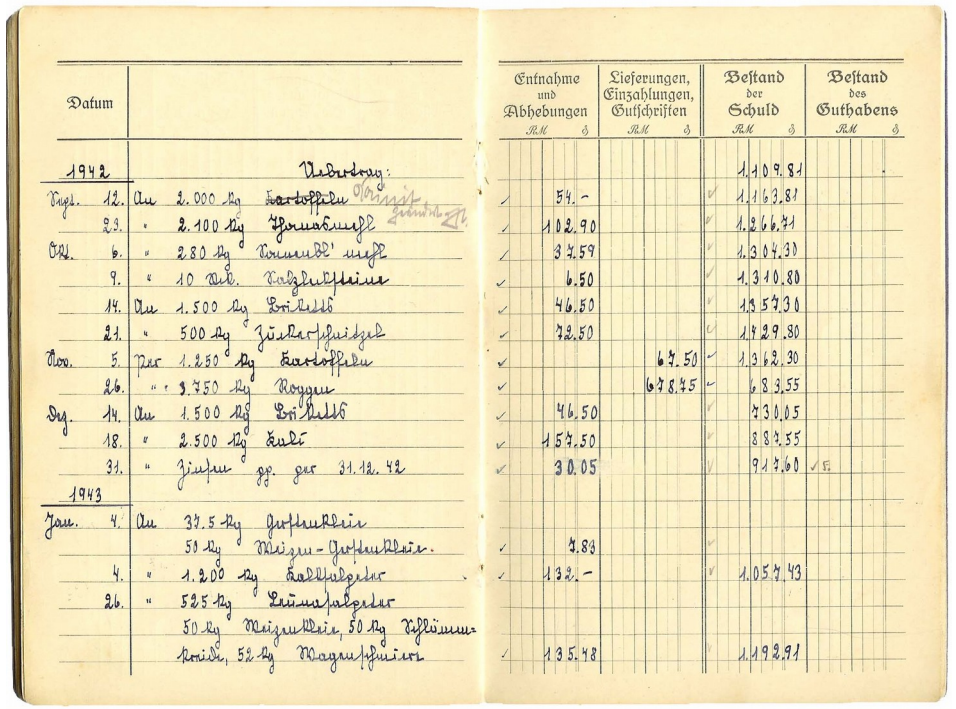
Blockchain, cryptography, and consensus

Christian Cachin
(Elli Androulaki, Angelo De Caro, Mike Osborne, Simon Schubert, Alessandro Sorniotti, Marko Vukolic, Thomas Weigold ...)

IBM Research – Zurich

October 2016

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	
1943 Jan. 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 Futterschrot	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 Kohle	157.50		887.55	
31.	" Zinsen pp. per 31.12.42	30.05		917.60	
1943 Jan. 4.	An 37.5 kg Getreideschrot				
	50 kg Weizen-Getreideschrot	7.83			
4.	" 1.200 kg Kohlenbriketts	132.–		1.057.43	
26.	" 525 kg Futtermischung				
	50 kg Weizenkleie, 50 kg Hülsenfrüchte, 52 kg Magnesiasteine	135.48		1.192.91	

