Blockchain – Zwischen Transparenz und Datenschutz

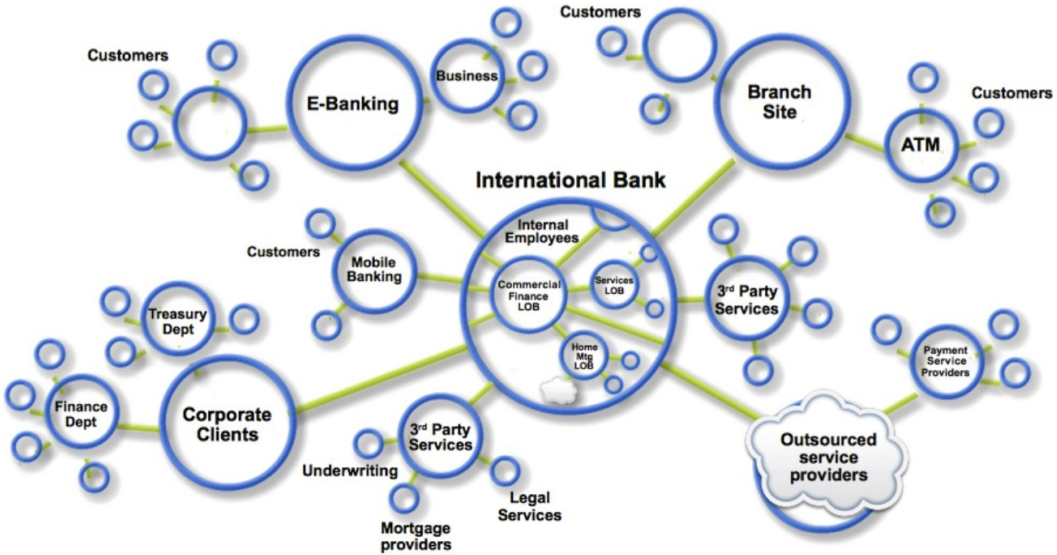
Christian Cachin

IBM Research – Zurich

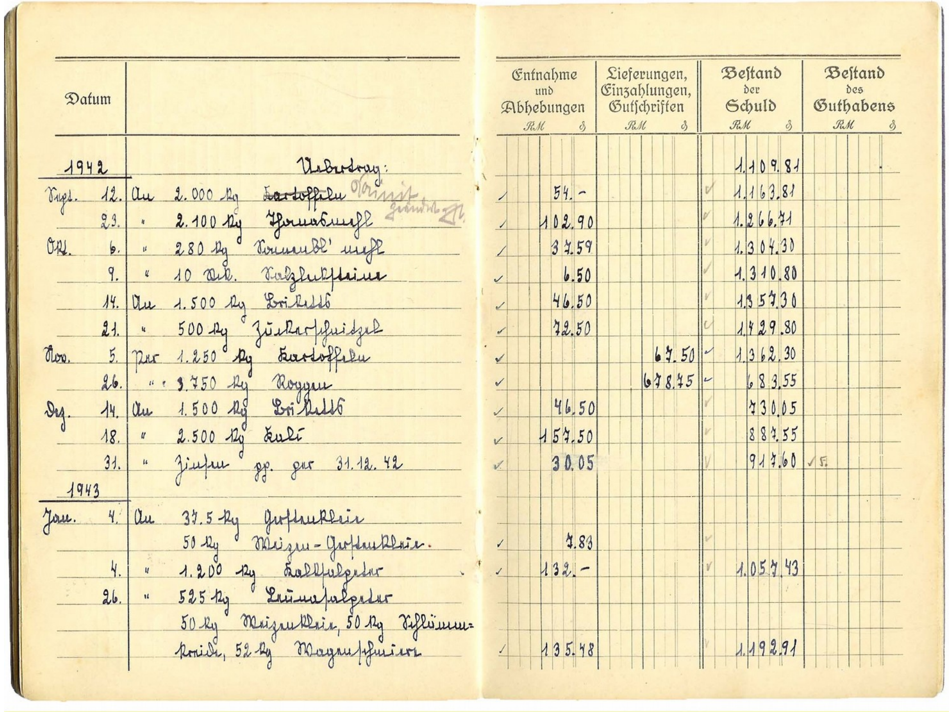
August 2017

Connected markets

- Networks connect participants
  - Customers, suppliers, banks, consumers
- Markets organize trades
  - Public and private markets
- Wealth generated by flow of assets and services among participants
  - Physical (house, car ...) and virtual assets (bond, patent ...)
  - Services
- Transactions exchange assets

