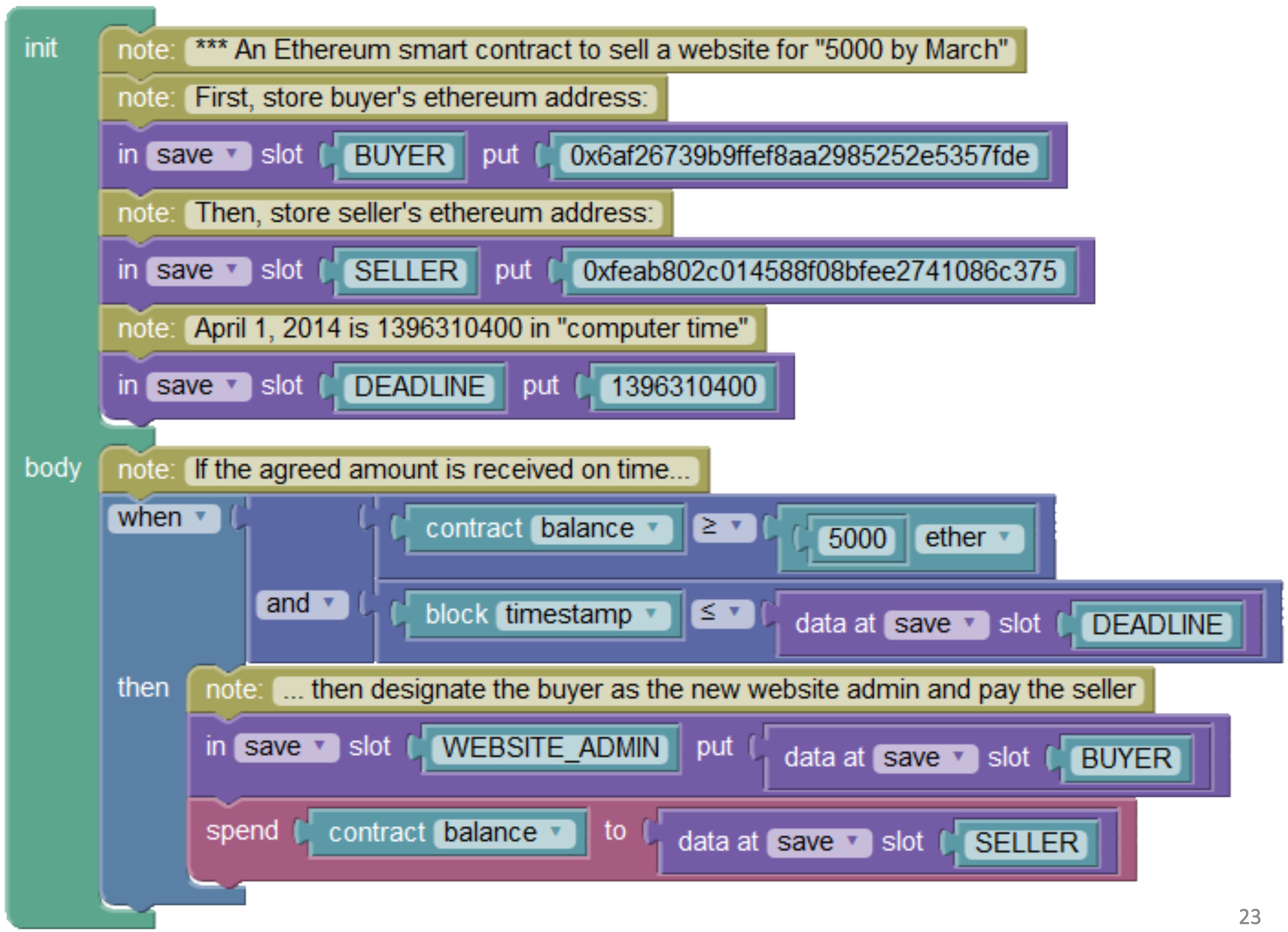


If *the train leaves on time, then* ***Person A*** *will receive 100 euro.*

If *the train leaves late, then* ***Person B*** *will receive 100 euro.*

Example

- Buy a domain name: „xyz.com“
  - Agree on
    - price,
    - latest payment date
    - accounts of seller and buyer
  - If money arrives (on payment date or before), then
    - Change admin of the website to buyer
    - Put money to account of seller



```
init
    note: *** An Ethereum smart contract to sell a website for "5000 by March"
    note: First, store buyer's ethereum address:
    in save slot BUYER put 0x6af26739b9ffef8aa2985252e5357fde
    note: Then, store seller's ethereum address:
    in save slot SELLER put 0xfeab802c014588f08bfee2741086c375
    note: April 1, 2014 is 1396310400 in "computer time"
    in save slot DEADLINE put 1396310400
body
    note: If the agreed amount is received on time...
    when
        and
            contract balance ≥ 5000 ether
            block timestamp ≤ data at save slot DEADLINE
    then
        note: ... then designate the buyer as the new website admin and pay the seller
        in save slot WEBSITE_ADMIN put data at save slot BUYER
        spend contract balance to data at save slot SELLER
```

Smart Contracts: Problems

- Once a smart contract is deployed in the Blockchain
  - Very hard to change
  - What if?
    - It has a bug?
    - Business changes?

DAO hack - Background

- Decentralized autonomous organizations
  - Known as DAOs
  - entities that operate through smart contracts
- Its financial transactions and rules
  - are encoded on a blockchain,
  - effectively removing the need for a central governing authority
  - -> hence the descriptors “decentralized” and “autonomous.”

The mother of all smart contract hacks

- Date: June 17, 2016
- Damage then: 3.6 million ether, US $50 million

- The DAO was a decentralised venture capital fund where investments were based on votings by the community
