HS18

DISTRIBUTED SYSTEMS ADVANCED & BLOCKCHAIN

Repetition VSS

T. Bocek

thomas.bocek@hsr.ch

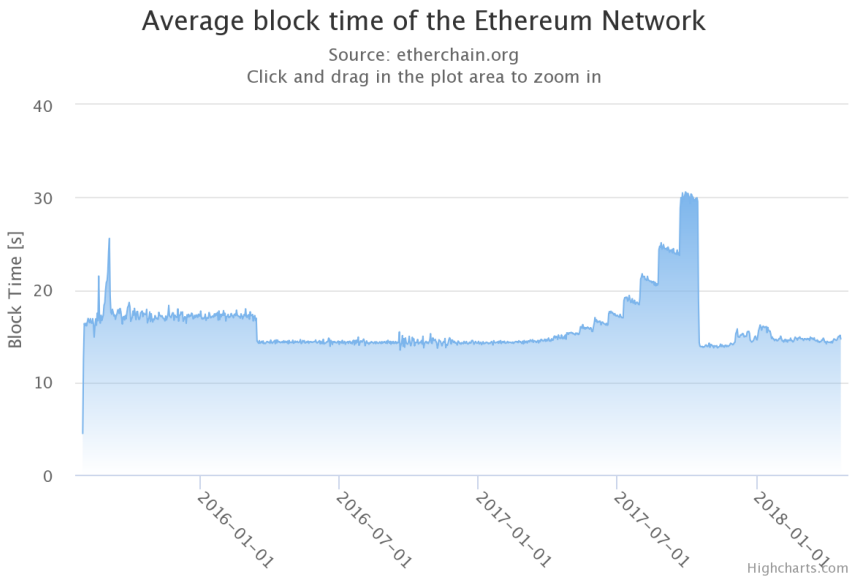
HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Distributed Systems - In the *News*

- **Announcement 20.09.2018: Trust Square Open Doors**
  - From 14:00 – 18:00
  - HSR is a partner of Trust Square
- **14.09.2018: The $1 Billion Tezos Blockchain Is Officially Launching Monday**
  - "Tezos is now fully operational without such a kill switch or other forms of centralised control."
  - Coinmarketcap: #16 ~990m USD
  - Largest ICO in July 2017 – ~230m USD
  - Contract issues: Johann Gevers vs. Breitman
    - "…granting himself excessive compensation…"
  - Tezos: proof-of-stake system and supports Turing complete smart contracts

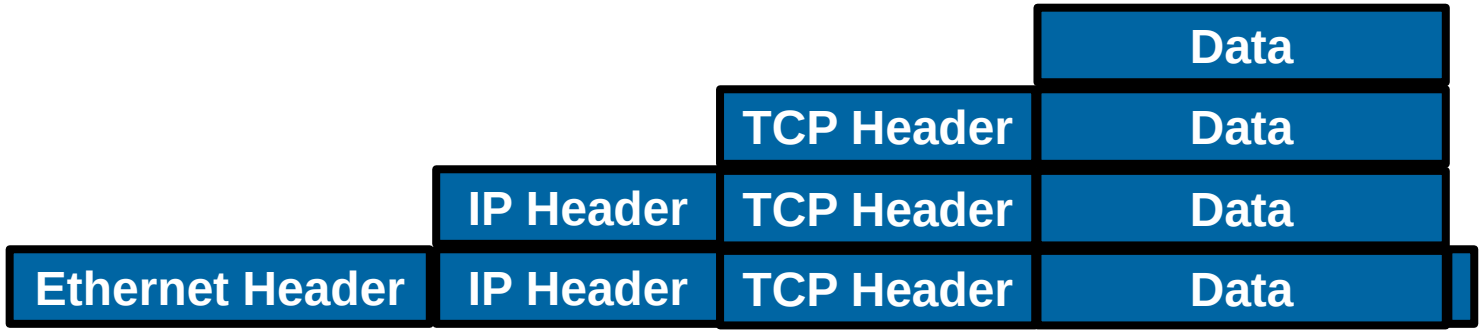
- **14.09.2018: Ethereum's Constantinople Hard Fork to Activate on Testnet in October**
  - EIP 145: efficient bitwise shifting
    - var shifted = i * 2 ** n; var shifted = i / 2 ** n;
  - EIP 1052: keccak256 hash of contract bytecode
  - EIP 1283: gas metering on SSTORE
  - EIP 1014: better support: state channels
  - EIP 1234: reward 3 → 2 ETH, diff. bomb 1y later



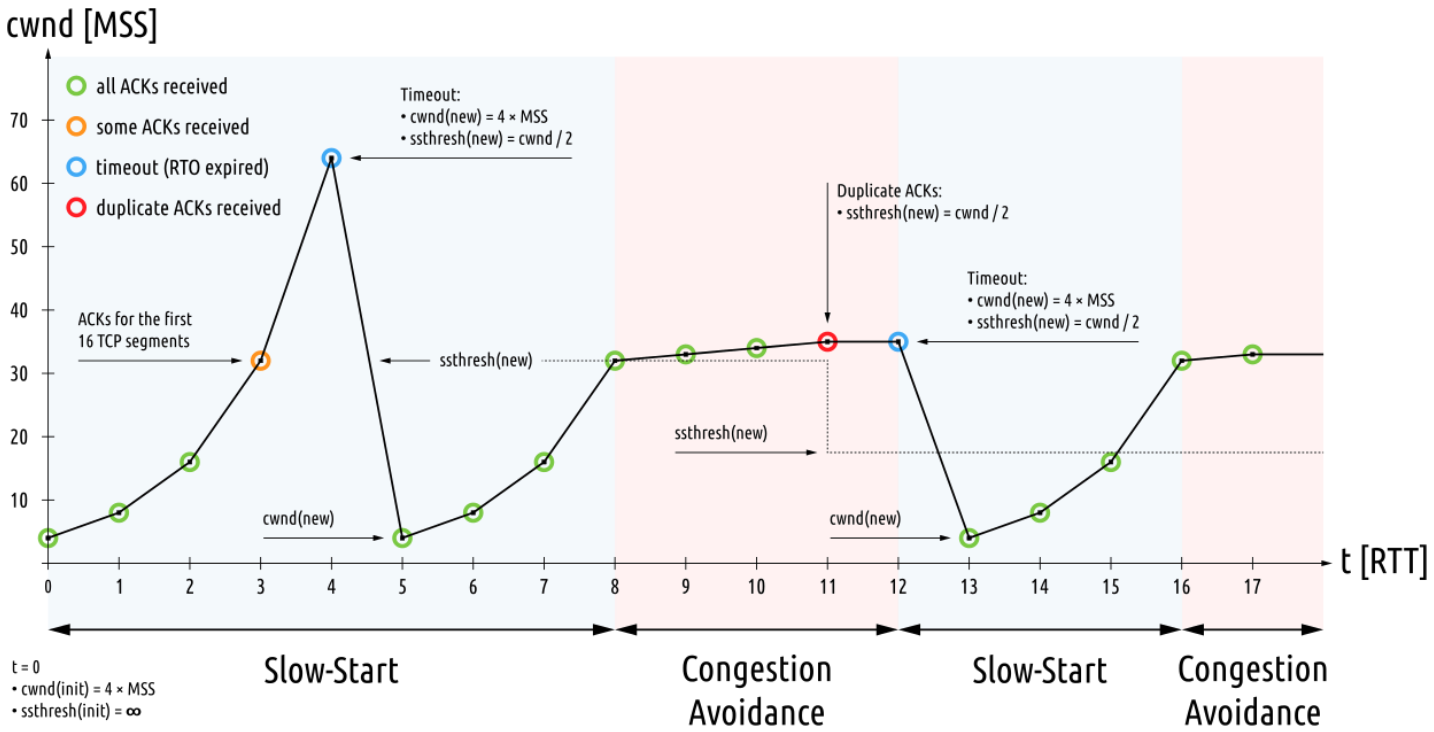
VSS W08: Networking: Layers

- **Networking: Each vendor had its own proprietary solution - not compatible with another solution**
- **Nowadays most vendors build compatible networks hardware/software from different vendors**
- **Goal of layers: interoperability**
- **1984: ISO 7498 - The Basic Reference Model for Open Systems Interconnection**

OSI model	"Internet model"
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Link
Pysical	

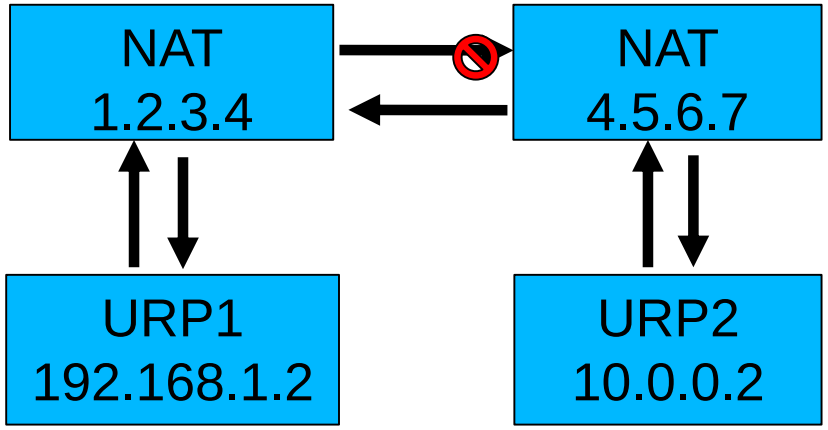


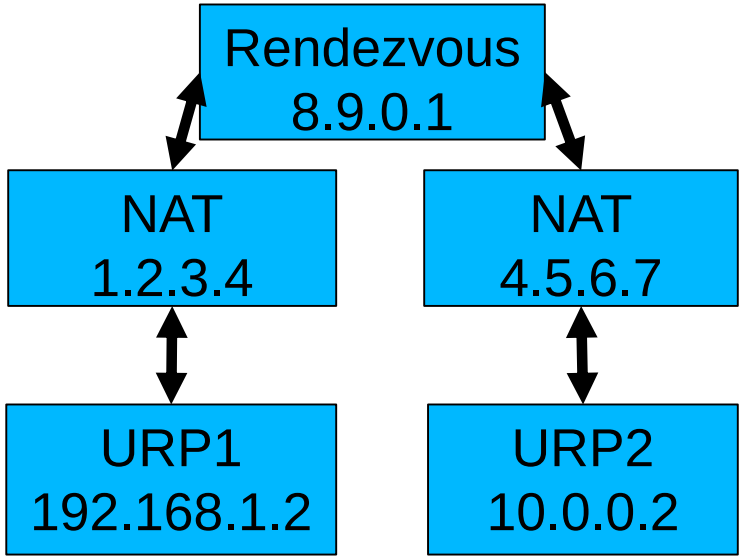
- **Flow control**
  - Sender is not overwhelming a receiver
  - Back pressure
  - Sliding window:
    - Receiver specifies the amount of additionally received data in bytes that can be buffered
    - Sender up to that amount of data before ACK
- **Congestion control**
  - slow-start
  - congestion avoidance
- **Difference flow/congestion control**
- **Wireshark - example**



https://upload.wikimedia.org/wikipedia/commons/thumb/2/24/TCP_Slow-Start_and_Congestion_Avoidance.svg/1280px-TCP_Slow-Start_and_Congestion_Avoidance.svg.png

- **Hole punching**
  - URP1 got 4.5.6.7:5000, URP2 got 1.2.3.4:4000
  - Unreachable peer 1 request to NAT 4.5.6.7, will fail – no mapping, however, unreachable peer 1 creates mapping with that request
  - Unreachable peer 2 sends request to unreachable peer 1 (1.2.3.4:4000) success!