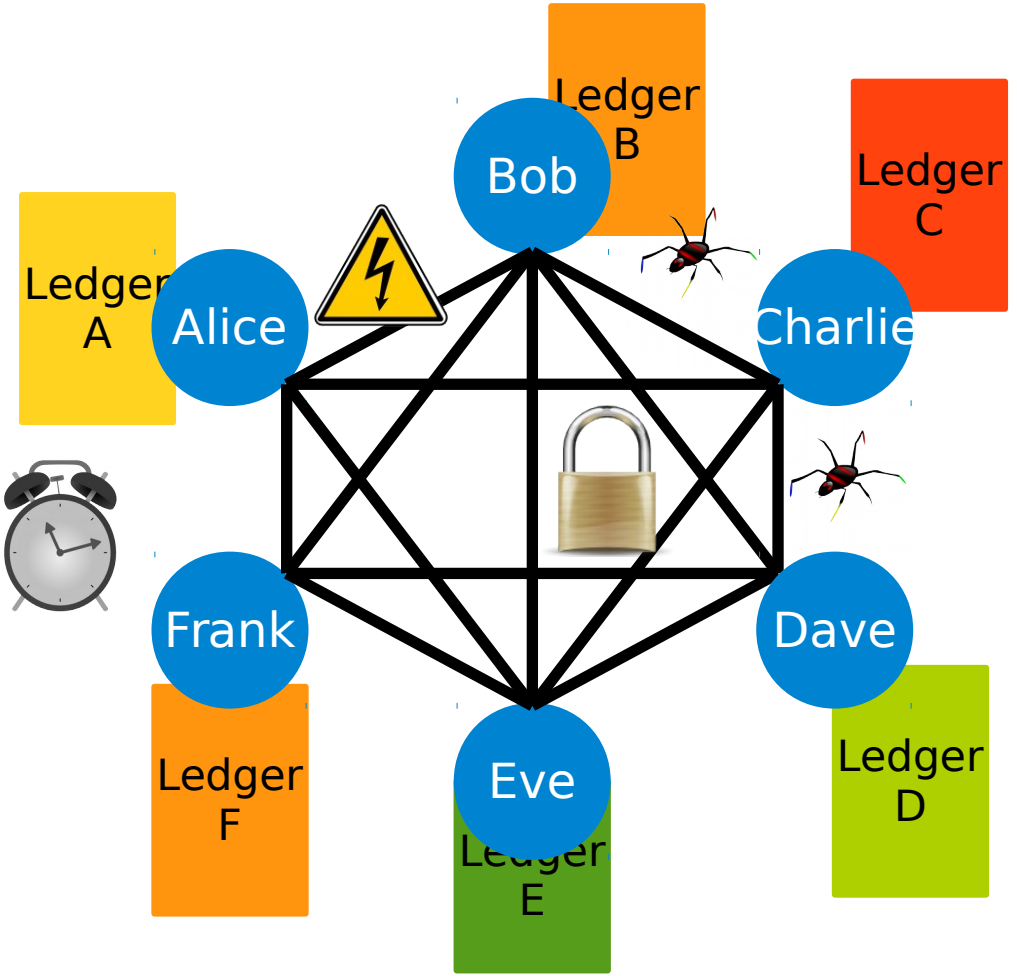


Ledger



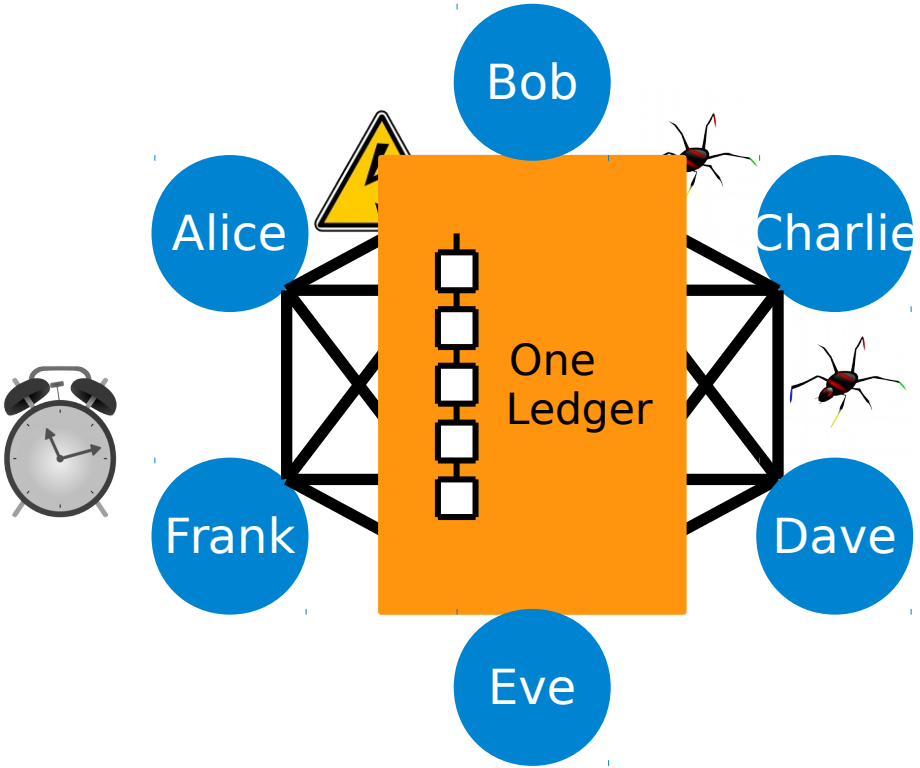
- Ledger records all business activity as transactions
  - Databases
- Every market and network defines a ledger
- Ledger records asset transfers between participants
- Problem — (Too) many ledgers
  - Every market has its ledger
  - Every organization has its own ledger

Multiple ledgers



- Every party keeps its own ledger and state
- Problems, incidents, faults
- Diverging ledgers

Blockchain provides one virtual ledger



- One common trusted ledger
- Today often implemented by a centralized intermediary
- Blockchain creates one single ledger for all parties
- Replicated, transparent and produced collaboratively
- Trust in ledger from
  - Cryptographic protection
  - Distributed validation

Four elements characterize Blockchain

Replicated ledger

- History of all transactions
- Append-only with immutable past
- Distributed and replicated

Cryptography

- Integrity of ledger
- Authenticity of transactions
- Privacy of transactions
