Blockchains and consensus protocols

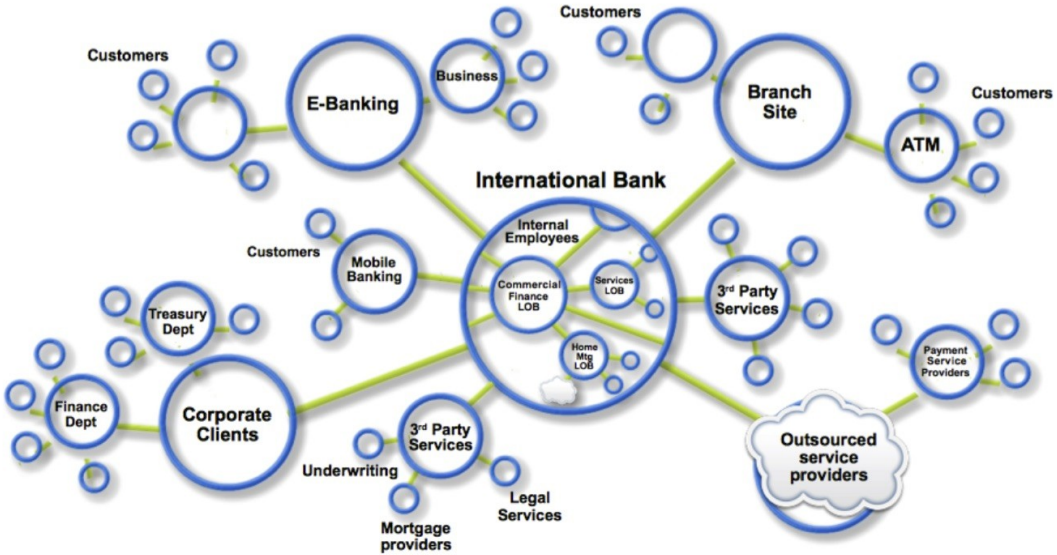
Christian Cachin
(with many others at IBM: Elli Androulaki, Angelo De Caro, Andreas Kind, Mike Osborne, Simon Schubert, Alessandro Sorniotti, Marko Vukolic ...)