Mapping for NAT 1.2.3.4 (Unreachable peer 1)			
192.168.1.2:4000	4.5.6.7:5000	4.5.6.7:5000	1.2.3.4:4000

Mapping for NAT 4.5.6.7 (Unreachable peer 2)			
10.0.0.2:5000	1.2.3.4:4000	1.2.3.4:4000	4.5.6.7:5000

- **Simple Distributed Architecture**
  - Call a function/procedure on another machine
  - Often: wait for result (synchronous)
- **Already solved?**
  - SOAP – lots of XML
  - CORBA - heavyweight
  - DCOM, COM+ - mostly windows
  - RMI – mostly Java
  - JSON, XML – no protocol description
- **Goal**
  - Language and platform neutral way for serializing structured data for communication protocols
  - Simplicity!
- **Binary vs. Text**
  - JSON: {“account_fee”:”1234”} – 22 bytes
    - Parsing needs time
    - Human readable – easy to debug
  - Binary: 0x2, 0x1234 – e.g. length+data – 3 bytes
    - No parsing
    - Needs a protocol, error-prone
- **Protocol Buffers, Apache Thrift, Apache Avro**
  - Interface Description Language (IDL)
  - Versioning
  - Binary format (performance)

URL: b.socrative.com/student/

Room: HSR

VSS W09: Comparison of Lookup Concepts

System	Per Node State	Communi-cation Overhead	Fuzzy Queries	No false negatives	Robust-ness
Central Server	O(N)	O(1)	✓	✓	✗
Flooding Search	O(1)	O(N)	✓	✗	✓
Distributed Hash Tables	O(log N)	O(log N)	✗	✓	✓

VSS W09: Kademlia Example

- **2^3, max size 8, #6 searches for 3**
  - Neighbors of 6, if k=1

1	2	3
7	4 (or 5)	0 (or 1, 2)

Routing Table of #6
6 xor 3 = 101b

  - Search for 3, ask 0, neighbors of 0

1	2	3
1	2	4 (or 5, 6, 7)

Routing Table of #0
0 xor 3 = 11b

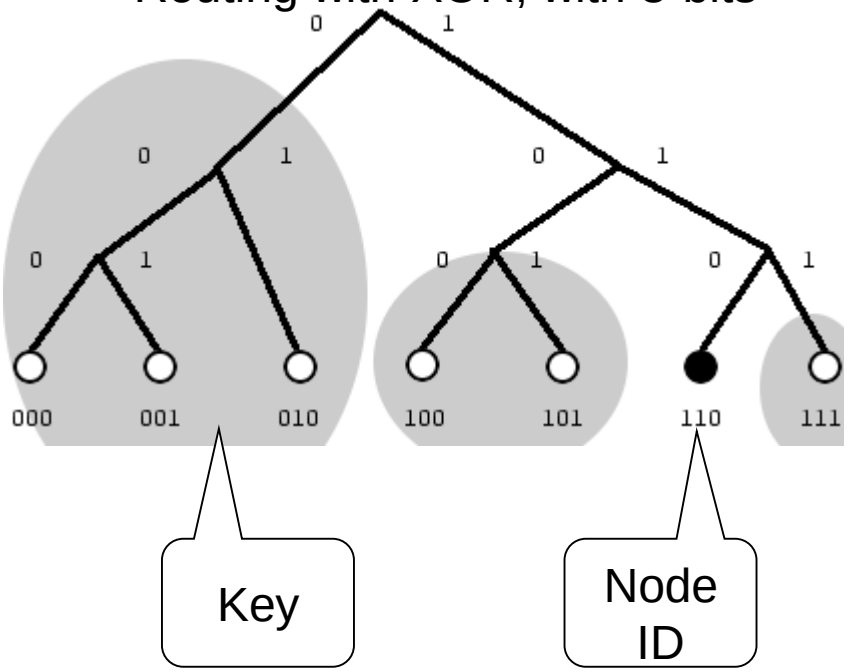
  - Ask 2, neighbors of 2

1	2	3
-	0 (or 1)	4 (or 5, 6, 7)

Routing Table of #2
2 xor 3 = 1b

  - Ask 2, 2 replies 0. 6 figures that there is no closer node, 2 is the closest one (2 xor 3 =1)

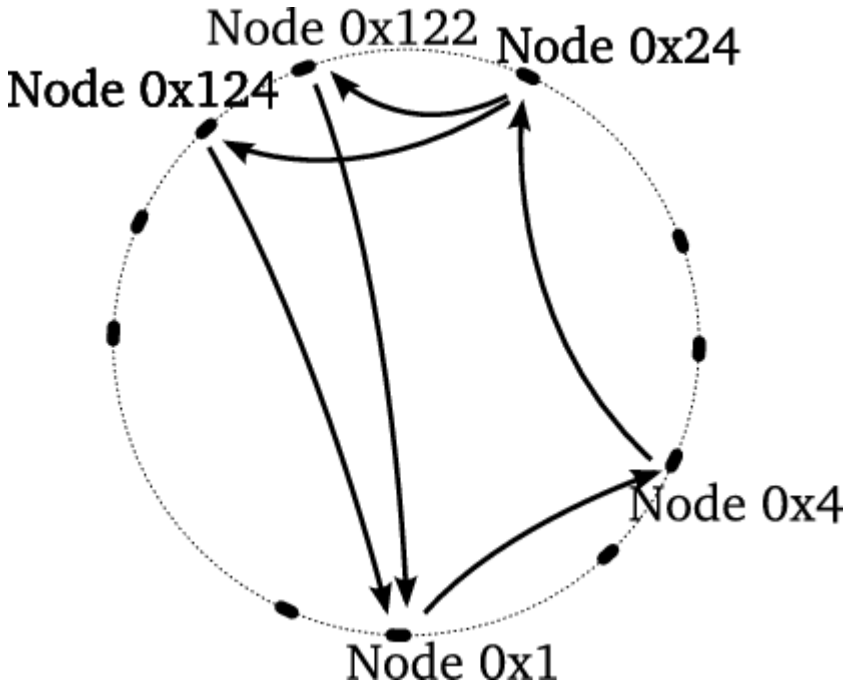
Routing with XOR, with 3-bits



URL: b.socrative.com/student/

Room: HSR

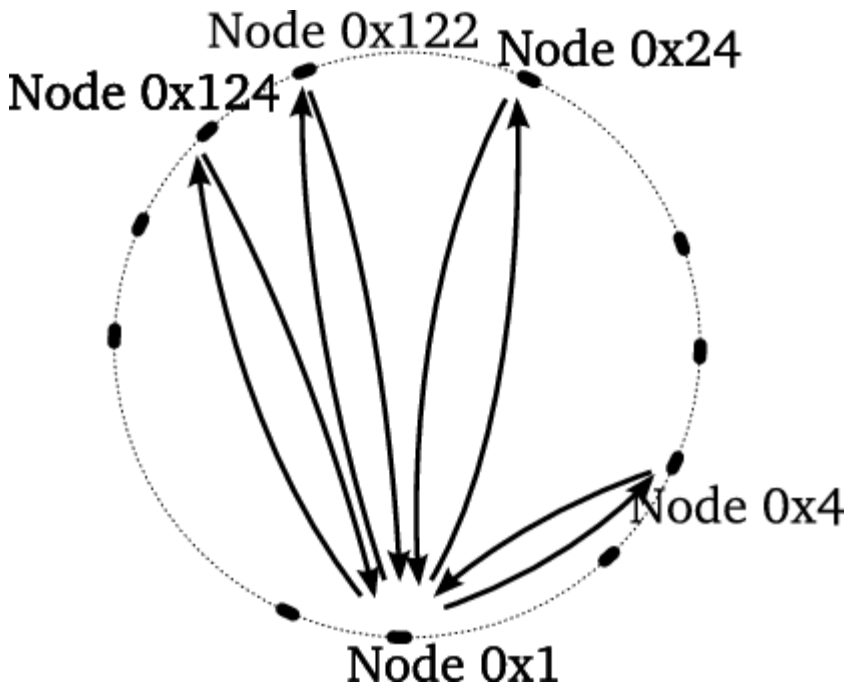
- **Recursive routing**



+ online status update

- faulty peers cause delay

- **Iterative routing**

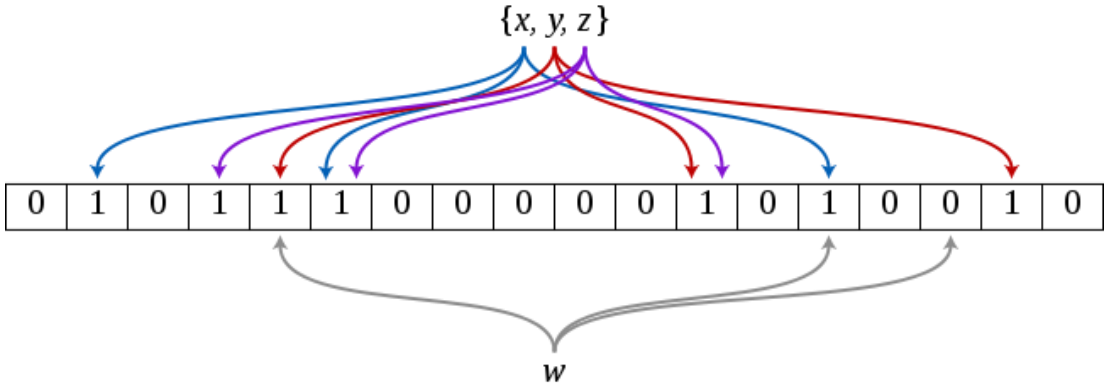


+ control

- neighbor maintenance

VSS W10: Query of an Element, m=18, k=3

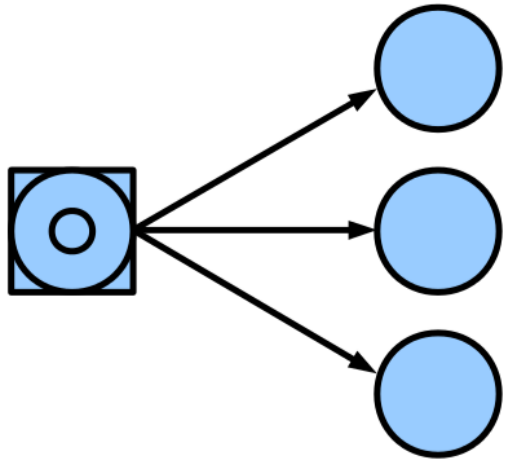
- **Insert x, y, z**
- **Query w**

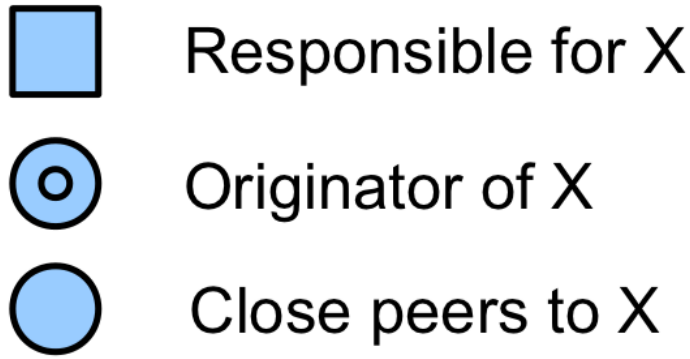


http://en.wikipedia.org/wiki/Bloom_filter

- **Example for False-positives**
  - Insertions
    - Hash („color printer“) => (1,4,6)
    - Hash („digital camera“) => (3,4,5)
    - Bloom filter (1,3,4,5,6)
  - Query
    - Hash („heat sensor“) => (3,4,6)
    - Matches since bits 3,4,6 are all set to 1
  - Online
- **False-negative**
  - Query
    - Hash (“color printer”) => (1,4,6) , matches (1,3,4,5,6) → no false-negative