- Identity of participants

Consensus

- Decentralized protocol
- Shared control tolerating disruption
- Transactions validated

Business logic

- Logic embedded in the ledger
- Executed together with transactions
- From simple "coins" to self-enforcing "smart contracts"

Blockchain simplifies complex transactions

Logistics

Real-time visibility
Improved efficiency
Transparency & verifiability
Reduced cost

Property records

Digital but unforgeable
Fewer disputes
Transparency & verifiability
Lower transfer fees

Capital markets

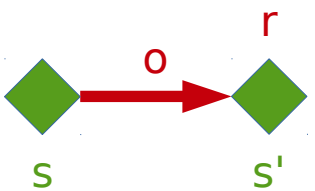
Faster settlement times
Increased credit availability
Transparency & verifiability
No reconciliation cost

What is a blockchain?

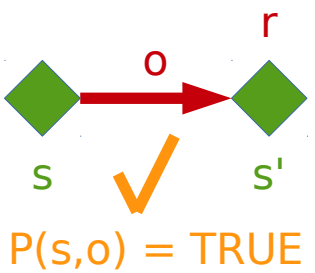
A state machine

- Functionality F
  - Operation o transforms a state s to new state s' and may generate a response r

$(s', r) \leftarrow F(s, o)$

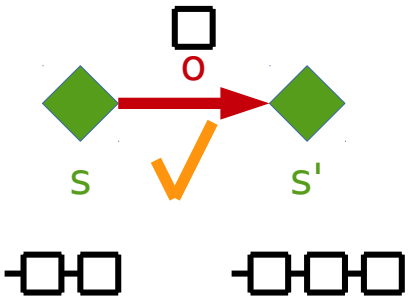


- Validation condition
  - Operation needs to be valid, in current state, according to a predicate P()