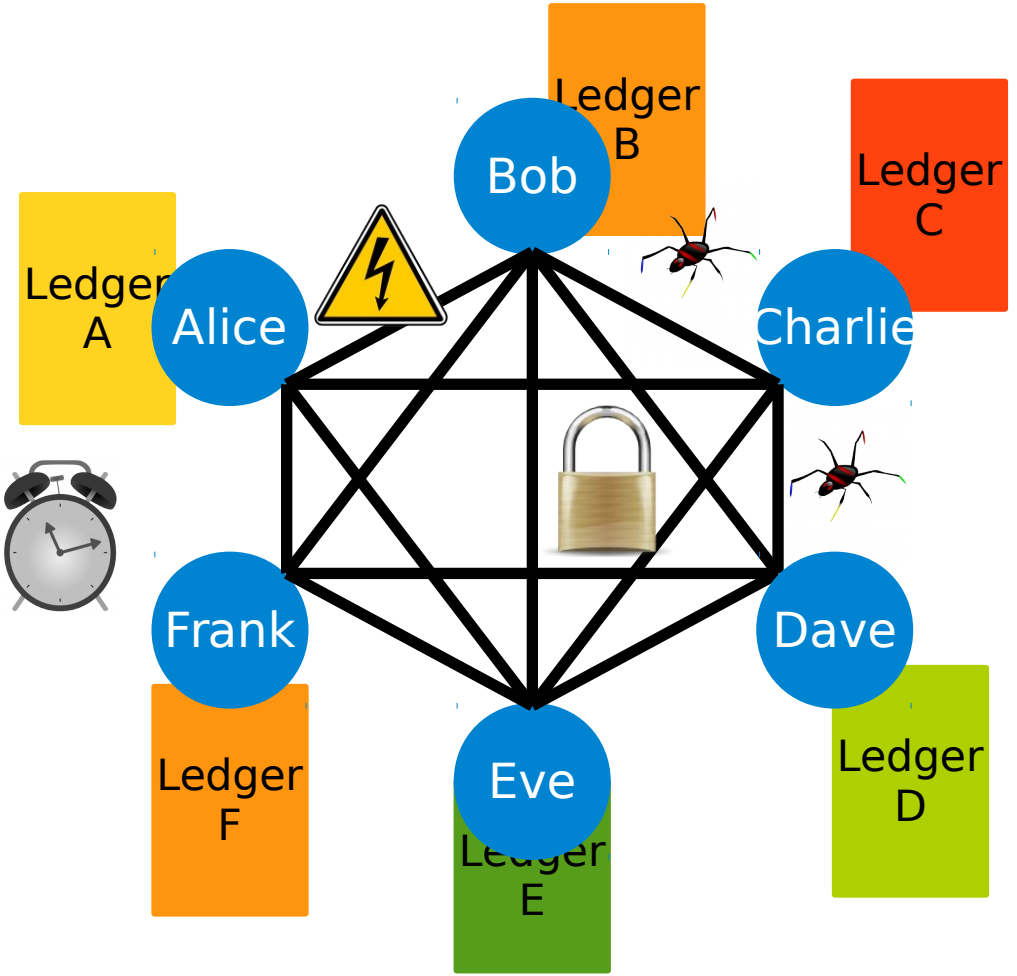
§ Ledger records all business activity as transactions
  – Databases