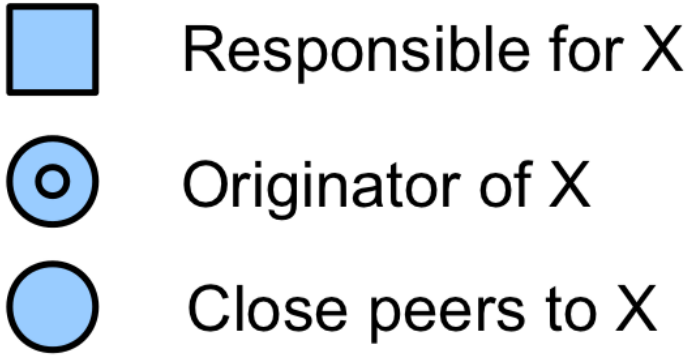
- **Direct Replication**
  - Enough replicas
    - Originator peer is responsible
    - Periodically refresh replicas
    - Example: tracker that announces its data

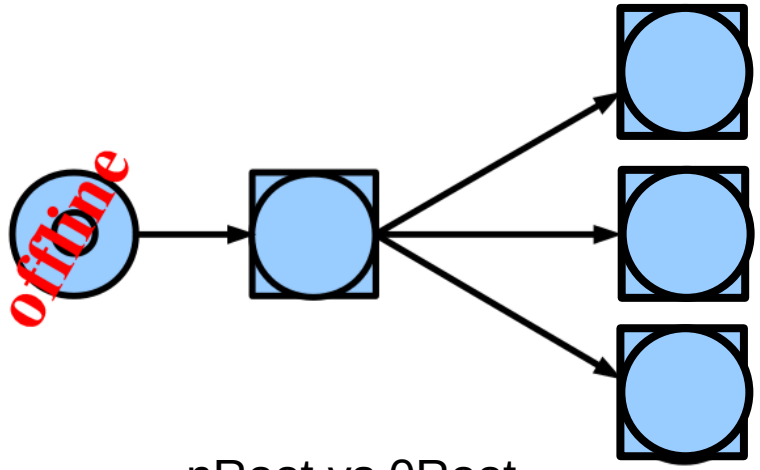




- **Problem**
  - Originator offline → replicas disappear. Content has TTL

- **Indirect Replication**
  - The closest peer is responsible, originator may go offline (0Root)
    - Periodically checks if enough replicas exist
    - Detects if responsibility changes





nRoot vs 0Root

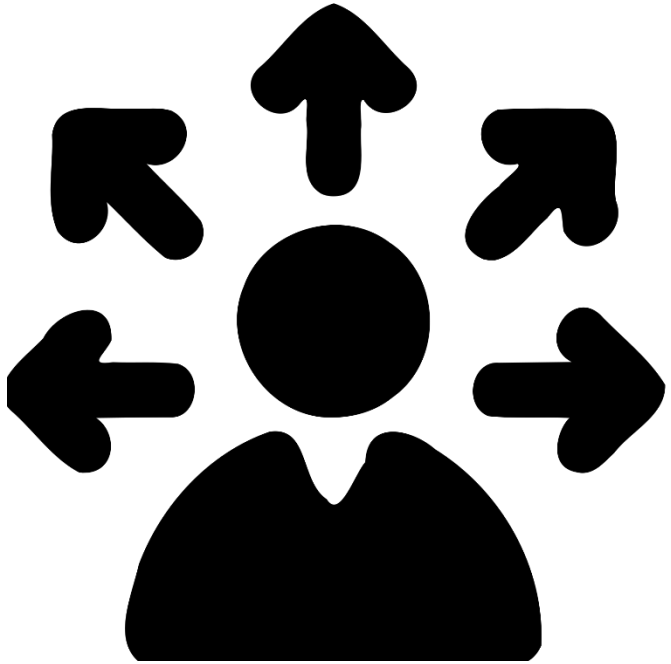
- **Problem**
  - Requires cooperation between responsible peer and originator
  - Multiple peers may think they are responsible for different versions → eventually solved

URL: b.socrative.com/student/

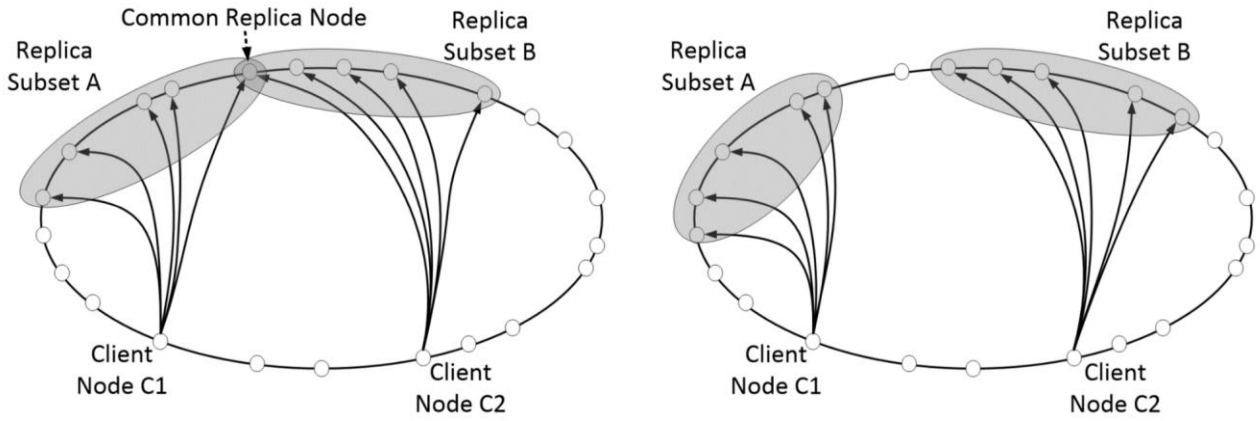
Room: HSR

VSS W11: Consistency

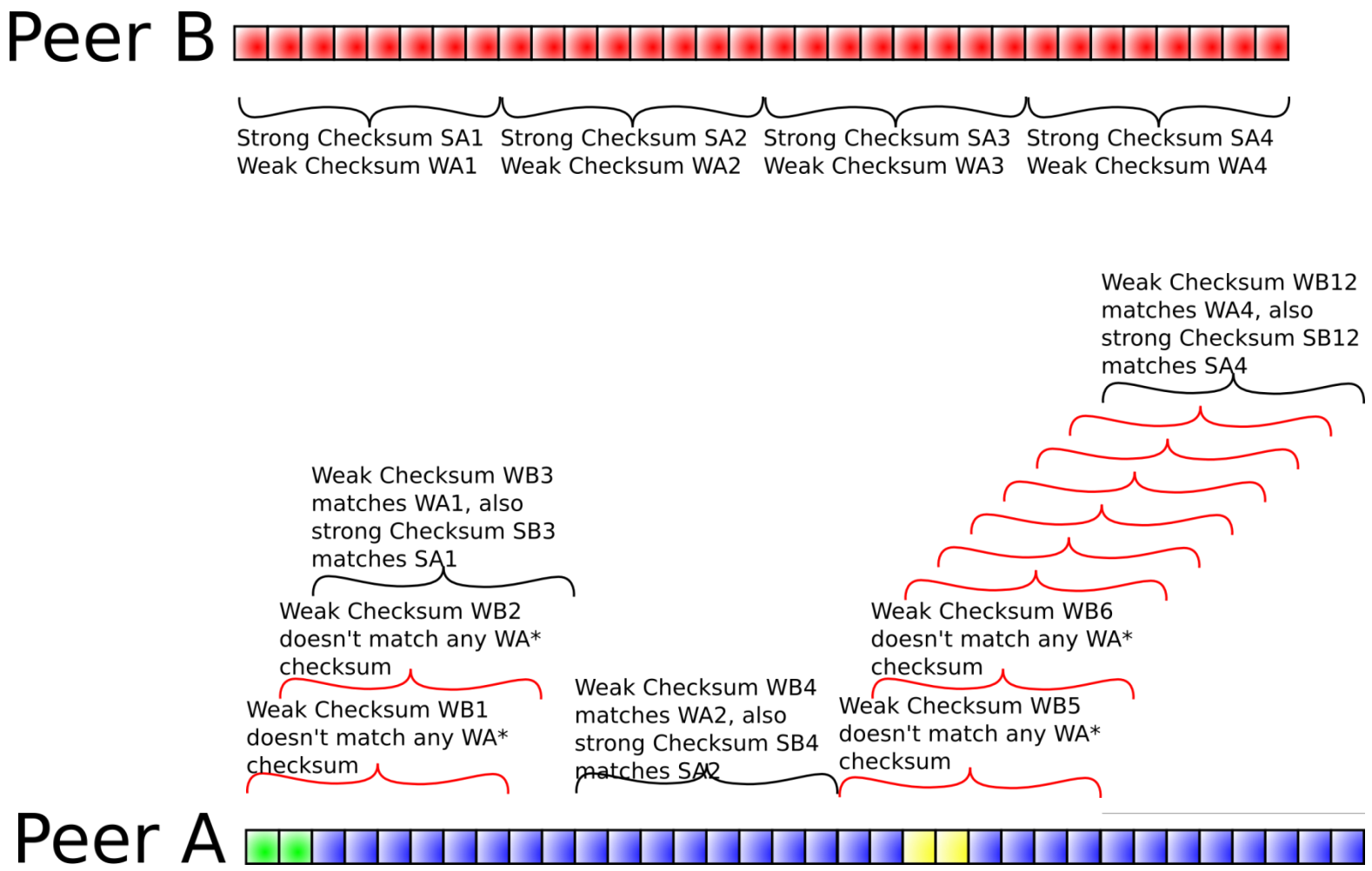
- **DHTs have weak consistency**
  - Peer A put X.1, Peer B gets X.1 modifies it puts B.2
  - Same time: Peer C gets X.1 modifies it puts C.2
    - Which one is stored B.2 of B or C.2 of C?
- **Consistency generic issue in distributed systems**
  - Coordinator required:
    - easy solution: centralized
    - Interesting solution: decentralized, in case failed peer, pick another peer
- **Coordinator needs to be defined**
  - Election, example Paxos