Blockchain state machine

- Append-only log
  - Every operation o appends a "block" of valid transactions (tx) to the log



- Log content is verifiable from the most recent element

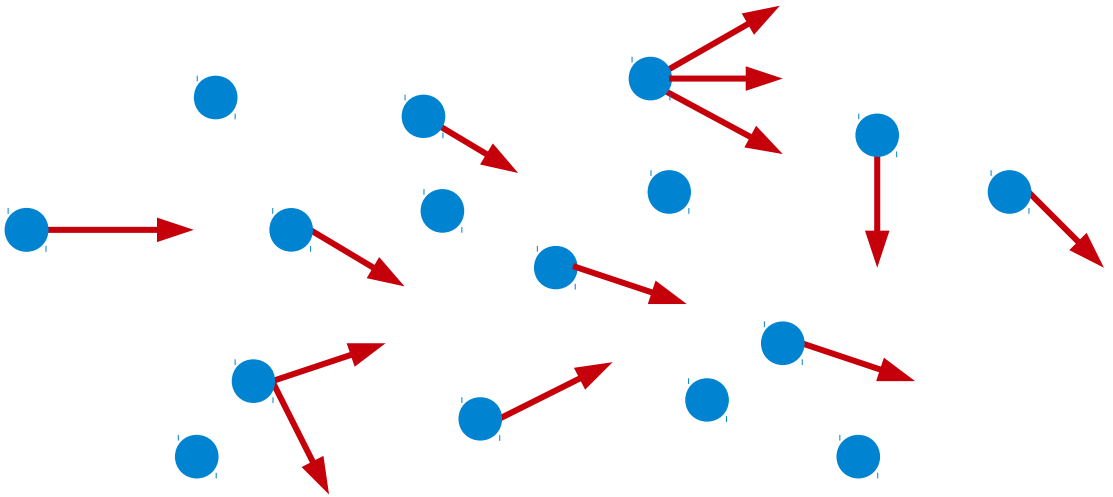
- Log entries form a hash chain

  $h_t \leftarrow \text{Hash}([tx_1, tx_2, \ldots] \,||\, h_{t-1} \,||\, t)$.

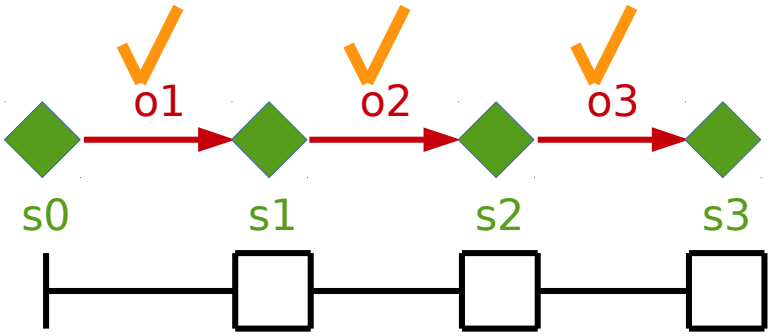
Example – The Bitcoin state machine

- Bitcoins are unforgeable bitstrings
  - "Mined" by the protocol itself (see later)
- Digital signature keys (ECDSA) own and transfer bitcoins
  - Owners are pseudonymous, e.g., 3JDs4hAZeKE7vER2YvmH4yTMDEfoA1trnC
- Every transaction transfers a bitcoin (fraction) from current to next owner
  - "This bitcoin now belongs to 3JDs..." signed by the key of current owner
  - (Flow linkable by protocol, and not anonymous when converted to real-world assets)
- Validation is based on the global history of past transactions
  - Signer has received the bitcoin before
  - Signer has not yet spent the bitcoin

Distributed p2p protocol to create a ledger



Nodes
produce
transactions



Nodes run a protocol to construct the ledger

Blockchain features

- A given task or problem, but no (central) trusted party available
- Protocol among multiple nodes, solving a distributed task
  - The writing nodes decide and reach consensus collectively
- Key aspects of the distributed task
  - Stores data
  - Multiple nodes write
  - Not all writing nodes are trusted
  - Operations are (somewhat) verifiable
- If all writing nodes are known → permissioned or consortium blockchain
- Otherwise, when writing nodes are not known → permissionless or public blockchain