IBM Research – Zurich

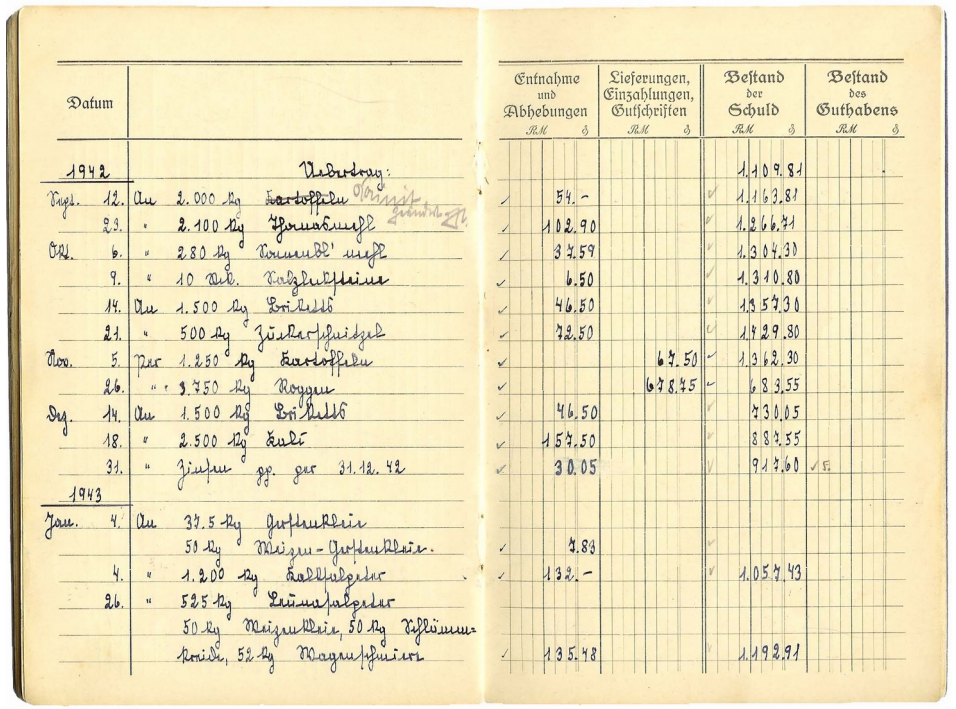
September 2017

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



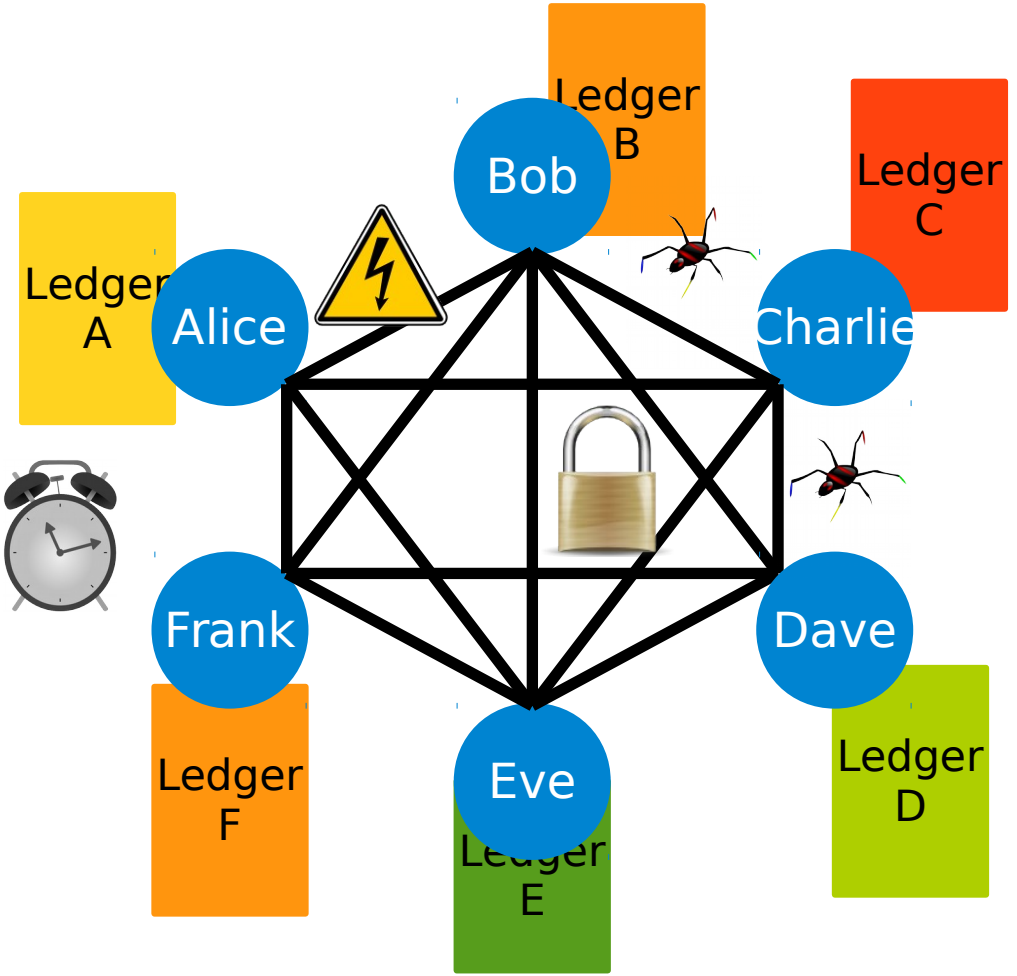
Ledger



Datum		Entnahme und Abhebungen RM ₰	Lieferungen, Einzahlungen, Gutschriften RM ₰	Bestand der Schuld RM ₰	Bestand des Guthabens RM ₰
1942	Übertrag:			1.109.81	
Dez. 12.	An 2.000 kg Kartoffeln	54.–		1.163.81	
29.	" 2.100 kg Brennholz	102.90		1.266.71	
Okt. 6.	" 280 kg Brennholz	37.59		1.304.30	
9.	" 10 Ztr. Kohlenbriketts	6.50		1.310.80	
14.	An 1.500 kg Briketts	46.50		1.357.30	
21.	" 500 kg Zuckerschnitzel	72.50		1.429.80	
Nov. 5.	Per 1.250 kg Kartoffeln		67.50	1.362.30	
26.	" " 3.750 kg Roggen		678.75	683.55	
Dez. 14.	An 1.500 kg Briketts	46.50		730.05	
18.	" 2.500 kg Kalk	157.50		887.55	
31.	" Zinsen pp. per 31.12.42	30.05		917.60	
1943					
Jan. 4.	An 37.5 kg Getreideabf.				
	50 kg Weizen-Getreideabf.	3.83			
4.	" 1.200 kg Kalkspeise	132.–		1.057.43	
26.	" 525 kg Zuckerspeise				
	50 kg Weizenkeime, 50 kg Kleiemm-				
	kuchen, 52 kg Magnesiumsuper	135.48		1.192.91	

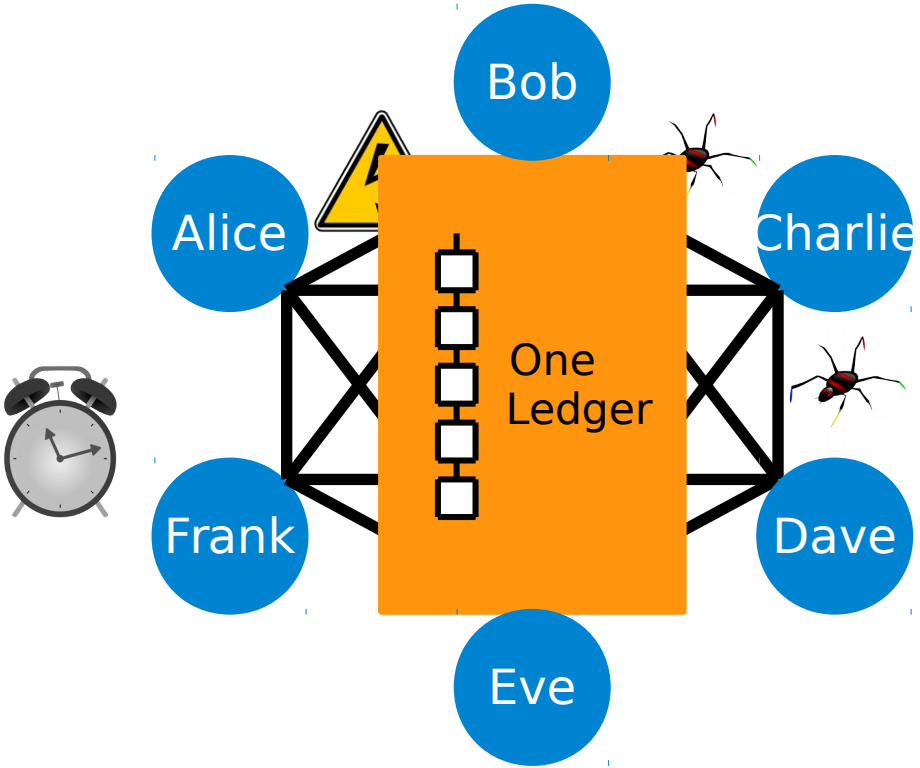
- Ledger records all business activity as transactions
  - Databases
- Every market and network defines a ledger
- Ledger records asset transfers between participants
- Problem — (Too) many ledgers
  - Every market has its ledger
  - Every organization has its own ledger

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 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

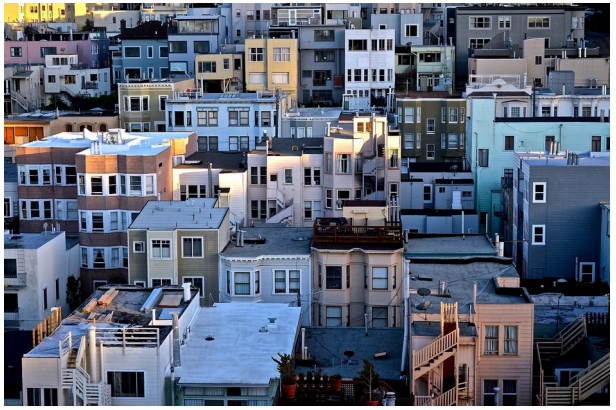
Business logic

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

Blockchain simplifies complex transactions

Financial assets

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



Property records

Digital but unforgeable
Fewer disputes
Transparency & verifiability
Lower transfer fees

Logistics

Real-time visibility
Improved efficiency
Transparency & verifiability
Reduced cost

Blockchain scenario 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

What is a blockchain?