- **Consistency in DHTs – vDHT**
  - Paxos and DHTs [1], [2]
  - vDHT: CoW, versions, 2PC, replication, software transactional memory (STM) → for consistent updates. Works for light churn
    - No locking, no timestamps (replication time may have an influence)
    - Every update – new version
      1. get latest version, check if all replica peers have latest version, if not wait and try again
      2. put prepared with data and short TTL, if status is OK on all replica peers, go ahead, otherwise, remove the data and go to step 1.
      3. put confirmed, don't send the data, just remove the prepared flag
- **In case of heavy churn, API user needs to resolve**

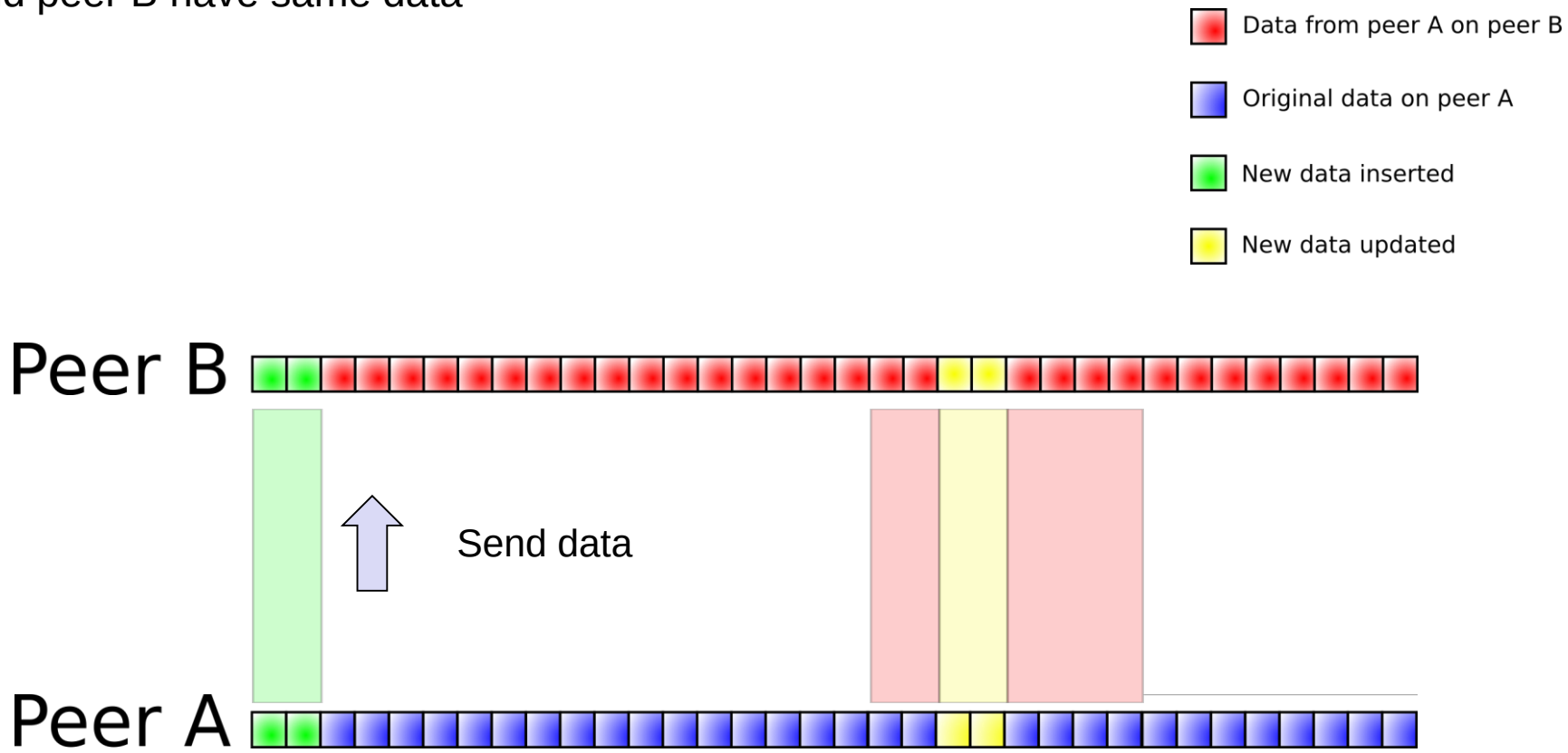


VSS W11: Rsync - Example



VSS W11: Rsync - Example

- **Peer A sends 2 + 8 blocks to peer B**
  - Peer A and peer B have same data



URL: b.socrative.com/student/

Room: HSR

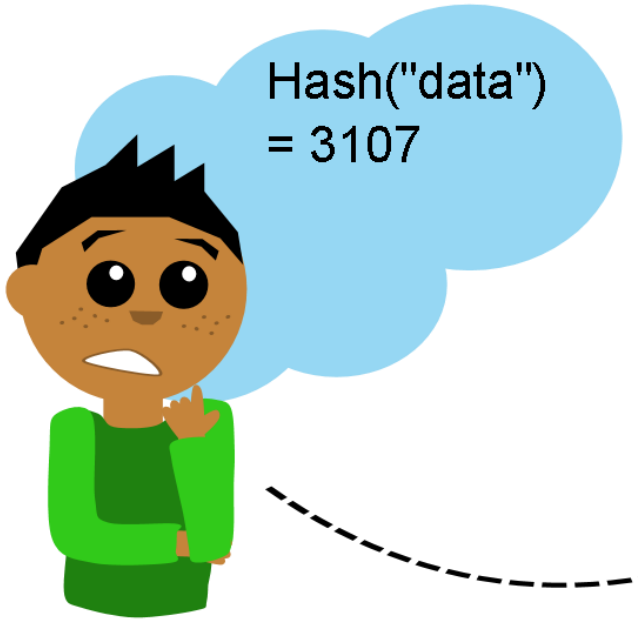
Reminder: register here for the challenge task

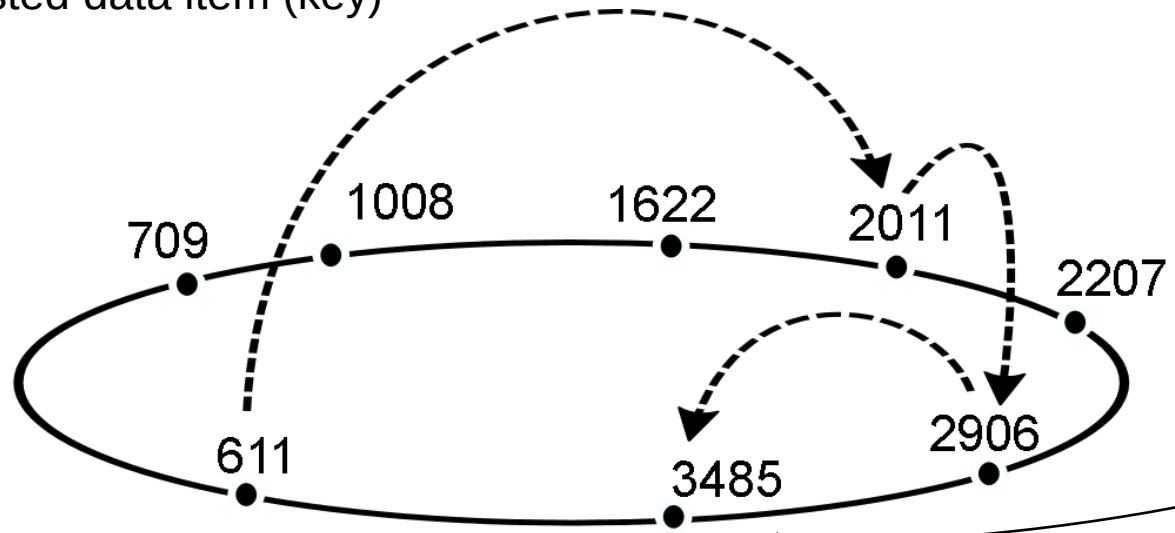
DSA Administration / “Mini” Project / Challenge Task

- **Each student's group is free to decide about specifics of building the P2P messaging application and how to organize the information within the distributed network.**
- **Requirements**
  1. Full decentralization and P2P mechanisms
  2. Send messages between peers
  3. Quickly and easily verify messages
  - The solution may use existing libraries and code, but those must be allowed to be published under an open software license.
  - The final presentation shall outline the application and architecture.
- **Requirements not met = ~cannot write exam**

- **If you want to use other frameworks libraries, please do so!**
  - Main goal: learn **hands-on** distributed systems and blockchain
  - If you fail but can show why libraries or tools do not work, you already learned a lot!
  - **Hands-on** learning is key, don’t be afraid if it won’t work. If you tried and can show it, we’ll take it into account.
  - You are encouraged to experiment with new technology!
  - I will present WebRTC in the upcoming lectures, as some students want to use it

- **Step 2: Locating the data / Routing to a K/V-pair**
  - Start lookup at arbitrary node of DHT
  - Routing to requested data item (key)





Node 3485 manages keys 2907-3485,

Initial node (arbitrary)

Key = H("*my data*")

(3107, (ip, port))

Value = pointer to location of data

VSS W09: Kademlia

- **Several approaches to build DHT**
  - Distance metric as key difference
  - Chord, Pastry: numerical closeness
  - Kademlia: XOR metric
- **Kademlia designed in 2002 by Maymounkov and Mazières**
  - Many implementations, application specific
    - Kad network: eMule and others
    - BitTorrent (tracker)
    - IPFS
    - Steem, Factom
- **Parallel queries**
  - For one query, α (alpha) concurrent lookups are sent
  - More traffic load, but lower response times
- **Each Kademlia node and data item has unique identifier**
  - 160 bit (SHA-1)
  - Nodes: Node ID (160bit)
    - Can be calculated from IP address or public key, and data item using secure hash function, or just random
  - Data items: Keys (160bit), hash of data item
- **Keys are located on the node whose node ID is closest to the key**
  - Kademlia: 160 buckets with size 20 (8)
    - In Kademlia: k contacts per distance 2^i to 2^(i+1)
  - Knows neighbors well, further nodes not that much
  - If distance can be represented in m bits, bucket m will be used