- On June 17, 2016 the DAO was attacked by unknown hackers, exploiting a combination of vulnerabilities in the DAO smart contract

The mother of all smart contract hacks

- The heart of the problem
  - an external “call” present in a codepath that should have been possible to execute only once per stakeholder (address)
  - However, it was positioned before the part of the code that revoked the stakeholder’s right to trigger the codepath after first execution
- Result
  - This allowed the hacker to use the external call to execute the codepath recursively multiple times, thereby draining much more funds from the contract than should’ve been possible

The mother of all smart contract hacks

- The hacker obtained **3.6 million ether**
  - Ether: the crypto currency of Ethereum
  - $50 million USD at the time of the attack

- What to do?

The mother of all smart contract hacks

- Due to the nature of the Child-DAO’s, the funds were locked for 28 days,
  - giving the community the chance to hard fork the Ethereum chain to refund the investors.
- This move sparked a highly controversial debate in the community
- After this ideological earthquake, a lot of investors argued that the immutability of the blockchain as sacrosanct and decided to still mine on the old chain resulting in the continuation of the legacy chain as Ethereum Classic
- Afterwards, the DAO was liquidated and investors were refunded on the Ethereum chain

Lessons Learned

- Even blockchains can be modified
  - If more than 51% of the voting power agrees
  - Trust problem

- Be very careful when coding blockchain contracts

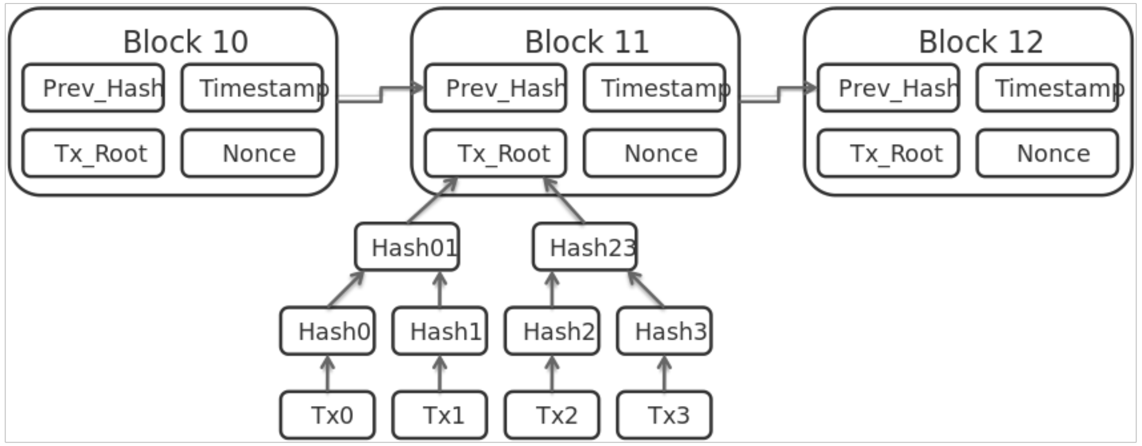
Today

- Smart Contracts
- Ethereum
- Hyperledger

ETHEREUM

Ethereum: Concept

- It is a blockchain… like Bitcoin and others…



… but…

With key additions

- A built-in programming language
- Two types of accounts
  - User accounts (controlled by private keys)
  - Contracts (controlled by code)
- Anyone can create an application with any rules by defining it as a contract

