Hyperledger Fabric V1

Christian Cachin
(with many others, at IBM Zurich & elsewhere; special thanks to Marko Vukolic)

IBM Research – Zurich

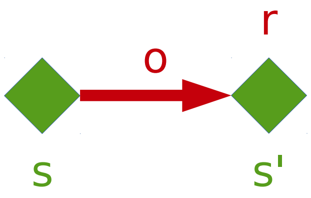
October 2017

What is a blockchain?

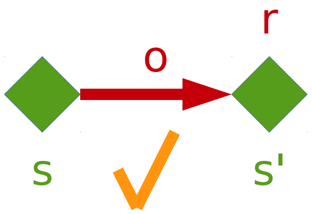
A replicated state machine (RSM) ...

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

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



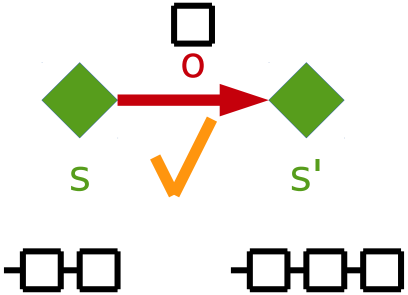
- Validation condition
  - Operation needs to be valid, in current state, according to a predicate P()



P(s,o) = TRUE

RSM with a hash chain → blockchain

- 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

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

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"