- **DHT-based overlay network using the XOR distance metric**
  - Simple operation
  - Symmetrical routing paths (A → B == B → A)
    - due to dXOR(A,B) == dXOR (B,A)

XOR Distance Calculation:

ID Node A: 110101
ID Node B: 010001

$d_{XOR}(A,B) = d(110101,010001)$

1 1 0 1 0 1
XOR
0 1 0 0 0 1
↓
1 0 0 1 0 0

$d_{XOR}(A,B) = 1\,0\,0\,1\,0\,0_2 = 36_{10}$

Kademlia Example

- **2^3, max size 8, #6 searches for 3**
  - Neighbors of 6, if k=1

1	2	3
7	4 (or 5)	0 (or 1, 2)

Routing Table of #6
6 xor 3 = 101b

  - Search for 3, ask 0, neighbors of 0

1	2	3
1	2	4 (or 5, 6, 7)

Routing Table of #0
0 xor 3 = 11b

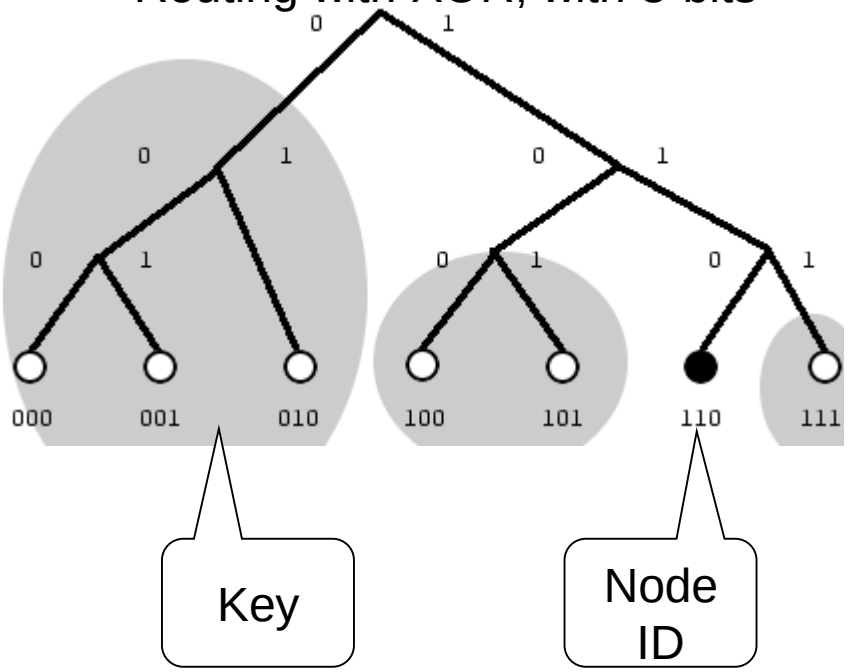
  - Ask 2, neighbors of 2

1	2	3
-	0 (or 1)	4 (or 5, 6, 7)

Routing Table of #2
2 xor 3 = 1b

  - Ask 2, 2 replies 0. 6 figures that there is no closer node, 2 is the closest one (2 xor 3 =1)

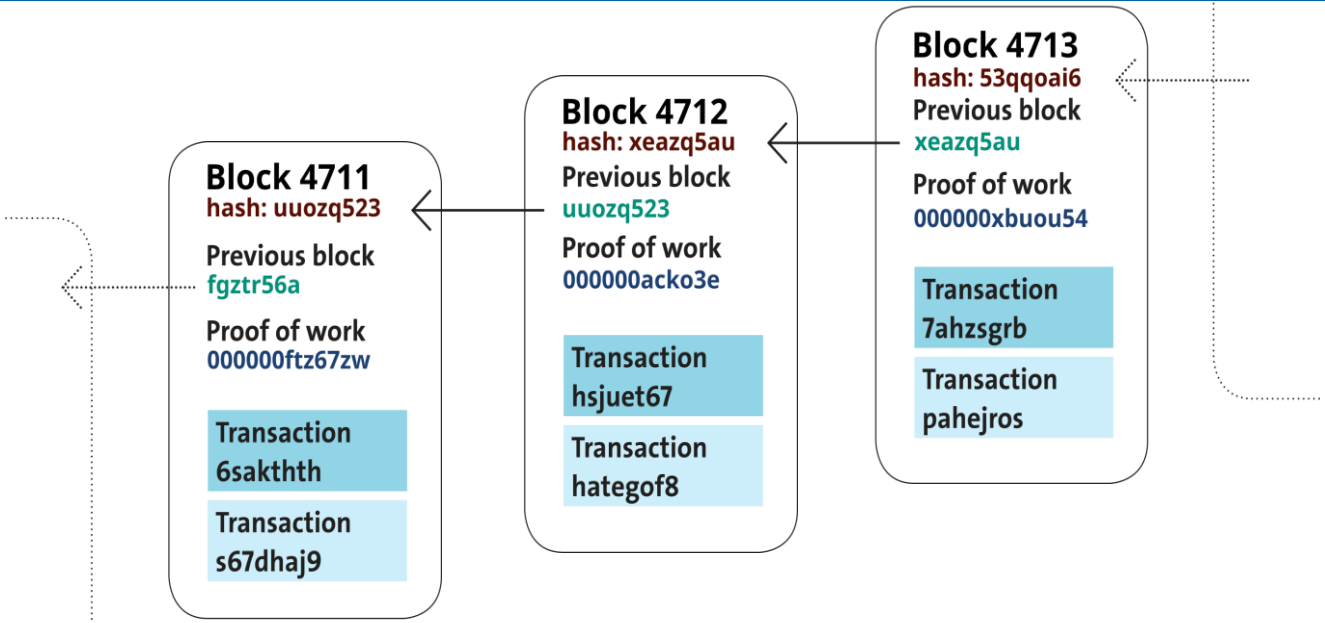
Routing with XOR, with 3-bits



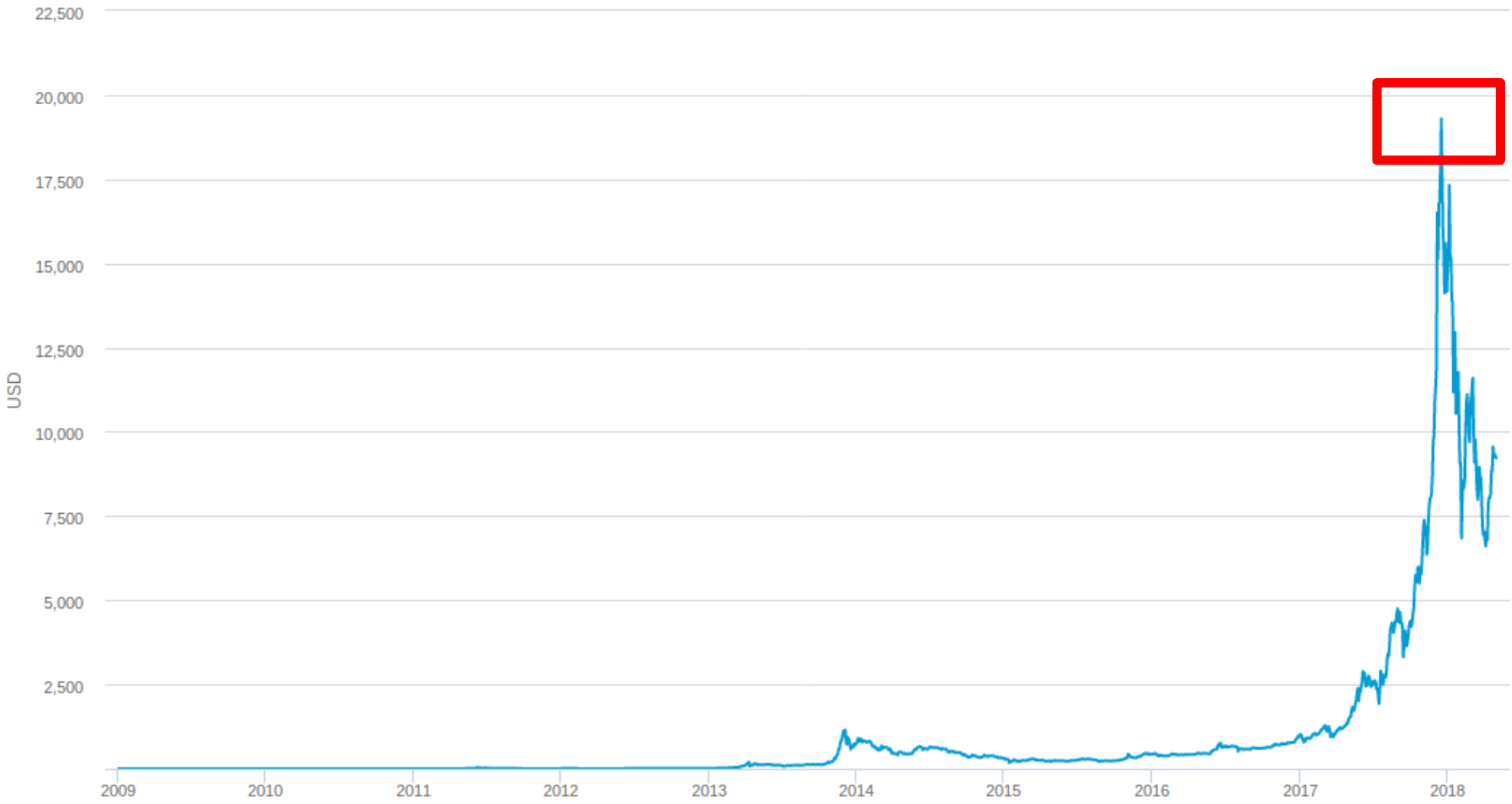
Kademlia exam question discussed in the exercises

VSS W12: Building Block

- **Hash function**
  - Cryptographical one-way function
  - Hash from a document - easy
  - Document from a hash is hard! (brute force)
- **Access to coins public / private keys**
  - Private key lost == funds lost
- **Transactions in blocks**
- **Create a valid block ~ 10min (Bitcoin)**
  - Find nonce that SHA256 hash has enough bits set to zero - brute force (crypto puzzle)
  - Creation of blocks is called mining (reward)
  - Waste of energy?
    - Country closest to Bitcoin in terms of electricity consumption: Austria