Ethereum

- Open Source
  - Like Bitcoin, Ethereum is a public blockchain no one controls or owns.
- Proof of Work
- Ethereum Virtual Machine
  - Transactions are more than just values, but "Turing" complete programs that run when blocks are processed by nodes
  - -> smart contracts

Ethereum

- General Purpose
  - Supports Bitcoin-like value transactions (e.g. send 10 ETH from account A to B)
  - or more complex applications (smart contracts)
- Smart Contracts
  - Turing complete languages (e.g. Solidity)
- Decentralized Apps (DApps)
  - GUIs for smart contracts allow users to interact in ways similar to web 2.0 (HTML/JS/CSS)

Blocks

- Ethereum: one block roughly every 12sec
  - Vs. Bitcoin: one block roughly every 10minutes
  - Reduces wait time until transactions are included in blockchain
  - More transactions per minute

Hash

- Ethash: Designed to be ASIC resistant
  - No benefits from implementing in ASIC (custom hardware)
  - Avoids mining farms as on Bitcoin scale
- But this changed spring 2018
  - Community is unsure on how to proceed
  - Whether to change to a new hash algorithm or not

Transactions and Account

- Two types of Accounts
  - Externally owned accounts
    - controlled by people/keys similar to Bitcoin
  - Contract accounts
    - controlled by smart contract code

Ethereum Virtual Machine (EVM)

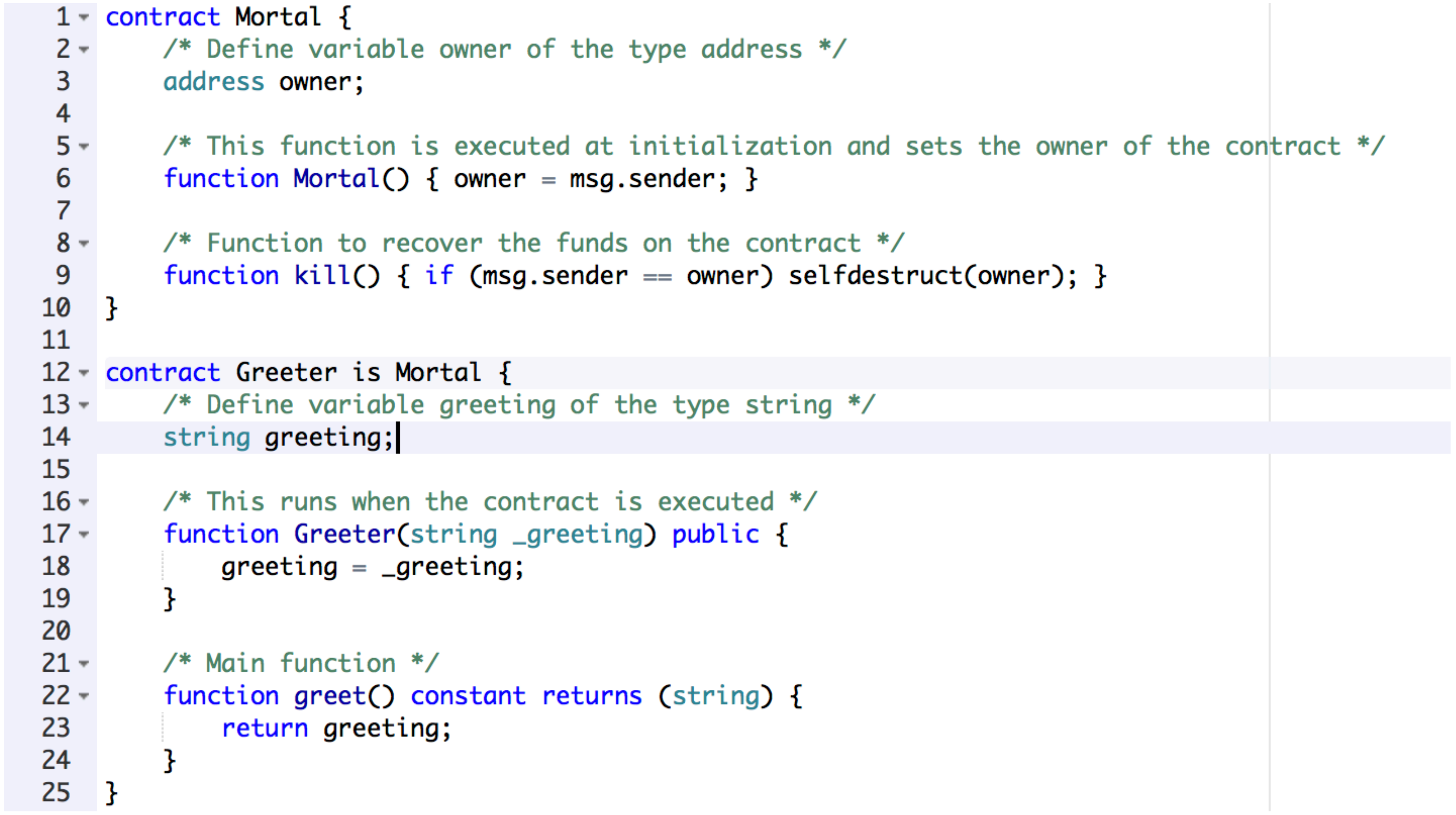
- A virtual machine, similar to Java VM etc.
- Just that code is executed ”everywhere”
- With
  - Stack
  - Memory
  - Storage
  - Environment variables
  - Logs
  - Sub-calling (“sub processes”)