A shorter version of this paper appears in *Proc. Intl. Conference on Dependable Systems and Networks (DSN-2001)*, Gothenborg, Sweden, IEEE, 2001.

Distributing Trust on the Internet

Christian Cachin

IBM Research
Zurich Research Laboratory
CH-8803 Rüschlikon, Switzerland
cca@zurich.ibm.com

March 8, 2001

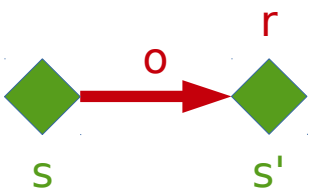
Abstract

This paper describes an architecture for secure and fault-tolerant service replication in an asynchronous network such as the Internet, where a malicious adversary may corrupt some servers and control the network. It relies on recent protocols for randomized Byzantine agreement and for atomic broadcast, which exploit concepts from threshold cryptography. The model and its assumptions are discussed in detail and compared to related work from the last decade in the first part of this work, and an overview of the broadcast protocols in the architecture is provided. The standard approach in fault-tolerant distributed systems is to assume that at most a certain fraction of servers fails. In the second part, novel general failure patterns and corresponding protocols are introduced. They allow for realistic modeling of real-world trust assumptions, beyond (weighted) threshold models. Finally, the application of our architecture to trusted services is discussed.

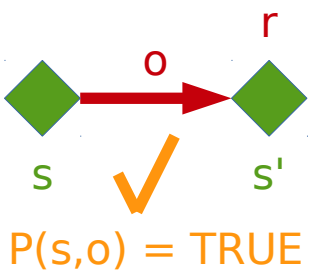
A state machine

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

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

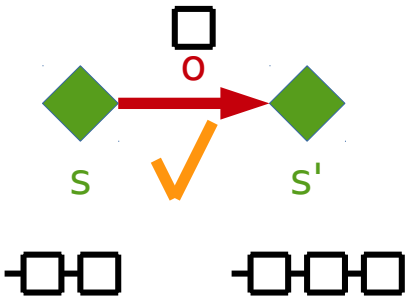


- Validation condition
  - Operation needs to be valid, in current state, according to a predicate P()



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

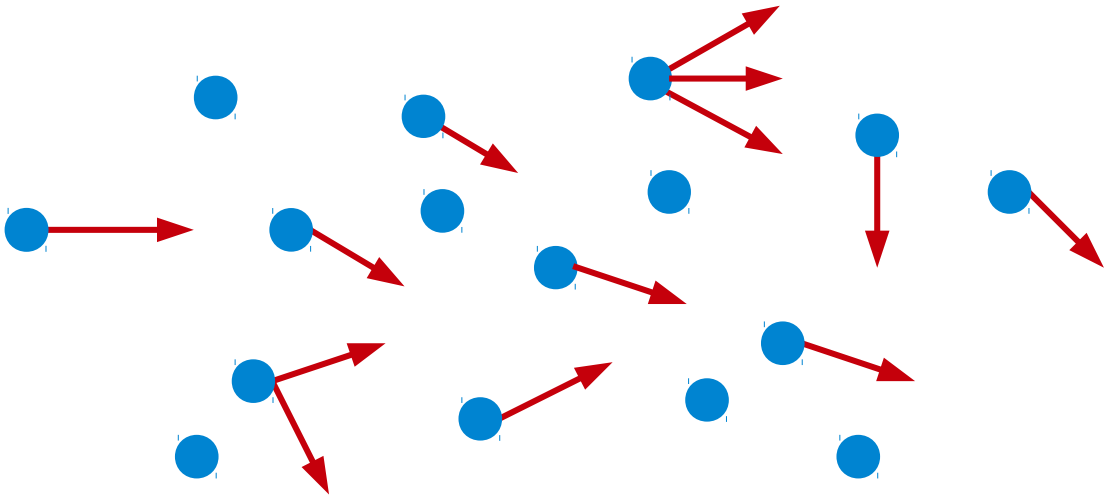
- Bitcoins are unforgeable bitstrings
  - "Mined" by the protocol itself (see later)

- Digital signature keys (ECDSA) own and transfer bitcoins
  - Owners are pseudonymous, e.g., 3JDs4hAZeKE7vER2YvmH4yTMDEfoA1trnC

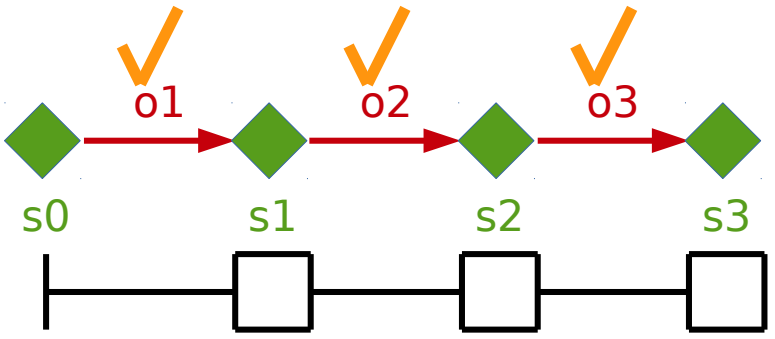
- Every transaction transfers a bitcoin (fraction) from current to next owner
  - "This bitcoin now belongs to 3JDs..." signed by the key of current owner
  - (Flow linkable by protocol, and not anonymous when converted to real-world assets)

- 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 protocol features

- Only "valid" operations (transactions) are "executed"

- Transactions can be simple
  - Bitcoin tx are statement of ownership for coins, digitally signed
    "This bitcoin now belongs to K2" signed by K1

- Transactions can be arbitrary code (smart contracts)
  - Embody logic that responds to events (on blockchain) and may transfer assets in response
  - Auctions, elections, investment decisions, blackmail ...