VSS W12: Bitcoin‘s Market Capitalization in USD



2009 ***2011*** ***2014*** ***2016*** ***2018***

VSS W12: Bitcoin - Protocol

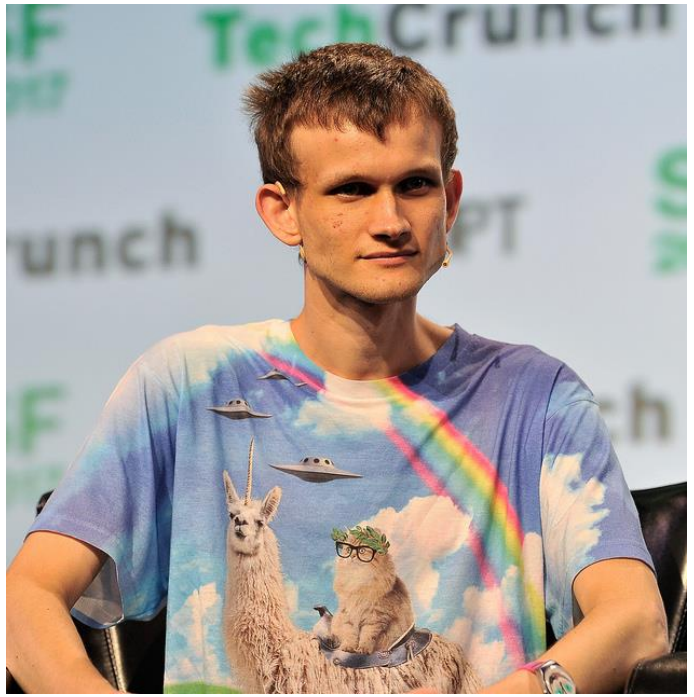
- **TX in details**

version		01 00 00 00
input count		01
input	previous output hash (reversed)	48 4d 40 d4 5b 9e a0 d6 52 fc a8 25 8a b7 ca a4 25 41 eb 52 97 58 57 f9 6f b5 0c d7 32 c8 b4 81
	previous output index	00 00 00 00
	script length	8a
	scriptSig	47 30 44 02 20 2c b2 65 bf 10 70 7b f4 93 46 c3 51 5d d3 d1 6f c4 54 61 8c 58 ec 0a 0f f4 48 a6 76 c5 4f f7 13 02 20 6c 66 24 d7 62 a1 fc ef 46 18 28 4e ad 8f 08 67 8a c0 5b 13 c8 42 35 f1 65 4e 6a d1 68 23 3e 82 01 41 04 14 e3 01 b2 32 8f 17 44 2c 0b 83 10 d7 87 bf 3d 8a 40 4c fb d0 70 4f 13 5b 6a d4 b2 d3 ee 75 13 10 f9 81 92 6e 53 a6 e8 c3 9b d7 d3 fe fd 57 6c 54 3c ce 49 3c ba c0 63 88 f2 65 1d 1a ac bf cd
	sequence	ff ff ff ff
output count		01
output	value	62 64 01 00 00 00 00 00
	script length	19
	scriptPubKey	76 a9 14 c8 e9 09 96 c7 c6 08 0e e0 62 84 60 0c 68 4e d9 04 d1 4c 5c 88 ac
block lock time		00 00 00 00

http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html

VSS W13: Bitcoin / Ethereum

- **“Issues” with Bitcoin**
  - Slow development, implementing new features difficult
    - SegWit vs. BTU (segregated witness vs. increasing block size voting)
  - Bitcoin Script limited
    - Lightning network
- **Ethereum (1 ETH ~330$)**
  - 1 ETH ~ 230$
  - Generalized blockchain (loops, arithemitcs, etc.)
  - «Blockchain 2.0» with smart contracts
  - White paper released in December 2013, release July 2015
  - Ethereum foundation located in Zug (initiator known) - non-profit foundation
  - Mining reward ~ block every ~14s – ~3 (2) ETH

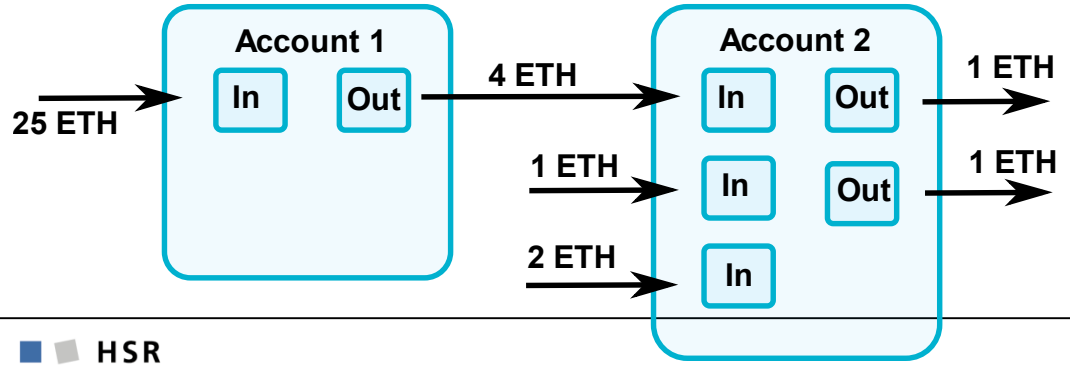


Vitalik Buterin

VSS W13: Account vs UTXO - Introduction

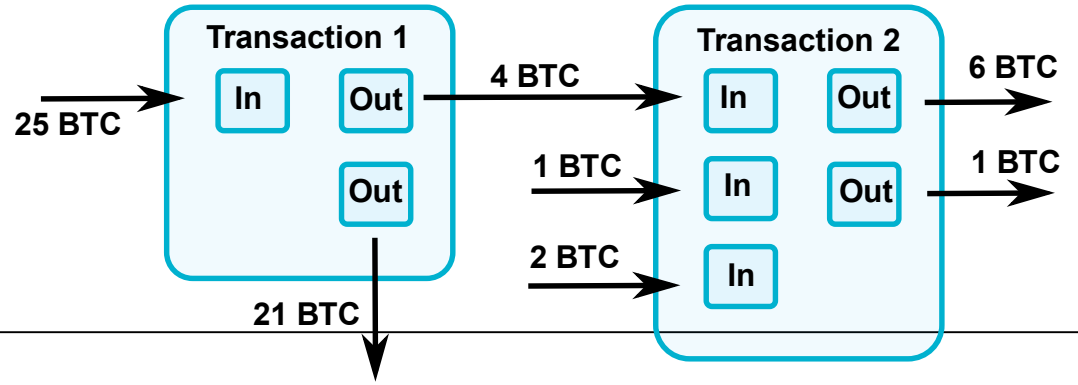
Account-based

- Global state stores a list of accounts with balances and code
- Transaction is valid if the sending account has enough balance
  - Balance on sender is deducted, new balance
- If the receiving account has code, the code runs, and state may be changed
  - Signature must match sending account



UTXO-based

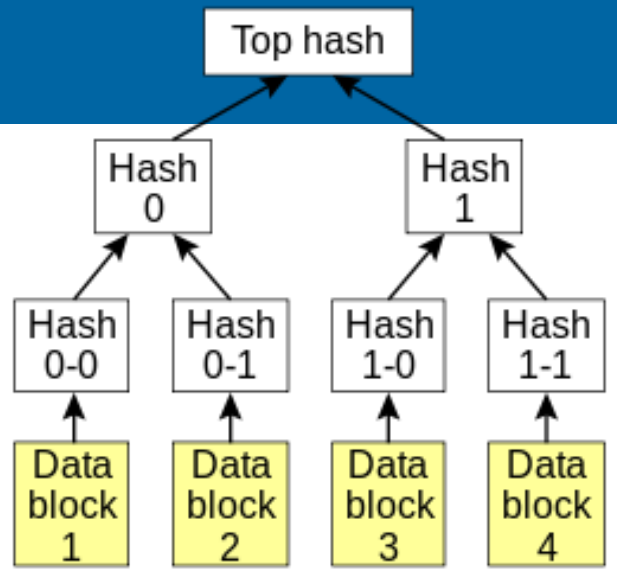
- Every referenced input must be valid and not yet spent
- Total value of the inputs must equal or exceed the total value of the outputs
  - You always spend all outputs
- Transaction must have a signature matching the owner of the input for every input
  - Script determines if input is valid



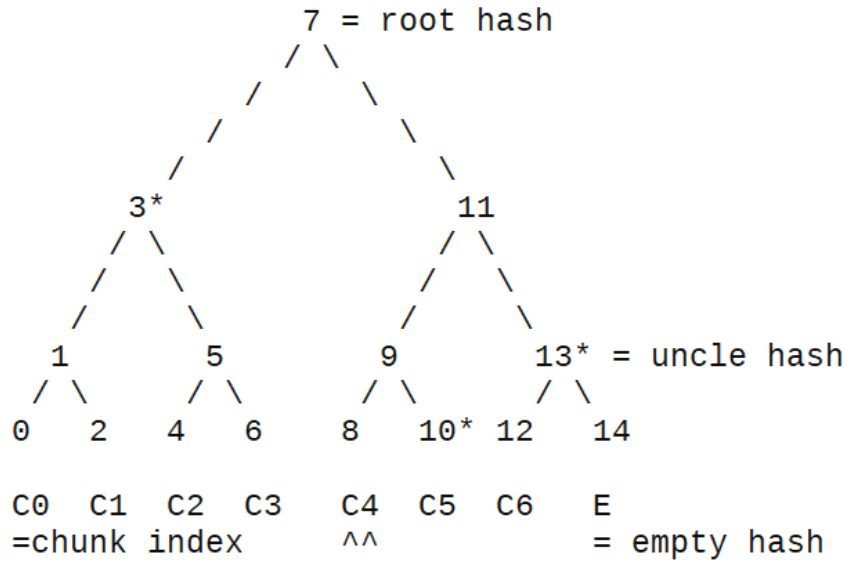
VSS W13: Merkle hash

- **Hash Tree**
  - Tree of hashes (|| → concatenation)
  - `hash 0 = hash( hash 0-0 || hash 0-1 )`
  - `hash 1 = hash( hash 1-0 || hash 1-1 )`
  - `Top hash = hash( hash 0 || hash 1 )`
- **Verification (BitTorrent)**
  - Peer A has top hash (root hash)
  - Peer downloads C4 from peer B
  - create hash 8
  - Need hash 10, 13, 3 (uncle hash)
  - Can be from peer B
  - With 8,10,13,3 can create root hash