Ethereum Virtual Machine (EVM)

- Low-level byte code language
  - Based on: Bitcoin Script, traditional assembly and Lisp
  - Compare: Java Byte Code

- High-level languages Compile to EVM code
  - Serpent
  - Solidity
  - LLL

Hello World (Solidity)



```
contract Mortal {
    /* Define variable owner of the type address */
    address owner;

    /* This function is executed at initialization and sets the owner of the contract */
    function Mortal() { owner = msg.sender; }

    /* Function to recover the funds on the contract */
    function kill() { if (msg.sender == owner) selfdestruct(owner); }
}

contract Greeter is Mortal {
    /* Define variable greeting of the type string */
    string greeting;

    /* This runs when the contract is executed */
    function Greeter(string _greeting) public {
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }
}
```

Another Smart Contract (Solidity)

```
contract MyToken {
    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    /* Initializes contract with initial supply tokens
       to the creator of the contract */
    function MyToken(uint256 initialSupply) {
        balanceOf[msg.sender] = initialSupply;// Give the creator all initial tokens
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) {
        if (balanceOf[msg.sender] < _value) throw;  //Check if the sender has enough
        if (balanceOf[_to] + _value < balanceOf[_to]) throw; // Check for overflows
        balanceOf[msg.sender] -= _value;                    // Subtract from the sender
        balanceOf[_to] += _value;                       // Add the same to the recipient
    }
}
```

What does this code do?

It is just code, most of you are able to understand and write this…

Ethereum Smart Contracts

- Uses blockchain to implement arbitrary social contracts without a central server

- Like Bitcoin accounts with balances
- Unlike Bitcoin accounts can be contracts
  - Sending money to a contracts activates the contract (executes its code)

Ethereum Smart Contracts

- Executed in the network
  - On all the machines

- Problem?

```
function foo() {
   while (true) {
      /* Loop forever! */
   }
}
```