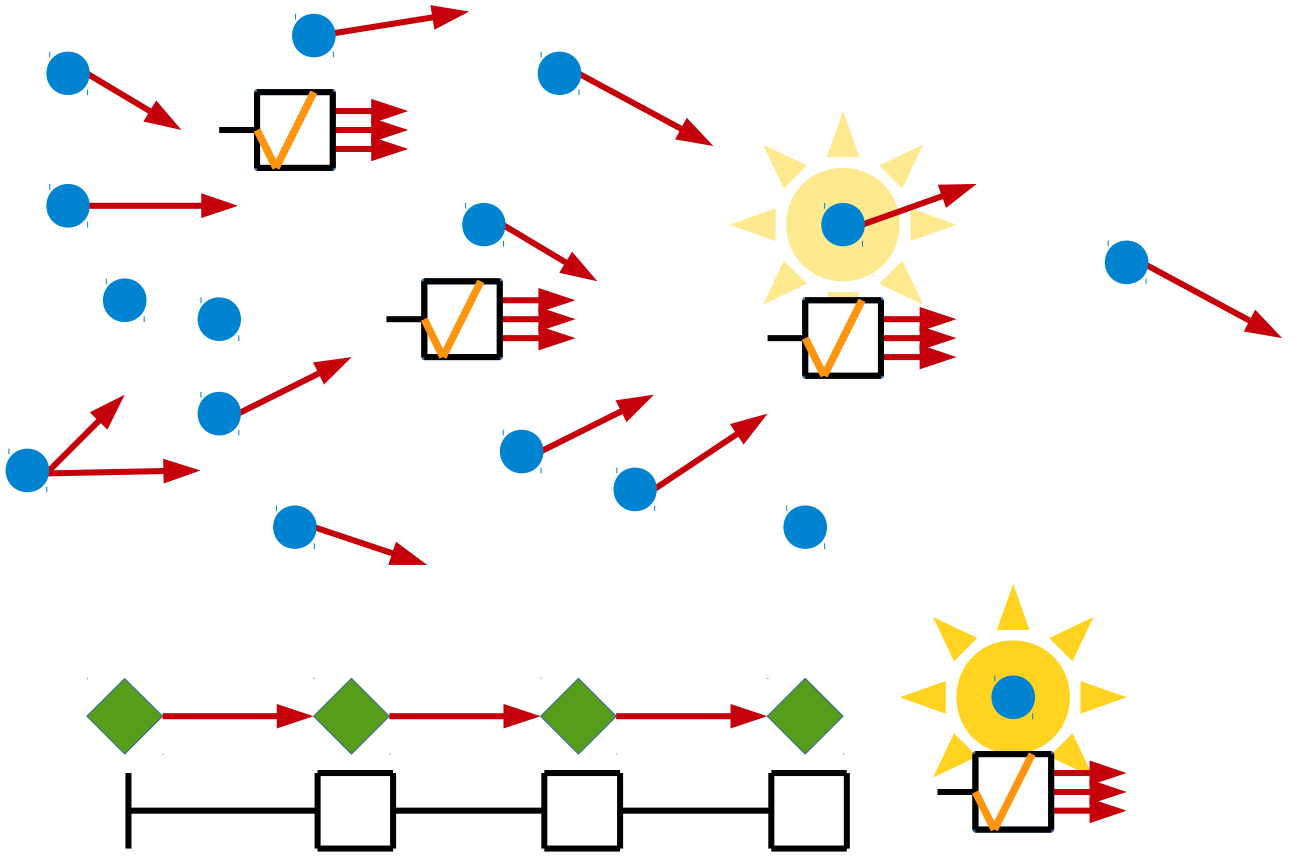
Consensus

Two kinds of consensus for blockchain

- Decentralized / permissionless
  - Bitcoin, Ethereum
- Consortium / permissioned
  - BFT (Byzantine fault tolerance) consensus

Decentralized – Nakamoto consensus/Bitcoin

- Nodes prepare blocks
  - List of transactions (tx)
  - All tx valid
- Lottery race
  - Solves a hard puzzle
  - Selects a random winner/leader
  - Winner's operation/ block is executed and "mines" a coin
- All nodes verify and validate new block
  - "Longest" chain wins