§ 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

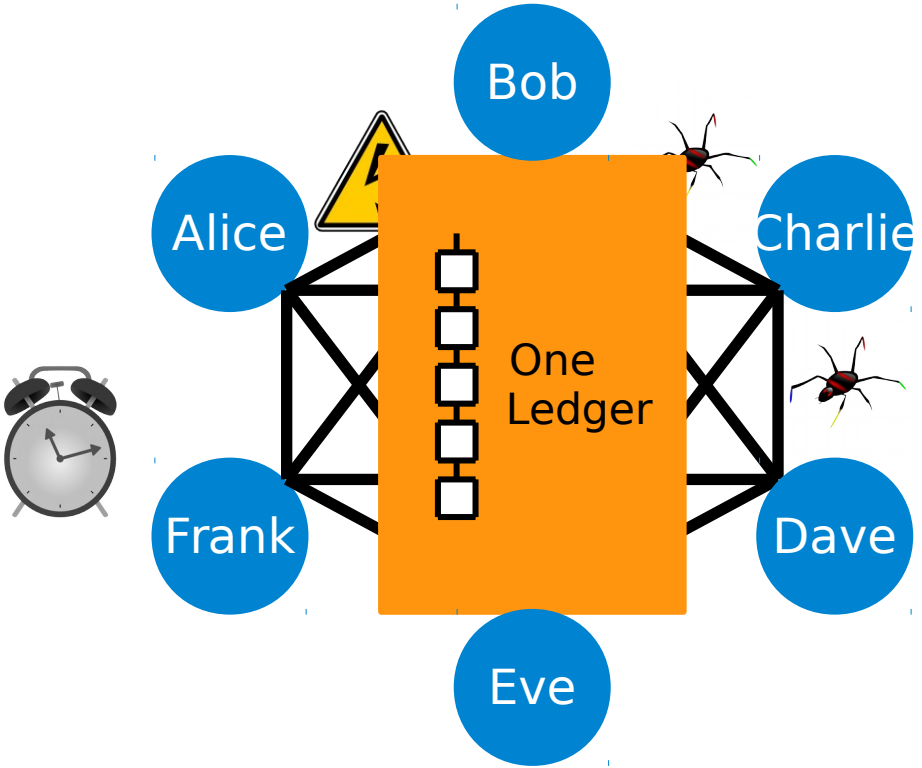


§ Every party keeps its own ledger and state

§ 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

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

Why blockchain now?

§ Cryptography has been a key technology in the financial world for decades
  – Payment networks, smart cards, online banking ...

§ Trust model of (financial) business has not changed
  – Trusted intermediary needed for exchange among non-trusting partners
  – Today cryptography mostly secures point-to-point interactions

§ Bitcoin started in 2009
  – Embodies only cryptography of 1990s and earlier
  – First prominent use of cryptography for generalized trust model (= trust no entity)

§ The promise of Blockchain – Reduce trust and replace it by technology
  – Exploit advanced cryptographic techniques

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

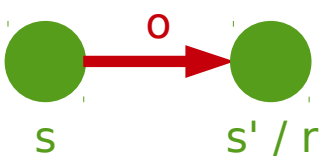
This paper describes an architecture for secure and fault-tolerant service replication in an asynchronous network such as the Internet, where a malicious adversary may corrupt some servers and control the network. It relies on recent protocols for randomized Byzantine agreement and for atomic broadcast, 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

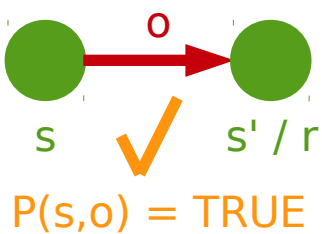
– Operation o transforms a state s to new state s' and may generate a response r

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



§ Validation condition

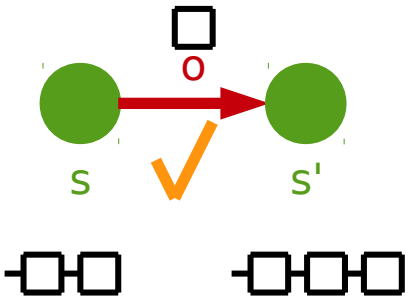
– Operation needs to be valid, in current state, according to a predicate P()



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

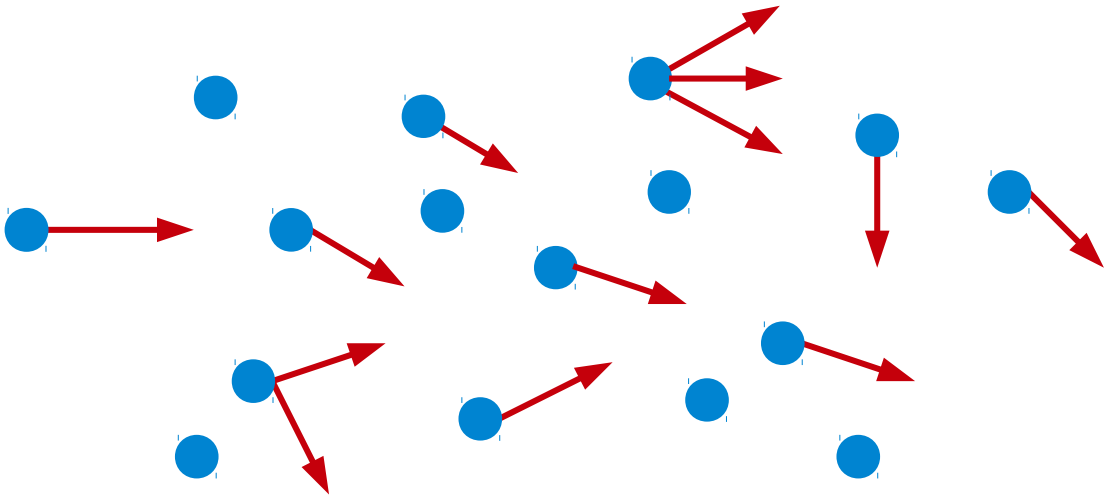
§ Bitcoins are unforgeable bitstrings
  – "Mined" by the protocol itself (see later)