Ethereum Smart Contracts

- Problem?
  - This takes computing power
  - Bugs

- What do we do?
  - We pay the ones executing our smart contracts

Gas

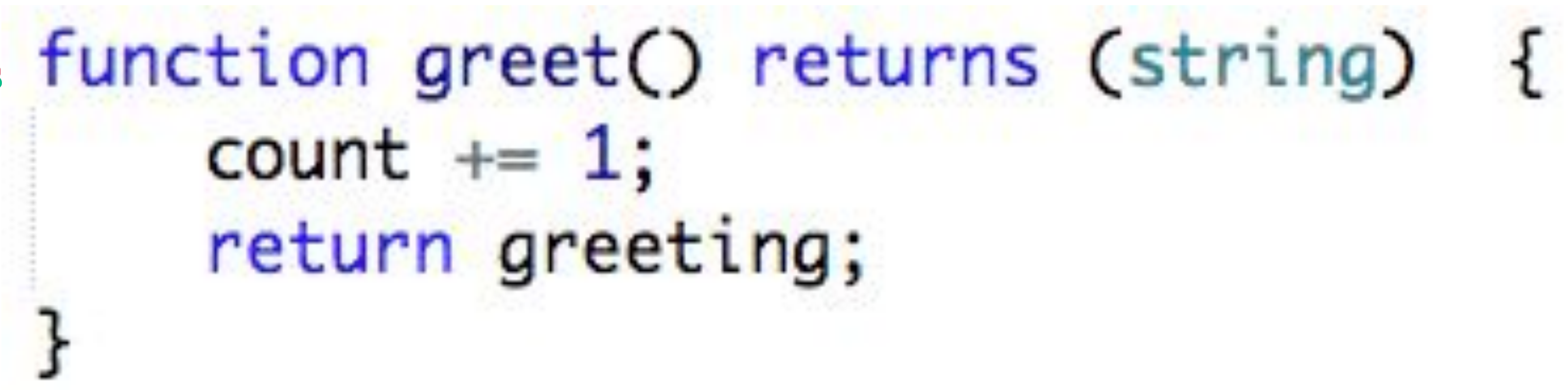
- Halting problem
  - Cannot tell whether or not a program will run infinitely
- Solution: charge fee per computational step (“gas”)
- Special gas fees also applied to operations that take up storage

Gas

- Every contract requires gas to execute
  - Every EVM-opcode consumes gas
- Every transaction specifies
  - Startgas: The maximum amount of gas it is willing to consume
  - Gasprize: The fee in ether it is willing to pay per unit of gass

Gas

- Total gas needed for **ADD**: 20
- startGas: 15

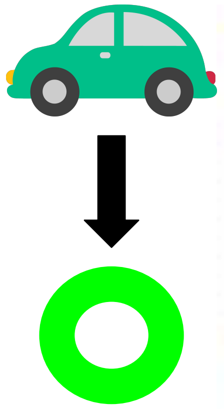


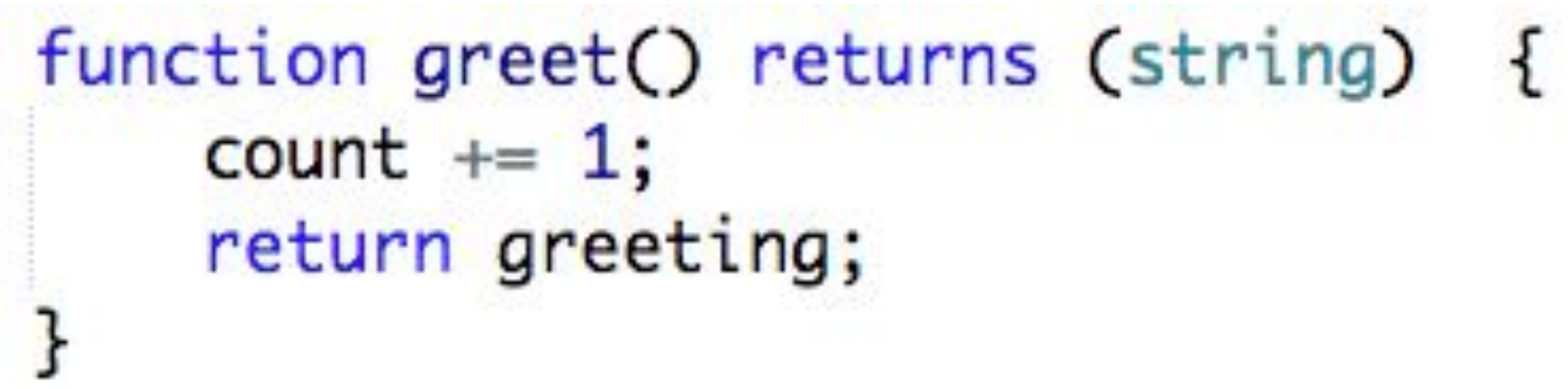
```
function greet() returns (string)  {
      count += 1;
      return greeting;
}
```