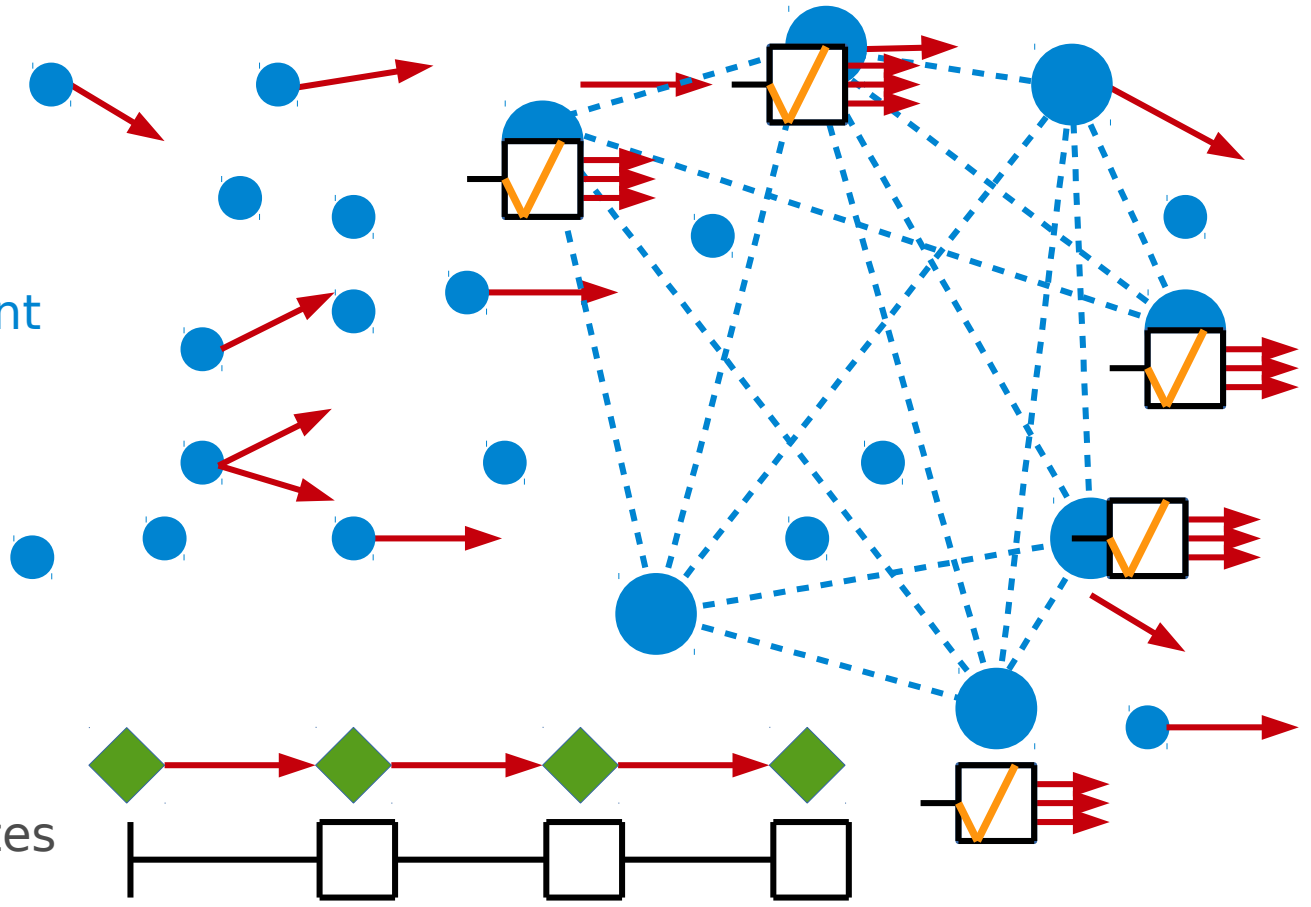


Decentralized = permissionless

- Survives censorship and suppression
  - No central entity
- Nakamoto consensus requires proof-of-work (PoW)
  - Original intent: one CPU, one vote
  - Majority of hashing power controls network
  - Gives economic incentive to participate (solution to PoW is a newly "mined" Bitcoin)
- Today, total hashing work consumes a lot of electricity
  - Estimates vary, 250-1000MW, from a major city to a small country ...
- Protocol features
  - Stability is a tradeoff between dissemination of new block (10s-20s) and mining rate (new block on average every 10min)
  - Decisions are not final ("wait until chain is 6 blocks longer before a tx is confirmed")