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

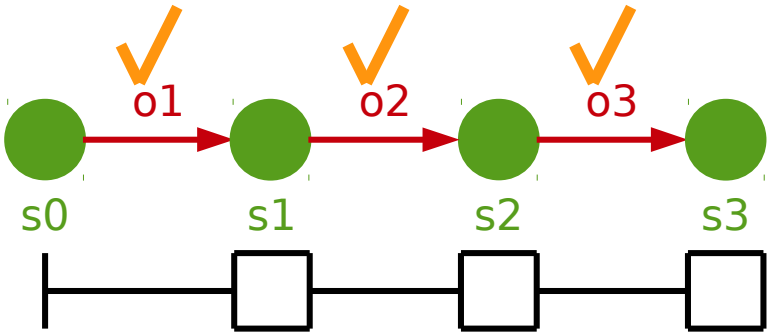
§ Every transaction transfers a bitcoin (fraction) from current to next owner
  – "This bitcoin now belongs to 3JDs..." signed by the key of current owner
  – (Flow linkable by protocol, and not anonymous when converted to real-world assets)

§ 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 ...

From cryptocurrency to blockchain

§ Cryptocurrencies = single-purpose
- Only one application: Bitcoin, Ripple, alt-coin ...
- Decentralized (permissionless), censorship-resistant, no authority beyond code

§ Blockchain = programmable
- Smart contracts, fully programmable
- With a cryptocurrency (Ethereum ...) or without (Hyperledger/fabric ...)
- Often in consortium model (permissioned)