OUT OF GAS EXCEPTION

Gas

- Total gas needed for **ADD**: 20
- startGas: 21



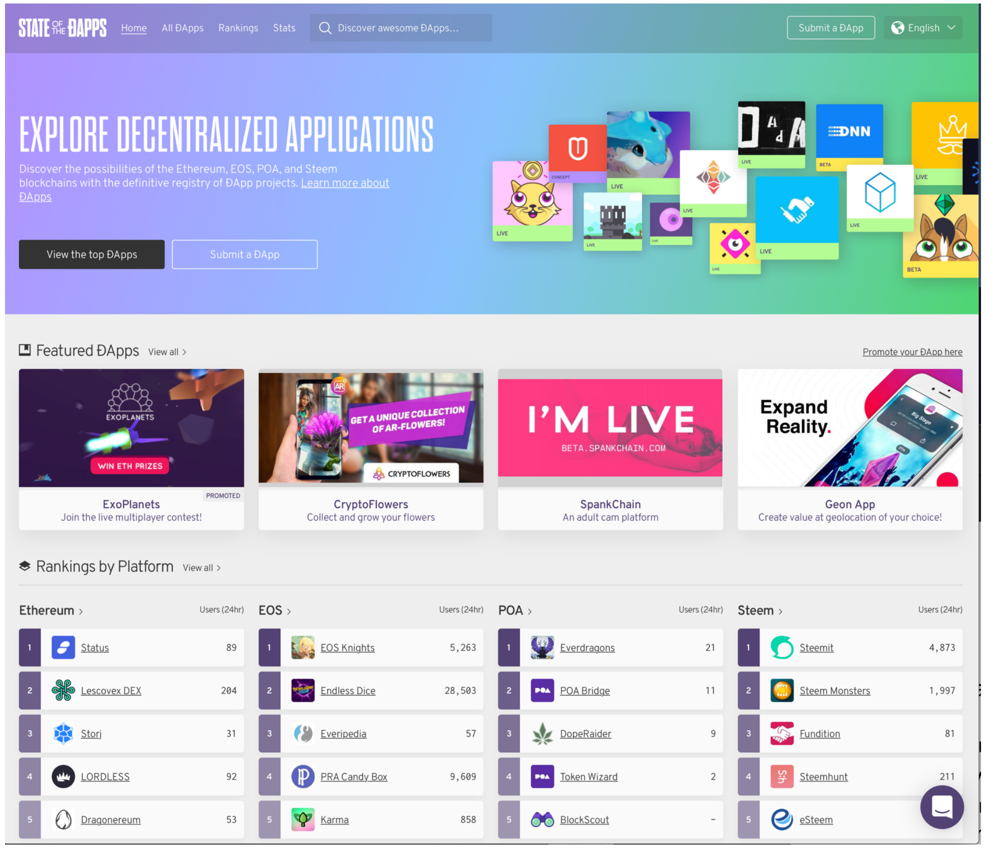


```
function greet() returns (string) {
        count += 1;
        return greeting;
}
```

‘greeting’ returned

App Store for Smart Contract Apps

- Called Distributed Applications (dApps)



EXPLORE DECENTRALIZED APPLICATIONS

Discover the possibilities of the Ethereum, EOS, POA, and Steem blockchains with the definitive registry of ĐApp projects. Learn more about ĐApps

Featured ĐApps View all

ExoPlanets — Join the live multiplayer contest! (PROMOTED)

CryptoFlowers — Collect and grow your flowers

SpankChain — An adult cam platform

Geon App — Create value at geolocation of your choice!

Rankings by Platform View all

Ethereum	Users (24hr)	EOS	Users (24hr)	POA	Users (24hr)	Steem	Users (24hr)
1 Status	89	1 EOS Knights	5,263	1 Everdragons	21	1 Steemit	4,873
2 Lescovex DEX	204	2 Endless Dice	28,503	2 POA Bridge	11	2 Steem Monsters	1,997
3 Storj	31	3 Everipedia	57	3 DopeRaider	9	3 Fundition	81
4 LORDLESS	92	4 PRA Candy Box	9,609	4 Token Wizard	2	4 Steemhunt	211
5 Dragonereum	53	5 Karma	858	5 BlockScout	-	5 eSteem	