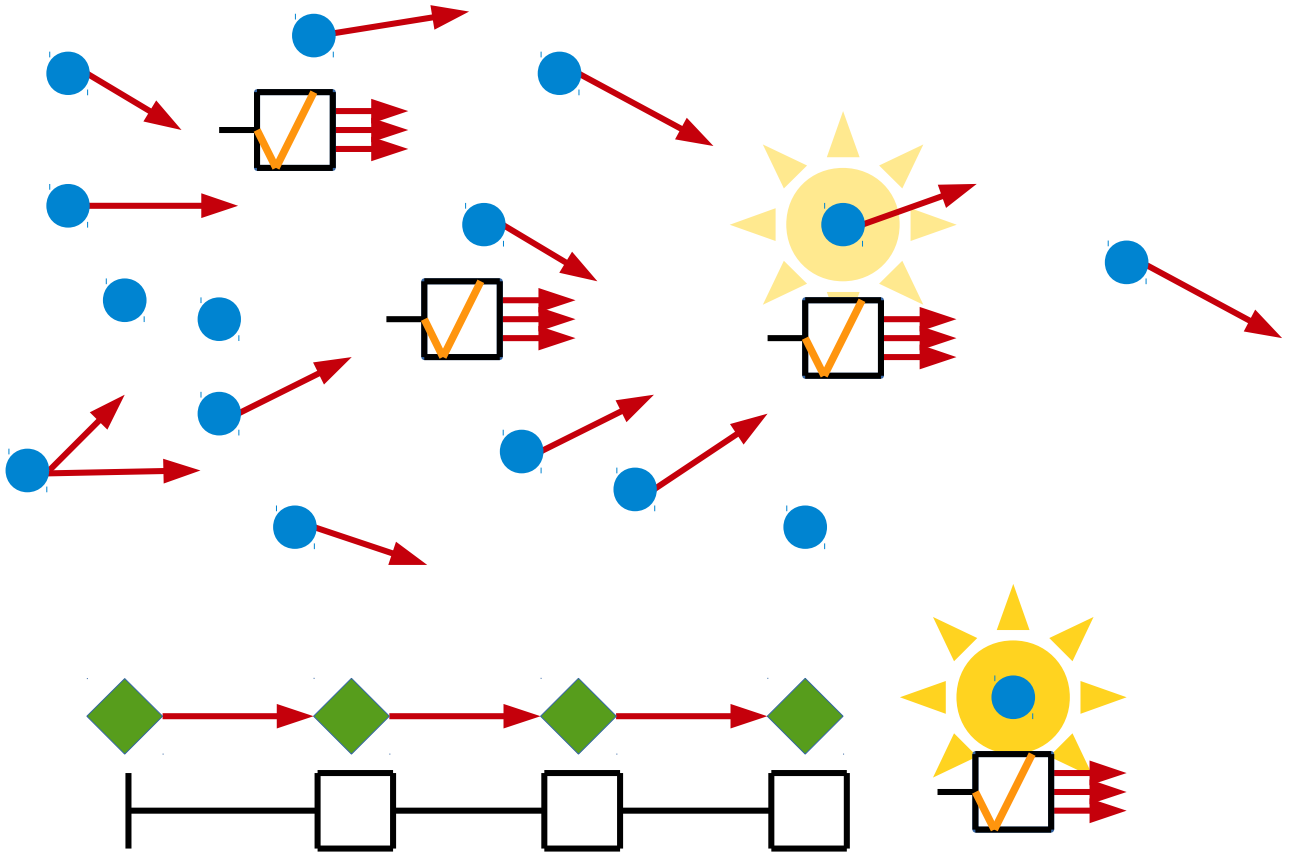
Consensus

Two kinds of consensus for blockchain

- Decentralized / permissionless
  - Bitcoin, Ethereum
- Consortium / permissioned
  - BFT (Byzantine fault tolerance) consensus
- [[Somewhat decentralized]]
  - [[Ripple, Stellar]]
- [[Alternative proposals, unstructured]]
  - Swirlds Hashgraph, IOTA Tangle, Skycoin, Tezos ...

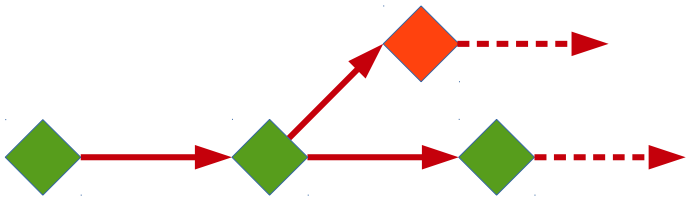
Decentralized – Nakamoto consensus/Bitcoin

- 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



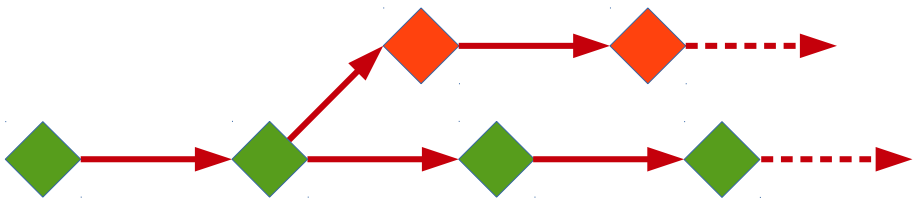
How does proof-of-work ensure consistency?

- Miners solve puzzle to create blocks
  - Concurrent, may have conflicting tx
  - Disseminate block, fast
  - Mining reward
- "Longest" chain wins



How does proof-of-work ensure consistency?

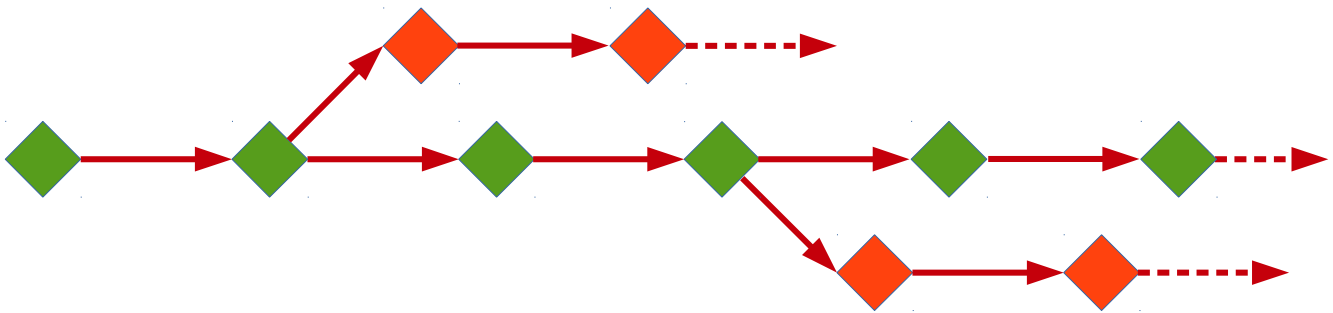
- Miners solve puzzle to create blocks
  - Concurrent, may have conflicting tx
  - Disseminate block, fast
  - Mining reward
- "Longest" chain wins



- Forks occur regularly
  - With probability independent of past

How does proof-of-work ensure consistency?