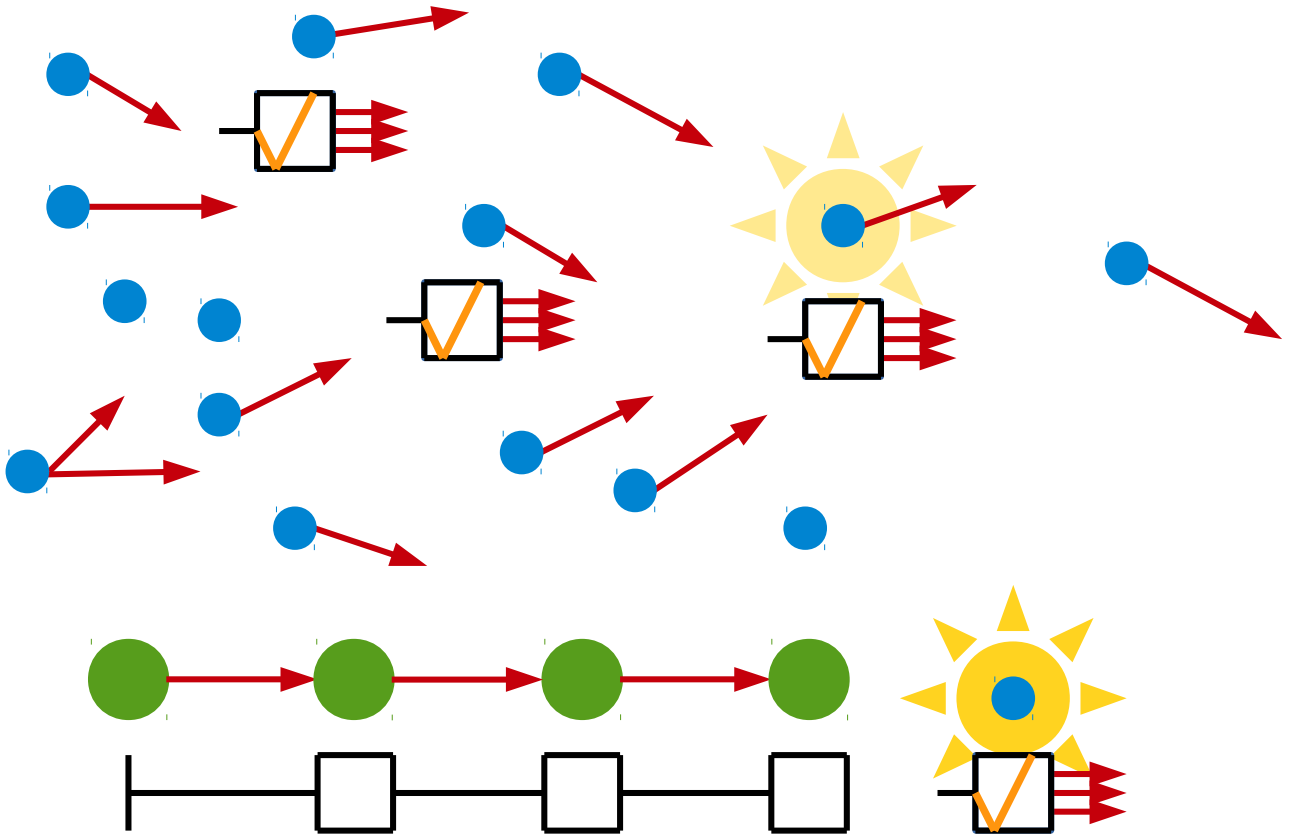
Consensus

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, 250-500MW, 