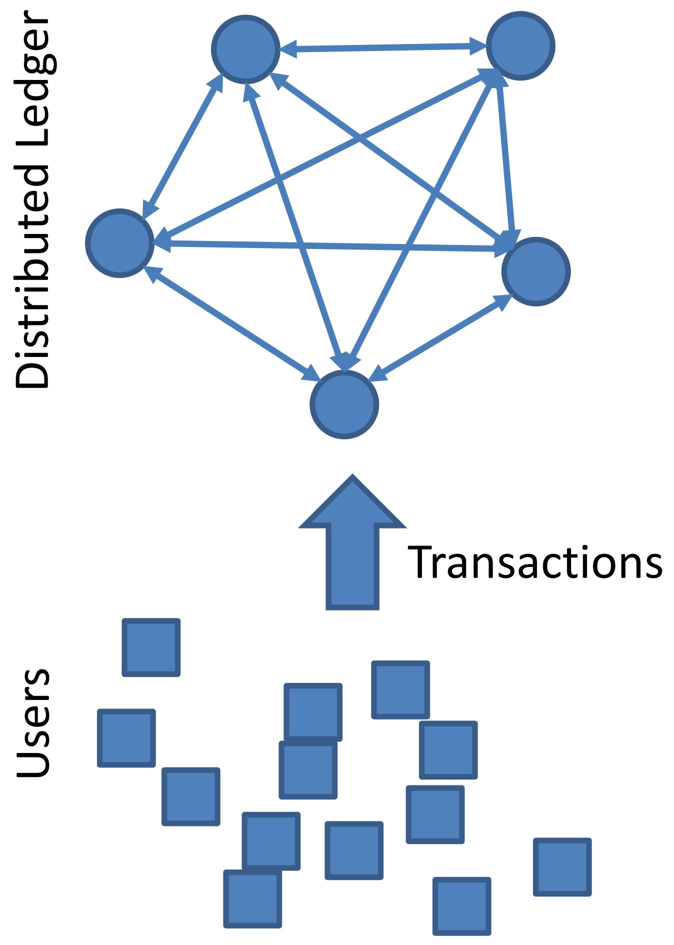
https://www.stateofthedapps.com

A Private and Permissioned Blockchain

HYPERLEDGER

Motivation: Blockchains High Level

- Distributed Ledger
  - Nodes do not trust each other
  - Need to avoid double spending etc
    - Nobody should get 51% of the voting power
    - -> Solution: Mining
- What if…
  - Nodes maintaining the ledger trust each other?



If Nodes maintaining the ledger trust each other?

- If Nodes maintaining the ledger trust each other?
  - 51% problems goes away
  - No need for miners
- But?
  - We have to trust the ledger
  - Not everyone can participate in maintaining ledger
    - Needs permission
  - Still can verify that nobody changed it by keeping a copy of the ledger
  - > Private or permissioned blockchains

Public Blockchains

- Anyone in the world can read this kind of blockchain
- Anyone can make transactions to the blockchain
- Anyone can participate in the consensus
- Examples of public (permissionless) blockchain
  - Bitcoin
  - Ethereum

Private Blockchains

- Right to read the blockchain might be restricted
- Right to make transactions to the blockchain is restricted to trusted party
- Consensus process is handled by selected nodes
- Often permission based
  - Need to be enrolled to do transactions
  - Admin can give permission to people who are in network to approve new people
- Examples of private blockchains
  - Hyperledger Fabric
  - Corda

Now

- Discuss a private blockchain: Hyperledger Fabric
  - In contrast to Bitcoin and Ethereum
    - The public blockchains we discussed previously