Consortium consensus (quorums & BFT)

- Designated set of homogeneous validator nodes
- BFT/Byzantine agreement
  - Tolerates f-out-of-n faulty/ adversarial nodes
  - Generalized quorums
- Tx sent to consensus nodes
- Consensus validates tx, decides, and disseminates result



Consortium consensus = permissioned

- Central entity controls group membership
  - Dynamic membership changes in protocol
  - Membership may be decided inline, by protocol itself
- Well-understood problem in distributed computing
  - BFT and consensus studied since ca. 1985
    - Clear assumptions and top-down design
    - Many protocols, textbooks, and open-source implementations (BFT-SMaRT)
  - Many systems already provide crash tolerant consensus (Chubby, Zookeeper, etcd ...)
  - Requires $\Omega(n^2)$ communication (OK for 10-100 nodes, not > 1000s)
- Revival of research in BFT protocols
  - Focus on scalability and communication efficiency

Validation

Validation of transactions

- Recall validation predicate P on state s and operation o: P(s, o)

- All nodes collectively validate transactions to be executed

- Validation can be expensive
  - Bitcoin blockchain contains the log of all tx – 130GB as of 8/2017
    (https://blockchain.info/charts/blocks-size)

Public validation vs. private state

- Blockchain is transparent and public – where is privacy?
- In Bitcoin, verification is a digital signature by key that owns coin
  - Addresses are pseudonyms, but flow of coins is traceable (by design)
- In ZeroCash, blockchain holds committed coins and tx flow is hidden
  - Using zero-knowledge proofs validated blockchain consensus
- Cryptography allows to keep state "off-chain" and produce verifiable tx that maintain privacy
- Privacy only with sophisticated cryptographic protocols
  - Only few solutions available so far

Forms of privacy in blockchains

- Transactional privacy
  - Anonymity or pseudonymity through cryptographic tools
  - Some is feasible today (e.g., anonymous credentials in IBM Identity Mixer)
- Contract privacy
  - Distributed secure cryptographic computation on encrypted data
- Accountability & non-repudiation
  - Identity and cryptographic signatures
- Auditability & transparency
  - Cryptographic hash chain
- Many of these need advanced cryptographic protocols

Blockchain for business
–
Hyperledger Fabric

Hyperledger

- A Linux Foundation project – www.hyperledger.org
  - Open-source collaboration, developing blockchain technologies for business
  - Started in 2016: Hyperledger unites industry leaders to advance blockchain technology
  - 135 members in Apr. '17

- Incubates and promotes blockchain technologies for business
- Today 4 frameworks and 4 tools, hundreds of contributors

- Hyperledger Fabric was originally contributed by IBM – github.com/hyperledger/fabric/
  - Architecture and consensus protocols originally contributed by IBM Research - Zurich

Hyperledger Fabric

- Blockchain fabric and distributed ledger framework for business
  - One of multiple blockchain platforms in the Hyperledger Project
  - First "active" platform under the Hyperledger umbrella (since 3/2017)

- Developed open-source, by IBM and others (DAH, State Street, HACERA ...)
  - github.com/hyperledger/fabric
  - Initially called 'openblockchain' and contributed by IBM to Hyperledger project
  - Key technology for IBM's blockchain strategy
  - Actively developed, IBM and IBM Zurich play key roles

- Technical details
  - Implemented in GO
  - Runs smart contracts or "chaincode" within Docker containers
  - Transactions Deploy new chaincode / Invoke an operation / Read state
  - Implements consortium blockchain using traditional consensus (BFT, ZooKeeper)

Conclusion

Conclusion

- Blockchain enables new trust models
  - Distributes trust over the Internet
- Combines distributed computing and cryptography
  - Need fresh of privacy
  - Truly privacy-preserving blockchains will be costly
- Tradeoff between transparency and privacy

  - www.hyperledger.org
  - www.ibm.com/blockchain/
  - www.research.ibm.com/blockchain/
  - www.zurich.ibm.com/blockchain/
  - www.zurich.ibm.com/~cca/