  → verify this root hash
- **Merkle tree / hash tree also used in [Bitcoin / Ethereum]**



http://en.wikipedia.org/wiki/Hash_tree



The Merkle hash tree of an interval of width W=8

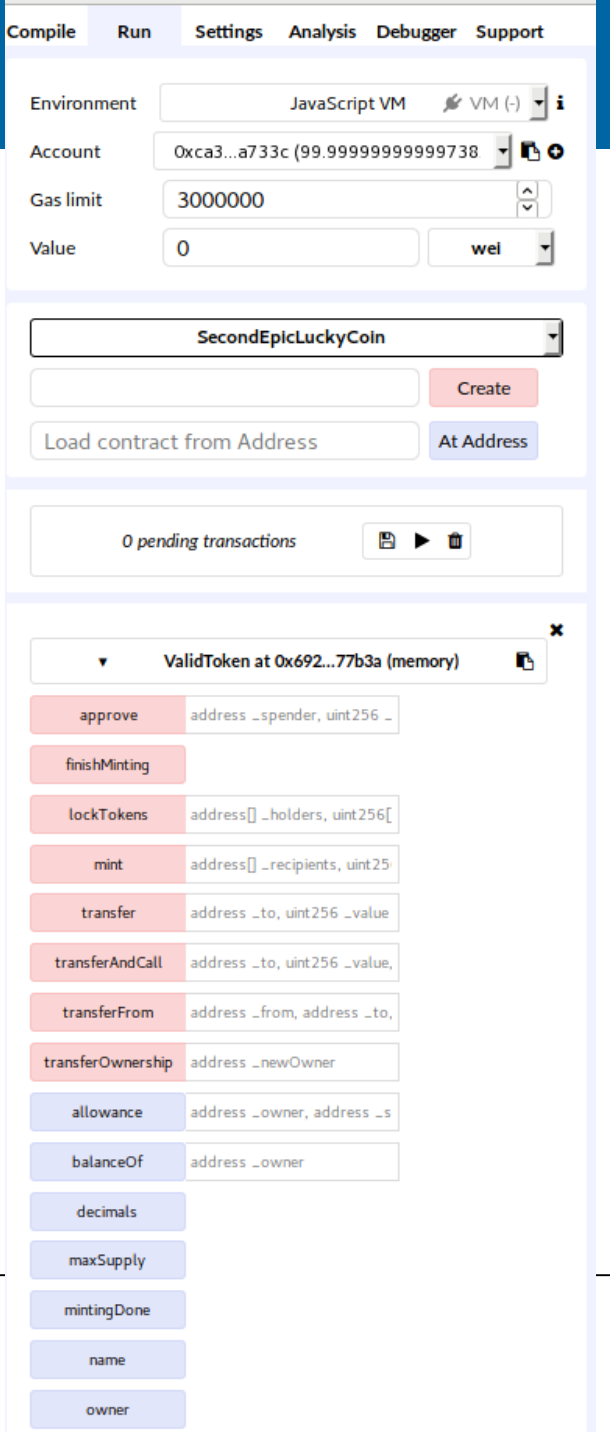
http://datatracker.ietf.org/doc/draft-ietf-ppsp-peer-protocol/ Section 5.2

VSS W13: Solidity IDE

- **Create your first contract**

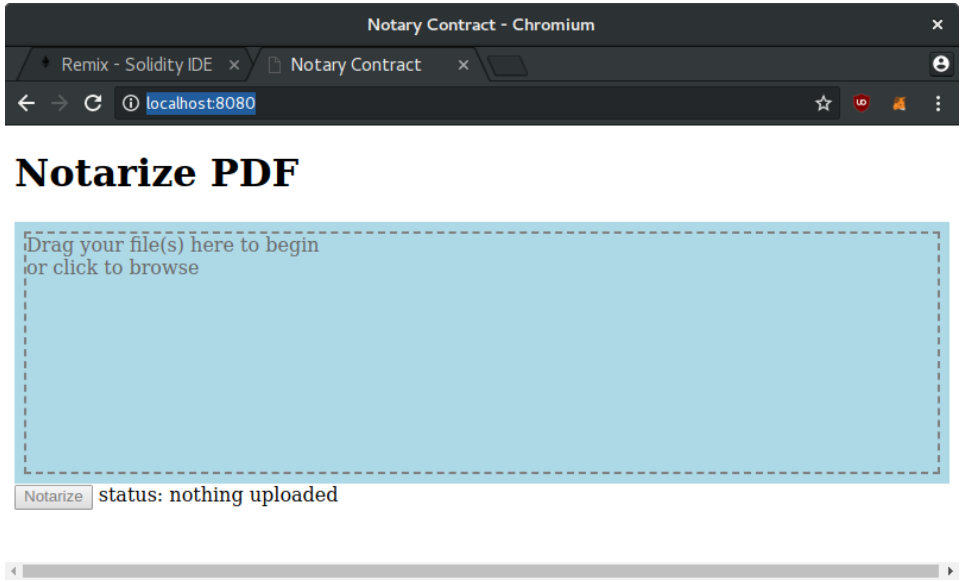
```
pragma solidity ^0.4.23;
//minimal contract
contract Example1 {
    uint counter;
}
```

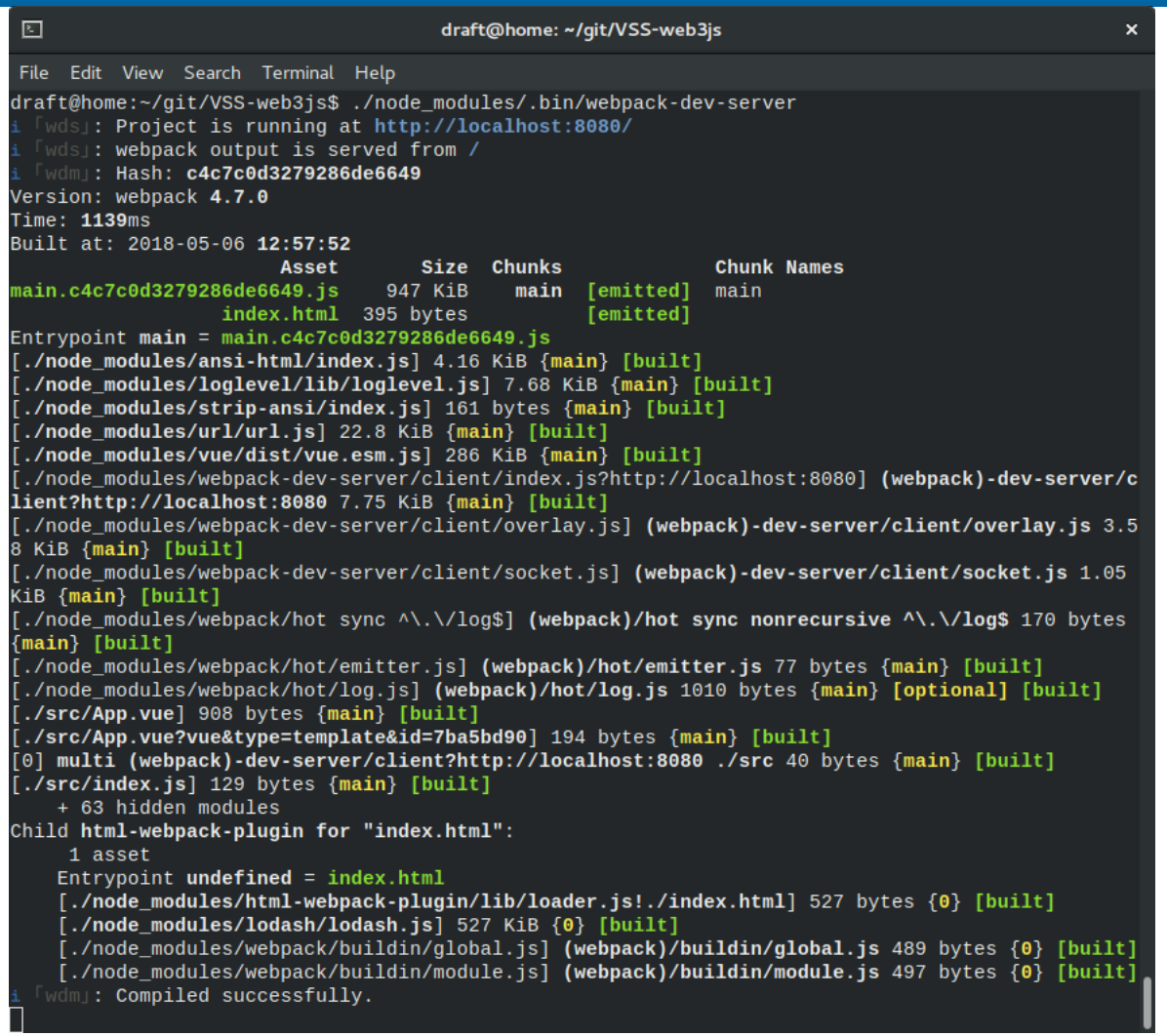
- **Remix – Solidity IDE – Environment: Web3 Provider**
  - http://localhost:8545
- **Push «Create» (red button)**
- **Mixed content - workaround: use http**
  - **https://remix.ethereum.org**
- **Select Web3 Provider (geth is your Web3 Provider)**
  - Alternatively: use scripts, MetaMask



VSS W15: Run it locally

- **Installation**
  - yarn install
  - ./node_modules/.bin/webpack-dev-server
- **Open Browser: http://localhost:8080/**





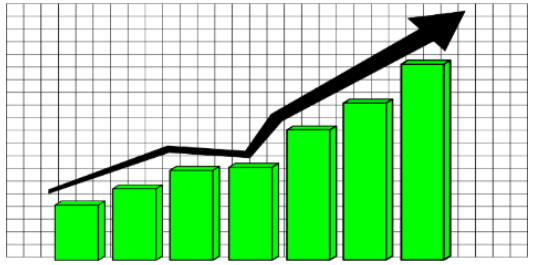
URL: b.socrative.com/student/

Room: HSR

Discussion (1)

Advantages

- «Low» (fixed) tx fees
  - 6 satoshi per byte (chart)
- Scalable
  - Hardware/storage gets faster



- Anonymity
  - No privacy concerns/ datamining difficult

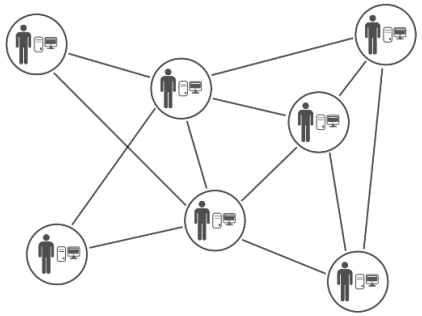
Disadvantages

- Power consumption
  - ~ 8400 Megawatt (KKW Beznau ~730MW)
- Not scalable
  - Bitcoin with 3-7 tps vs. VISA 57,000 tps (23.12) [tps: transactions per sec]
- Anonymity
  - Can be used for illegal activities

Discussion (2)

Advantages

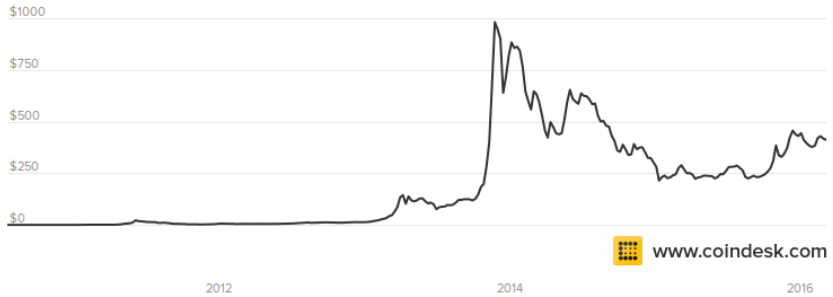
- No major “crashes”
  - Mt.Gox was exchange site!
- Decentralized
  - Open protocol