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")

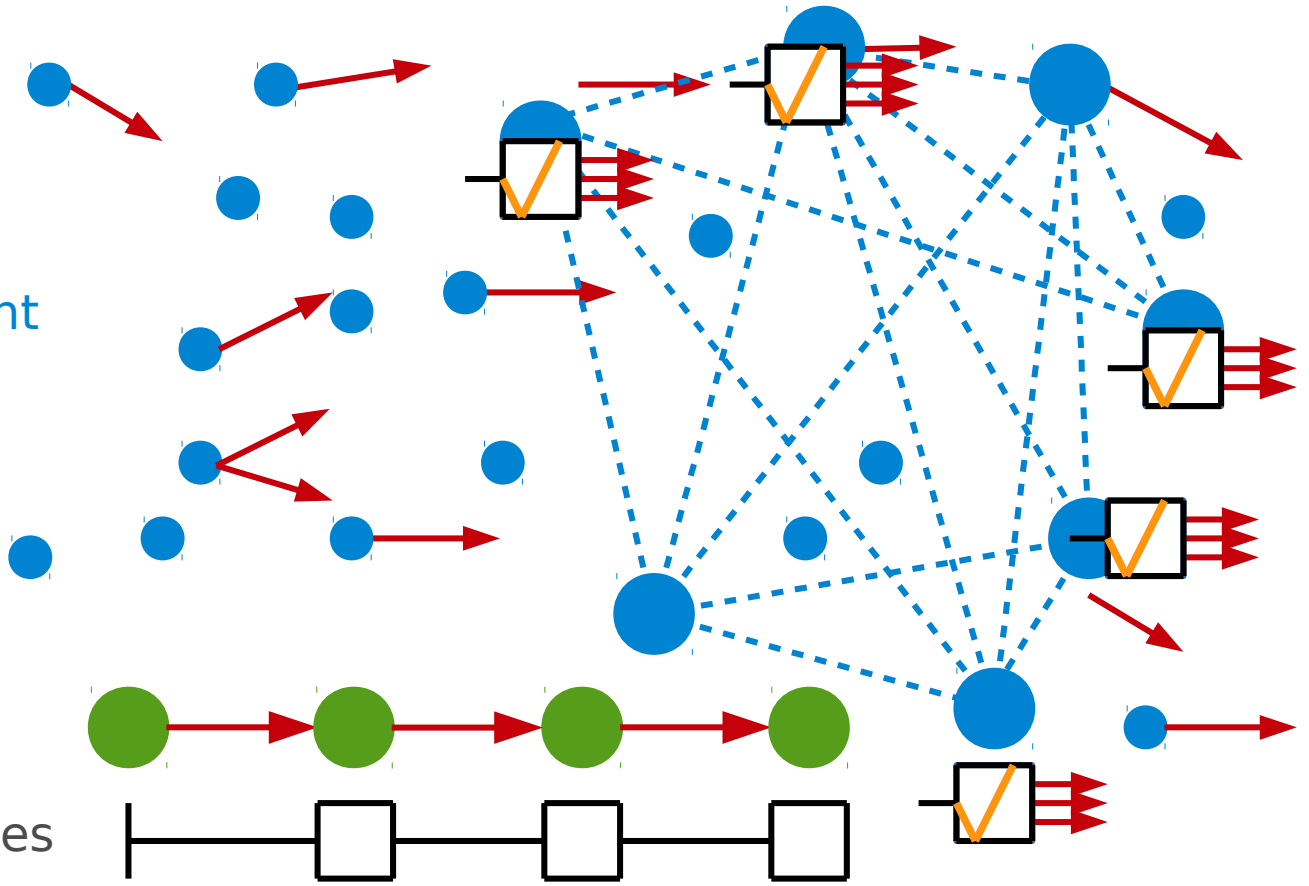
Consortium consensus (BFT, Hyperledger)