Hyperledger foundation

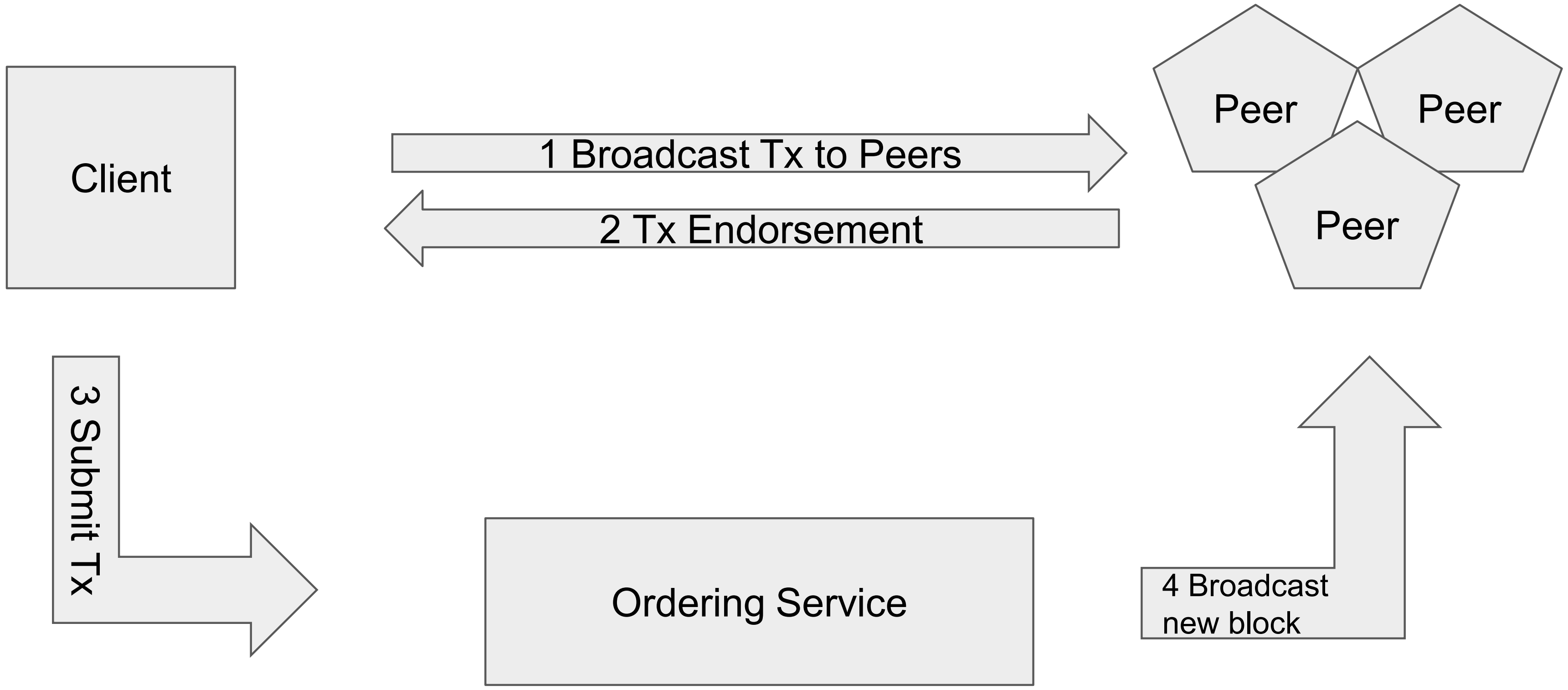
- Founded in 2015 by Linux Foundation
- Sole purpose of Hyperledger is to advance blockchain technology.
- Hyperledger is not a single standard, it is a collaboration of the community to create different blockchain technologies

HYPERLEDGER

Hyperledger Fabric

- One of the project from the Hyperledger Foundation
- Private (or Permissioned) blockchain
- Elastic and extensible architecture

Transaction Flow



Hyperledger Fabric Model

- Nodes
  - Peer nodes: execute smart contracts, interface with applications
  - Orderer nodes: ensure the consistency of the blockchain
- Transactions
- Membership Service Provider
  - Permissions nodes
- Channels
  - Private subnet between members
- Chaincode
  - Code for smart contracts

Consensus in Private Blockchains

- Still: some nodes may be malicious
  - Bugs
  - Attacks
- Need a robust consensus algorithm
  - Lecture content?
  - Byzantine generals?
  - Practical Byzantine Fault Tolerance (PBFT)
    - Similar to Byzantine generals
    - If we have time in one of the next lectures

Public Blockchains: Pros/Cons

- Pros
  - Public blockchains are open, everyone can participate
  - Developers have little power of changing the rules of the application
- Cons
  - High transactions fees
  - Probability of 51% attack
  - Cost of mining?

Private Blockchains: Pros/Cons

- Pros
  - Consortium can easily change the rules of a blockchain, revert transactions. Etc.
  - Validators are known in the network
  - Transactions are cheaper
  - Privacy
- Cons
  - It may be considered more centralized than decentralized.
  - More suited for Enterprise
  - Identifies are known

Summary Hyperledger

- Blockchain
  - Permissioned / private
    - Good for business applications
    - People inolved are known anyway
  - No mining overhead

Discussion Blockchain

- Parties who may not trust each other still need to engage in transactions
  - So, they commonly use a Trusted Third Party (TTP) to maintain a single ledger of the transactions.
  - The TTP takes time, typically days
  - The TPP charges a fee per transaction
- Blockchain removes need for this middle man
  - But transactions still cost a fee
- Your view?

Your turn

- A smart contract is ...
- Who executes a smart contract?
- Ethereum Virtual Machine?
- Who pays for smart contract invocation?
- What is a dApp?
- What powers the
- Proof of work is not required in private blockchains
- Public Blockchains

https://olafland.survey.fm/blockchain-iii

- Questions?

- Inspired from / based on slides from
  - Jason Madden, Mehmet H. Gunes
  - Johannes Kofler
  - Alex Fisher
  - Adrian Hetman
  - And many others