- Miners solve puzzle to create blocks
  - Concurrent, may have conflicting tx
  - Disseminate block, fast
  - Mining reward
- "Longest" chain wins



- Forks occur regularly
  - With probability independent of past
- Forks do not last forever, with high probability
  - Bitcoin tx confirmed if 6 blocks deep
  - Probability of k-blocks long fork exponentially small in k
- Alternative rules exist to select winning chain (GHOST ...)

Decentralized = permissionless

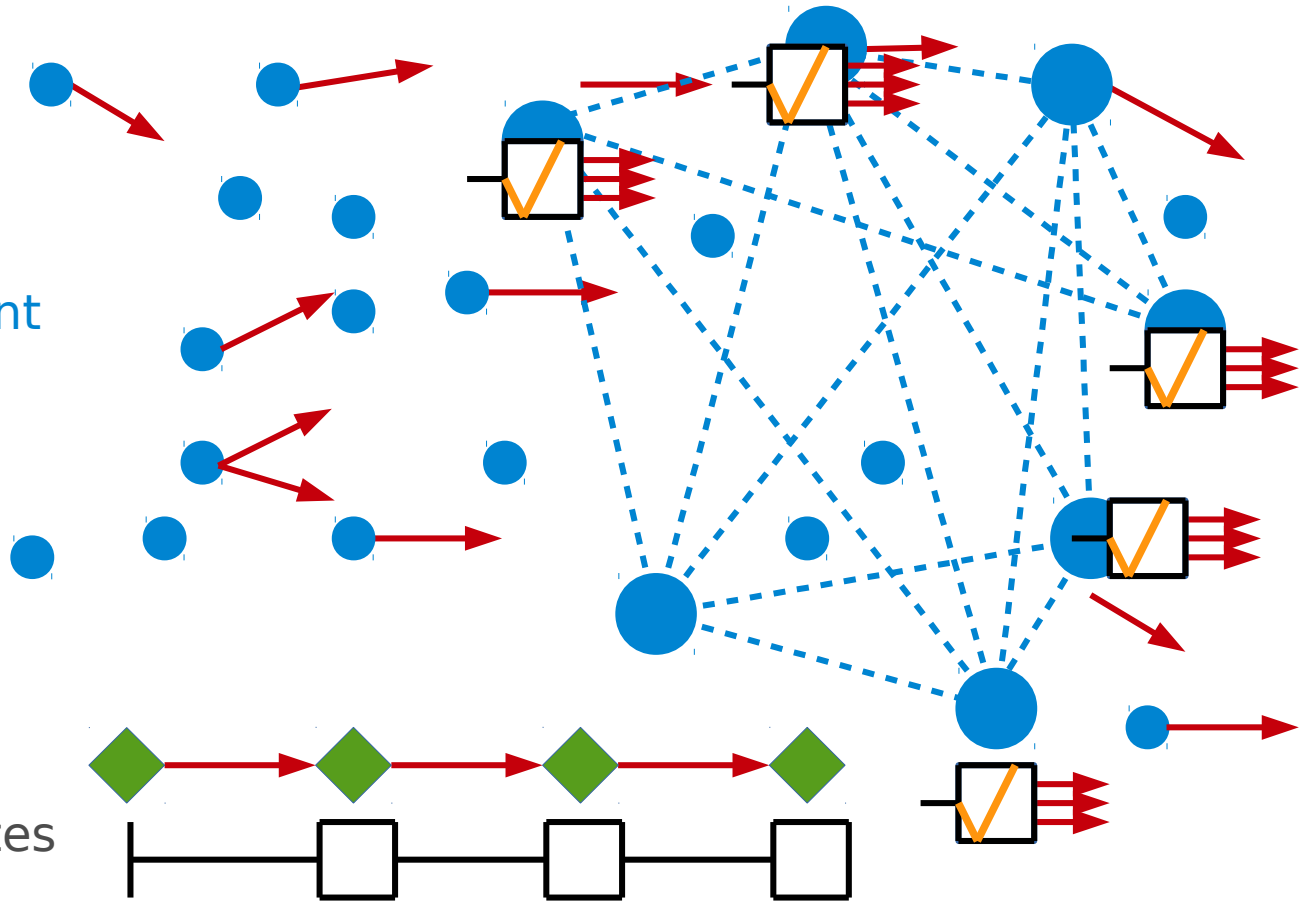
- Survives censorship and suppression
  - No central entity
- Nakamoto consensus requires proof-of-work (PoW)
  - Original intent: one CPU, one vote
  - Majority of hashing power controls network
  - Gives economic incentive to participate (solution to PoW is a newly "mined" Bitcoin)
- Today, total hashing 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 (bitcoin tx confirmed only when 6 blocks deep in chain)

Decentralized – deployment

- Bitcoin
  - Many (100s? 1000s?) of alt-coins and blockchains
- Ethereum
  - First digital currency with general-purpose smart contract execution
- Sawtooth ledger (Intel contribution to Hyperledger)
  - PoET consensus (proof of elapsed time)
    - Nodes run PoET program in "trusted execution environment" (Intel SGX)
    - PoET waits a random amount of time (say, E[wait] = 10min)
    - Creates an attested proof of elapsed time
    - Rest like in Bitcoin protocol

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
    - 700 protocols and counting [AGK+15]
    - Textbooks [CGR11]
    - 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