§ 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



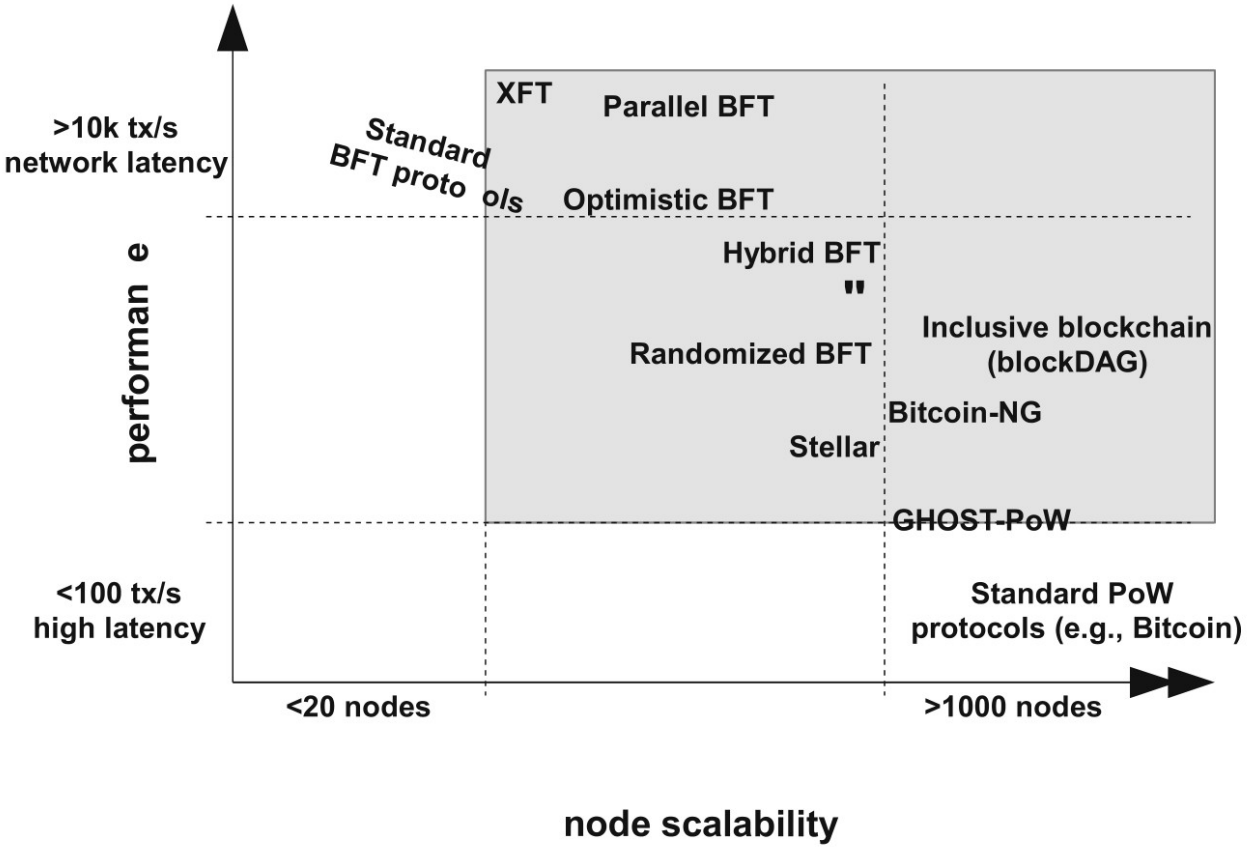
Consortium consensus = permissioned

§ Central entity controls group membership
- Dynamic membership changes in protocol
- Membership may be decided inline, by protocol itself

§ Features
- Byzantine agreement / Byzantine Fault Tolerance (BFT) [Lamport et al., ca. 1980]
- BFT consensus is a very-well understood problem
  - Clear assumptions and top-down design
  - Textbooks [CGR11 => www.distributedprogramming.net]
  - Open-source implementations (BFT-SMaRT => github.com/bft-smart/library)
- Many systems already provide crash tolerant consensus (Chubby, Zookeeper, etcd ...)
- Usually takes $\Omega(n^2)$ communication (OK for 10-100 nodes, not > 1000s)

§ Revival of research in BFT protocols
- Focus on scalability and communication efficiency

Scalability–performance tradeoff



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

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 (85GB as of 10/2016)

Validation of transactions – BFT protocols

§ Every node among the group executes the same transactions
  – State machine replication paradigm

§ Well-known properties of Byzantine consensus
  – Validity, Agreement, Termination ... from textbooks

§ Recent work: partition state among nodes for improved scalability
  – E.g., "next consensus architecture" of Hyperledger fabric

Public validation vs. private state

§ So far everything on blockchain is public – where has privacy gone?

§ Use advanced cryptography – keep state "off-chain" and produce verifiable tx

– In Bitcoin, verification is a digital signature by key that owns coin
  • Keys are pseudonyms, tx can be linked

– More privacy with zero-knowledge proofs, verified by P (ZeroCash …)
  • Anonymous credentials for unlinkable tx

– For smart contracts with cryptographic verifiable computation (VC) (Hawk …)
  • P checks correctness of proof for VC

Open issues

Do we have consensus?

§ Cost of bitcoin mining rising
  – Seems inherent for truly decentralized permissionless

§ How to scale to 1'000s or 1'000'000s of nodes?
  – Throughput versus scalability versus consistency

§ Strict consistency for wide-area cloud service?
  – Cloud providers have not solved it either …!

§ Revived research on consensus protocols
  – Workshop on Bitcoin and Blockchain Research, 2016 (fc16.ifca.ai/bitcoin/)
  – Distributed Cryptocurrencies and Consensus Ledgers, 2016 (www.zurich.ibm.com/dccl/)

Privacy versus transparency – or both?

§ Privacy vs. accountability

§ Auditability, transparency, and verifiability

§ Total anonymity or limited privacy?

- Cryptographic protocols can deliver both
- Integrate blockchain networks with real world (countries, borders, regulation…)
- Adjust technology to real-world constraints

§ Finding efficient cryptographic tools

- Protocols developed in research are seen as too expensive
- Home-grown cryptographic protocols are often "snake oil"

§ Many opportunities for cryptographers

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

Questions?

Christian Cachin

IBM Research – Zurich
www.zurich.ibm.com/~cca/

www.ibm.com/blockchain/

www.hyperledger.org