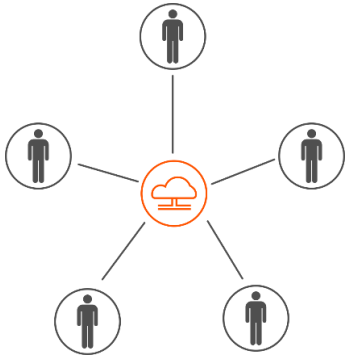
- Other blockchain usage
  - Smart contracts, Registry of Deeds

Disadvantages

- Volatile exchange rate

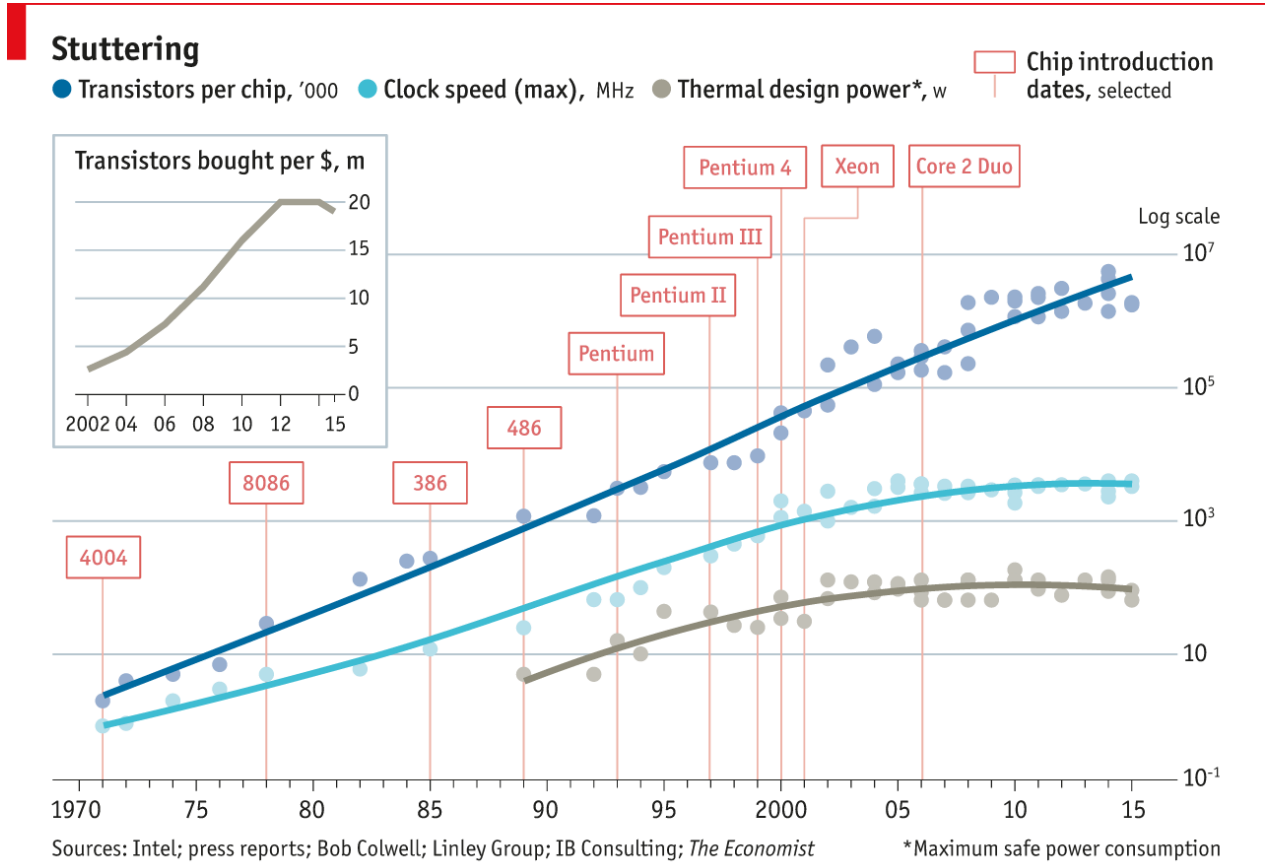


- Central elemements
  - Core developers



Trends (1)

- **Moore's Law – nr. of transistors doubles every 18 month**
- **Dead in 2021**



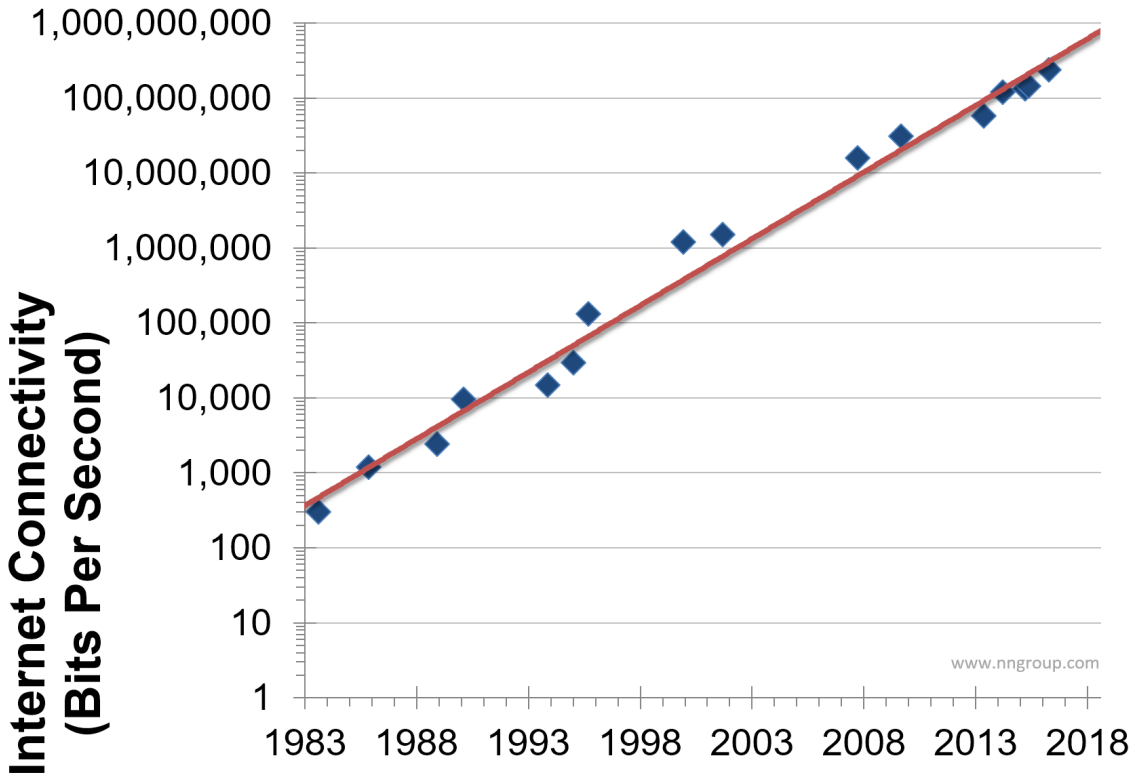
Trends (2)

- **Nielsen’s Law: a high-end user's connection speed grows by 50% per year**
- **Bandwidth grows slower than computer power**
- **Zmap complete scan of the IPv4 address space in under 5 minutes**
  - Not every provider is friendly for experiments:

From: network-abuse@hetzner.com
Sent: 24 July 2018 13:57:05 CEST
Subject: Abuse-Message: NetscanOutLevel: Netscan detected from 188.40.119.70 [Network-Normal]

wir haben Hinweise, dass von Ihrem Server aus ein Angriff durchgeführt wurde.

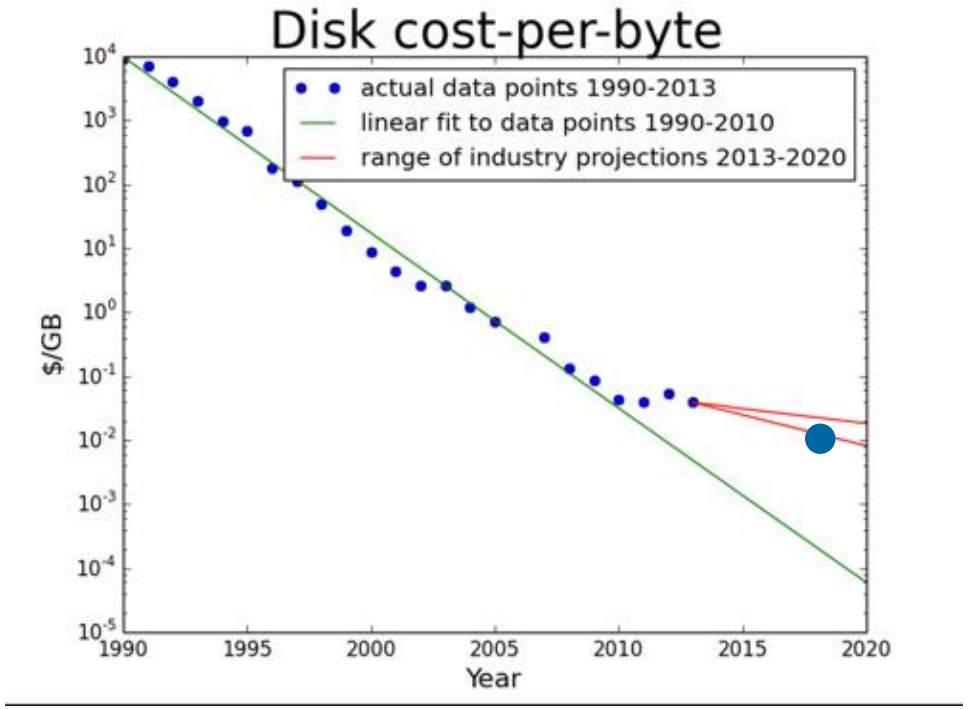
Wir bitten Sie alle nötigen Maßnahmen zu treffen, um dies künftig zu vermeiden und um die Lösung des Problems.
Außerdem bitten wir Sie um die Abgabe einer kurzen Stellungnahme an uns. Diese Stellungnahme soll Angaben enthalten, wie es zu dem Vorfall kommen konnte, bzw. was Sie dagegen unternehmen werden.
Sollten folgende Schritte nicht erfolgreich durchgeführt werden, kann Ihr Server zu jedem Zeitpunkt nach dem 2018-07-24 17:57:04 +0200 gesperrt werden.



Trends (3)

- **Kryder's Law: disk density doubling every 13 month**
- **«Soon hard drives will migrate into phones, still cameras, PDAs, cars and everyday appliances»**
  https://www.scientificamerican.com/article/kryders-law/ , Aug. 2005

- **User behavior changed**
  - SSD, speed is important
- **Cloud – Dropbox, Spotify**
  - Streaming



http://blog.dshr.org/2016/05/the-future-of-storage.html

Questions?

Prof. Dr. Thomas Bocek thomas.bocek@hsr.ch
Roman Blum roman.blum@hsr.ch