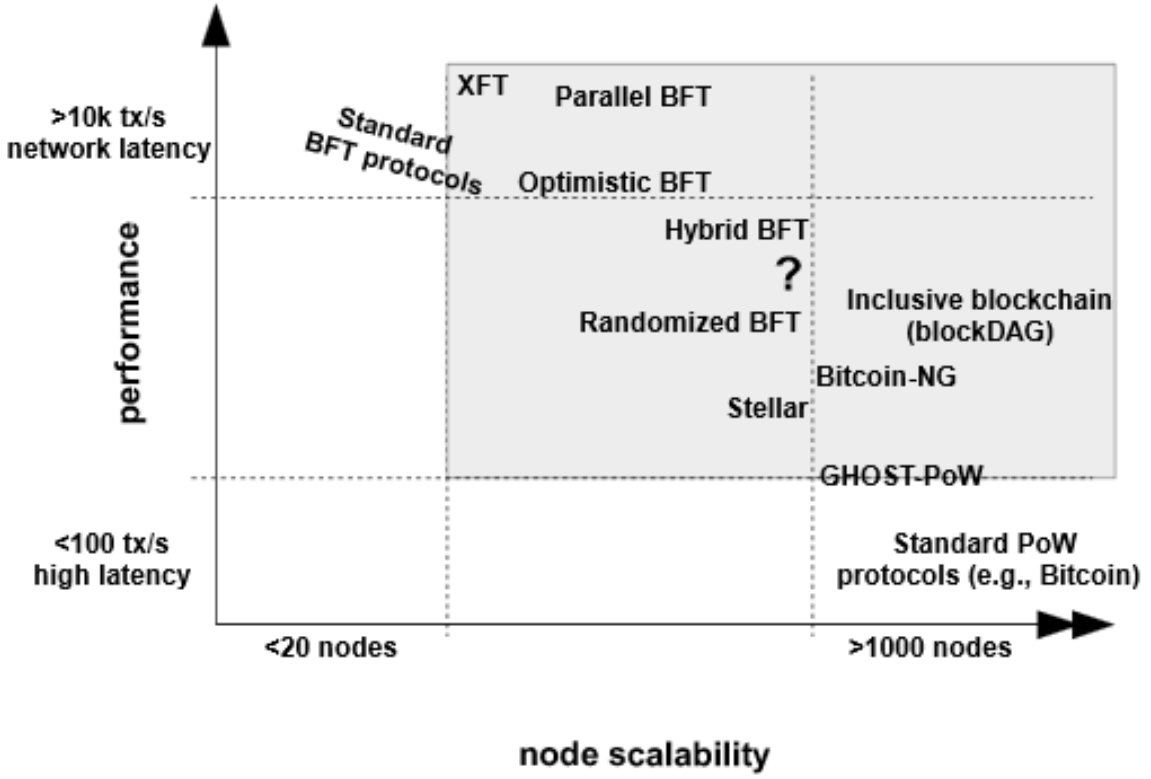
Consortium consensus – under development

- Hyperledger Fabric (originally started by IBM)
  - v0.6 includes PBFT protocol [CL02]
  - 1.0 includes ZooKeeper/Kafka protocol (crash tolerant)
- Some BFT libraries predate blockchain
  - BFT-SMaRT, Univ. Lisbon (github.com/bft-smart/library)
  - Currently as prototype for several blockchain platforms (Fabric, Symbiont, R3-Corda ...)
- Multiple "new" permissioned consensus protocols
  - Tendermint, Juno/Kadena, JPMC Quorum, Iroha, Chain ...

More variations of blockchain consensus

- Bitcoin-NG [EGS+16]
  - Bitcoin PoW elects a leader, it is responsible for ordering the next K tx
- Hybrid, BFT over proof-of-work [DSW16,PS17,]
  - PoW protocol to elect nodes in one consensus group
  - Group runs ordinary BFT consensus
- Proof-of-stake (many research papers, explored by Ethereum)
  - Voting power relative to asset holdings (through cryptocurrency held by blockchain)
- Hierarchical & partitioned, randomized [LNB+15]
  - Random sub-groups, nodes and tx assigned randomly to sub-groups
  - Each sub-group runs ordinary BFT consensus

Scalability–performance tradeoff between decentralized and consortium consensus



M. Vukolic: The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. Proc. iNetSec 2015, LNCS 9591.

Consensus in permissioned blockchains

- Trust model
  - n nodes, some fraction f are faulty
- Protocols are well-understood
  - Paxos/Viewstamped Replication (VSR), ZooKeeper, Raft (many more!) tolerate crashes
  - PBFT, Byzantine randomized consensus (many more!) tolerate malicious nodes
- Despite that, many platforms have invented their own protocols
  - Often without sufficient public analysis
- Many platforms come out [CV17]
  - Hyperledger Fabric, Tendermint, Symbiont, R3-Corda, Iroha, Kadena, Chain, Quorum, MultiChain

Consensus protocols are like cryptosystems

- It is impossible to demonstrate that a cryptosystem works by demonstration
  - One can only demonstrate that it fails
- Kerckhoffs principle: System design is public, assumptions are explained
- Modern cryptography (1975 – today) introduced mathematical models, proofs, and public scientific study
- Open discussion, expert reviews, broad validation, and standards
- During Internet boom (~2000) many "unbreakable cryptosystems" popped up
  - Ignoring established practice and analysis
  - Soon refuted as "cryptographic snake oil" by Schneier and others

Permissioned consensus overview

Which faults are tolerated by a protocol?	**Special-node crash**	**Any $t < n/2$ nodes crash**	**Special-node subverted**	**Any $f < n/3$ nodes subverted**
Hyperledger Fabric/Kafka	.	✓	.	–
Hyperledger Fabric/PBFT	.	✓	.	✓
Tendermint	.	✓	.	✓
Symbiont/BFT-SMaRt	.	✓	.	✓
R3 Corda/Raft	.	✓	.	–
R3 Corda/BFT-SMaRt	.	✓	.	✓
Iroha/Sumeragi (BChain)	.	✓	.	✓
Kadena/ScalableBFT	?	?	?	?
Chain/Federated Consensus	–	(✓)	–	–
Quorum/QuorumChain	–	(✓)	–	–
Quorum/Raft	.	✓	.	–
MultiChain +	.	✓	.	–
Sawtooth Lake/PoET	⊕	✓	⊕	–
Ripple	⊗	(✓)	⊗	–
Stellar/SCP	?	?	?	?
IOTA Tangle	?	?	?	?

Table 1: Summary of consensus resilience properties, some of which use statically configured nodes with a *special* role. Symbols and notes: '✓' means that the protocol is resilient against the fault and '–' that it is not; '.' states that no such *special node* exists in the protocol; '?' denotes that the properties cannot be assessed due to lack of information; (✓) denotes the crash of *other* nodes, different from the special node; + MultiChain has non-final decisions; ⊕ PoET assumes trusted hardware available from only one vendor; ⊗ Ripple tolerates *one* of the five default Ripple-operated validators (special nodes) to be subverted.

C. Cachin, M. Vukolic: Blockchain consensus protocols in the wild. arXiv:1707.01873 [cs.DC], 2017

Validation

Validation of transactions – PoW protocols

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

- When constructing a block, the node
  - Validates all contained tx
  - Decides on an ordering within block

- When a new block is propagated, all nodes must validate the block and its tx
  - Simple for Bitcoin – verify digital signatures and that coins are unspent
  - More complex and costly for Ethereum – re-run all the smart-contract code

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

Validation of transactions – BFT protocols

- Properties of ordinary Byzantine consensus
  - Weak Validity: Suppose all nodes are correct: if all propose v, then a node may only decide v; if a node decides v, then v was proposed by some node.
  - Agreement: No two correct nodes decide differently.
  - Termination: Every correct node eventually decides.

- Standard validity notions do not connect to the application!

- Need validity anchored at external predicate [CKPS01]
  - External validity: Given predicate P, known to every node, if a correct node decides v, then P(v); additionally, v was proposed by some node.

  - Can be implemented with digital signatures on input tx

Public validation vs. private state

- So far everything on blockchain is public – where is privacy?
- Use cryptography – keep state "off-chain" and produce verifiable tx
  - In Bitcoin, verification is a digital signature by key that owns coin
  - In ZeroCash [BCG+14], blockchain holds committed coins and transfers use zero-knowledge proofs (zk-SNARKS) validated by P
  - Hawk [KMS+16] uses verifiable computation (VC)
    - Computation using VC performed off-chain by involved parties
    - P checks correctness of proof for VC
- Private computation requires additional assumption (MPC, trusted HW ...)

Security and privacy

- 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

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


PREMIER MEMBERS
accenture
High performance. Delivered.
AIRBUS
AMERICAN EXPRESS
CHANGE HEALTHCARE
CME Group
DEUTSCHE BÖRSE GROUP
DAIMLER
Digital Asset
DTCC
Securing Today. Shaping Tomorrow.
FUJITSU
HITACHI
Inspire the Next
IBM
intel
J.P.Morgan
NEC
r3.
SAP
万达·飞凡科技
WANDA FFAN TECHNOLOGY
ASSOCIATE MEMBERS
BANK OF ENGLAND
CHAMBER OF DIGITAL COMMERCE
CSA cloud security alliance
ColoredCoins
FEDERAL RESERVE BANK OF BOSTON
IC3
Investrata Foundation
NXT FOUNDATION
OSCRE
sovrin
BLOCKCHAIN RESEARCH INSTITUTE
TNO innovation for life
The Illinois BLOCKCHAIN Initiative
fondazione INUIT
VSP Global


GENERAL MEMBERS
1World Blockchain
ABN·AMRO
AESTHETIC INTEGRATION
AlphaPoint
ALTOROS
ANZ
BBVA
博图纵横
belink
BITMARK
bitse
BLOCKCHAIN
blocko
Blockstream
bloq
BNP PARIBAS
BNY MELLON
Broadridge
bubi.cn
ca technologies
Calastone
链链
Chain Connected
招商银行
CISCO
中信 CITIC
clause
cloudsoft
CLS
Fundamental to FX
coinplug
collector
colu.
ConsensusBase
consensys
Cuscal
The complete payments partner
Deloitte.
ENERGY BLOCKCHAIN LABS
EY
Eurostep Digital
FACTOM
房掌柜
FZG360.COM
Gem
PeerSafe
guardtime
33.CN
复杂美
HASHED HEALTH
HUAWEI
HUNDSUN
趣链科技
Hyperchain
intellect
intuit
IROOTECH
KAISER PERMANENTE
KRX
KOREA EXCHANGE
KSD
Korea Securities Depository


koscom
kubique
LedgerDomain
Libra
loyyal
Lykke
MadHive
Milligan Partners
MIRACL
MONAX
MonetaGo
ML
MOSCOW EXCHANGE
MURPHY & McGONIGLE
NSE
NEC
NETKI
NEX
NOKIA
norbloc
NTT DATA
onchain
橙色魔方
PAXOS
PDX
RadarWin
redhat
SAMSUNG
SAMSUNG SDS
SANY
SBERBANK
Schroders
SECURE KEY
GINGKOO
onechain
四方精创
FORMS SYNTRON
NEXGO
点融网
SORAMITSU
STATE STREET
SWIFT
swisscom
symbiont
tequa creek
THOMSON REUTERS
TMX
UMF 联动优势
vmware
WELLS FARGO
云象
YUNPHANT
梧桐树
保全网
BaoQuan.com

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

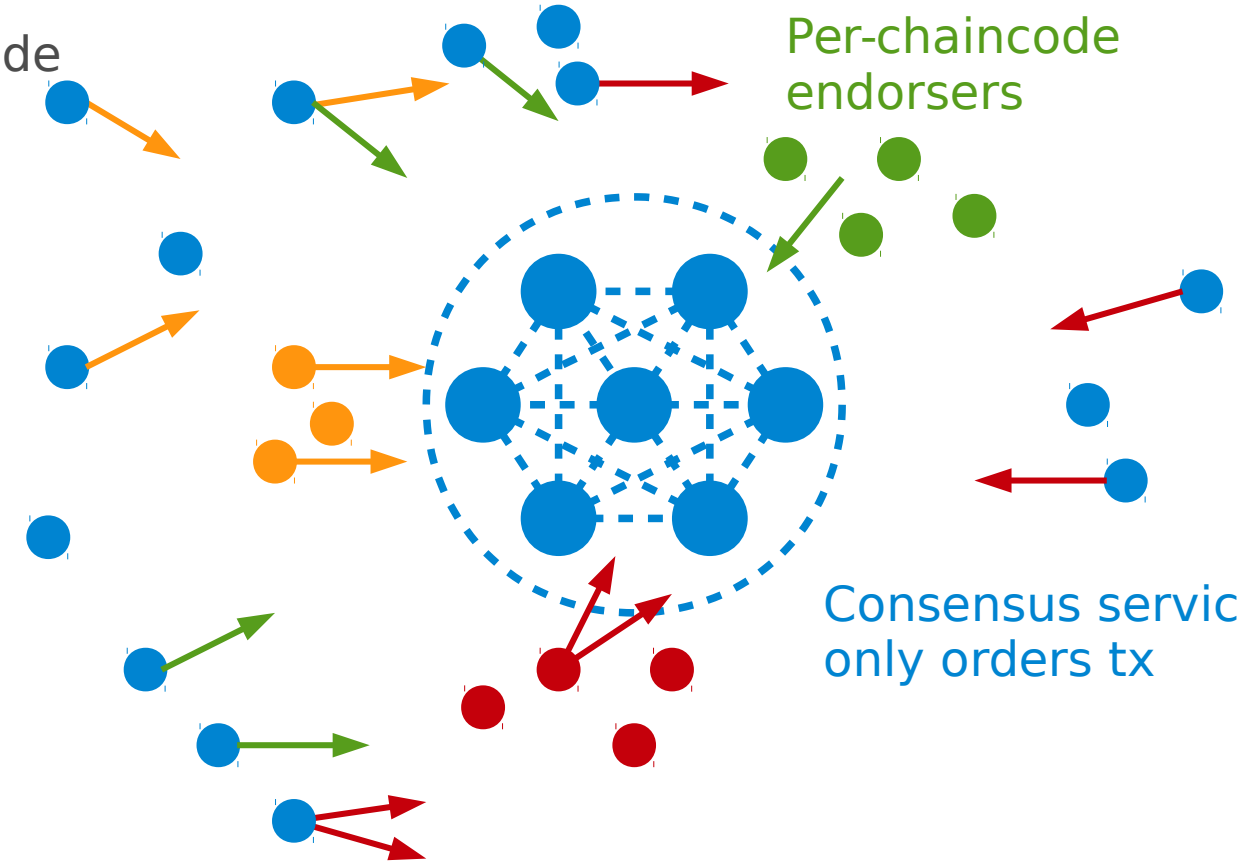
- Technical details
  - Implemented in GO
  - Runs smart contracts or "chaincode" within Docker containers
  - Transactions Deploy new chaincode / Invoke an operation / Read state
  - Implements consortium blockchain using traditional consensus (BFT, ZooKeeper)

Hyperledger Fabric V1

- Separate the functions of nodes into endorsers and consensus nodes
  - Every chaincode may have different endorsers
  - Endorsers have state, run tx, and validate tx for their chaincode
  - Chaincode specifies endorsement policy
  - Consensus nodes order endorsed and already-validated tx
  - All peers apply all state changes in order, only for properly endorsed tx
- Functions as replicated database maintained by peers [PWSKA00, KJP10]
  - Replication via (BFT) atomic broadcast in consensus
  - Endorsement protects against unauthorized updates
- Scales better – only few nodes execute, independent computations in parallel
- Permits some confidential data on blockchain via partitioning state
  - Data seen only by endorsers assigned to run that chaincode

Separation of endorsement from consensus

- Validation is by chaincode
- Dedicated endorsers per chaincode
- Consensus service
  - Only communication
  - Pub/sub messaging
  - Ordering for endorsed tx
- State and hash chain are common
  - State may be encrypted



Transactions in Fabric V1

- Client
  - Produces a tx (operation) for some chaincode (smart contract)
- Submitter peer
  - Execute/simulates tx with chaincode
  - Records state values accessed, but does not change state → readset/writeset
- Endorsing peer
  - Re-executes tx with chaincode and verifies readset/writeset
  - Endorses tx with a signature on readset/writeset
- Consensus service
  - Receives endorsed tx, orders them, and outputs stream of "raw" tx (=atomic broadcast)
- All peers
  - Disseminate tx stream from consensus service with p2p communication (gossip)
  - Filter out the not properly endorsed tx, according to chaincode endorsement policy
  - Execute state changes from readset/writeset of valid tx, in order

Modular consensus in Fabric V1

- "Solo orderer"
  - One host only, acting as specification during development (ideal functionality)
- Apache Kafka, a distributed pub/sub streaming platform
  - Tolerates crashes among member nodes, resilience from Apache Zookeeper inside
  - Focus on high throughput
- SBFT - Simple implementation of PBFT (6/2017 - under development)
  - Tolerates f < n/3 Byzantine faulty nodes among n
  - Focus on resilience

Conclusion

Conclusion

- Blockchain enables new trust models
- Many interesting technologies
  - Distributed computing for consensus
  - Cryptography for integrity, privacy, anonymity
- We are only at the beginning
- Blockchain = Distributing trust over the Internet
  - www.hyperledger.org
  - www.ibm.com/blockchain/
  - www.research.ibm.com/blockchain/

References

[AGK+15] P.-L. Aublin, R. Guerraoui, N. Knezevic, V. Quéma, M. Vukolic: The Next 700 BFT Protocols. ACM TOCS, 32(4), 2015.

[BCG+14] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza: Zerocash: Decentralized Anonymous Payments from Bitcoin. IEEE S&P 2014.

[CKPS01] C. Cachin, K. Kursawe, F. Petzold, V. Shoup: Secure and Efficient Asynchronous Broadcast Protocols. CRYPTO 2001.

[CGR11] C. Cachin, R. Guerraoui, L. Rodrigues: Introduction to Reliable and Secure Distributed Programming (2. ed.). Springer, 2011.

[CSV16] C. Cachin, S. Schubert, M. Vukolic: Non-determinism in Byzantine Fault-Tolerant Replication. OPODIS 2016.

[CL02] M. Castro, B. Liskov: Practical Byzantine fault tolerance and proactive recovery. ACM TOCS, 20(4), 2002.

[DSW16] C. Decker, J. Seidel, R. Wattenhofer: Bitcoin meets strong consistency. ICDCN 2016.

[EGS+16] I. Eyal, A. Gencer, E.G. Sirer, R. van Renesse: Bitcoin-NG: A Scalable Blockchain Protocol. NSDI 2016.

References

[KMS+16] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou: Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. IEEE S&P 2016.

[LNB+15] L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert, P. Saxena: A Secure Sharding Protocol For Open Blockchains. ACM CCS 2016.

[MR98] D. Malkhi, M. Reiter: Byzantine Quorum Systems. Distributed Computing, 1998.

[MXC+16] A. Miller, Y. Xia, K. Croman, E. Shi, D. Song: The Honey Badger of BFT Protocols. ACM CCS 2016.

[V16] M. Vukolic: The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. LNCS 9591, Proc. iNetSeC 2015.

Hyperledger Fabric references

- www.hyperledger.org
- Docs – hyperledger-fabric.readthedocs.io/en/latest/
- Chat – chat.hyperledger.org, all channels like #fabric-*
- Designs – wiki.hyperledger.org/community/fabric-design-docs
- Architecture of V1 – github.com/hyperledger/fabric/blob/master/proposals/r1/Next-Consensus-Architecture-Proposal.md
- Code – github.com/hyperledger/fabric