Hyperledger Fabric

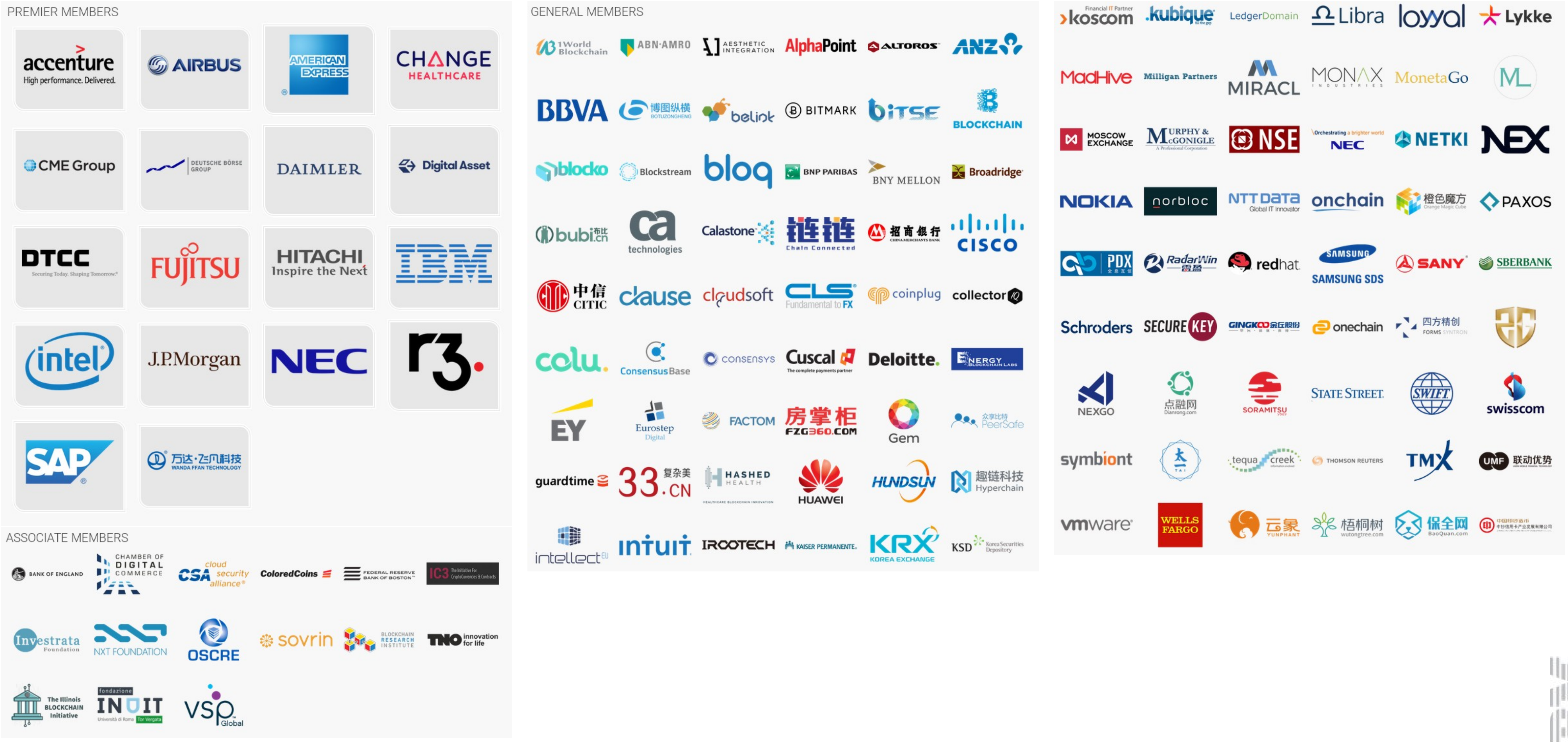
Hyperledger

- 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
  - ca. 160 members in Sep. '17

- Incubates and promotes blockchain technologies for business
- Today 5 frameworks and 3 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

Some Hyperledger members …



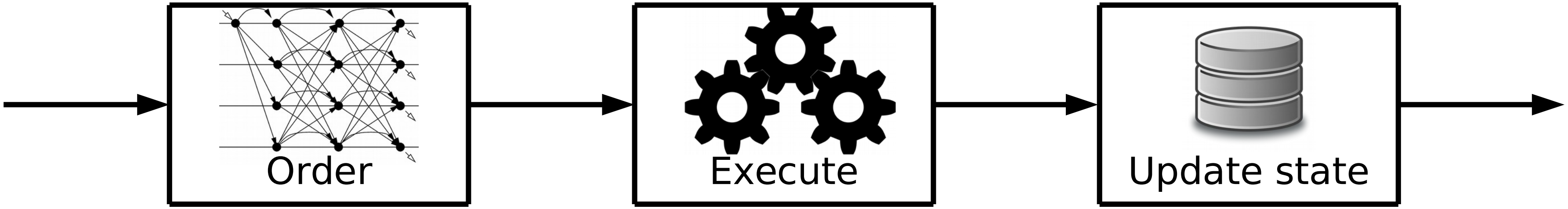
Hyperledger Fabric

- Blockchain fabric and distributed ledger framework for business
  - One of multiple blockchain platforms in the Hyperledger Project
  - First "active" platform in Hyperledger project (Mar. '17)
  - First "production-ready" platform (Jul. '17)

- 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
  - Programmable, replicated, sharded blockchain state machine; implemented in GO
  - Runs smart contracts or "chaincode" within Docker containers
  - Implements consortium blockchain using traditional consensus (BFT, Kafka/ZooKeeper)

Traditional RSM architecture



- Consensus protocol
- Deterministic (!) execution
- Persistent state changes

- All prior BFT systems operate like this, starting with PBFT
- All prior permissioned blockchain systems operate like this [Schneider '90]
  - Including Hyperledger Fabric until V0.6

Problems with the traditional architecture

- Sequential execution
  - Increased latency – or – complex schemes for parallelism

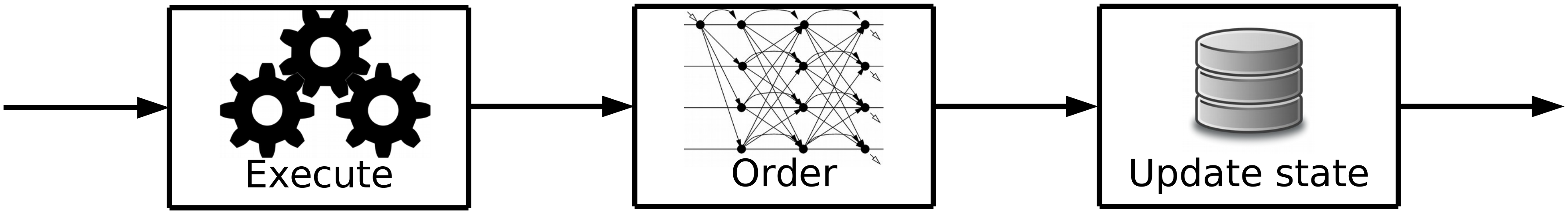
- Non-deterministic operations
  - Difficult to enforce with generic programming language (difficult per se!)
  - Modular filtering of non-deterministic op. is costly [C-Schubert-Vukolic, OPODIS '16]

- Trust model is fixed for all applications/smart contracts
  - Typically (f+1) validator nodes must agree to result (at least one correct)
  - Fixed to be the same as in consensus protocol

- Data proliferation, concerns about privacy
  - All nodes execute all applications

- All these are lessons learned from Hyperledger Fabric, before V0.6

Fabric V1 architecture



- Simulate op. and endorse
- RW-set
- Nodes differ per application

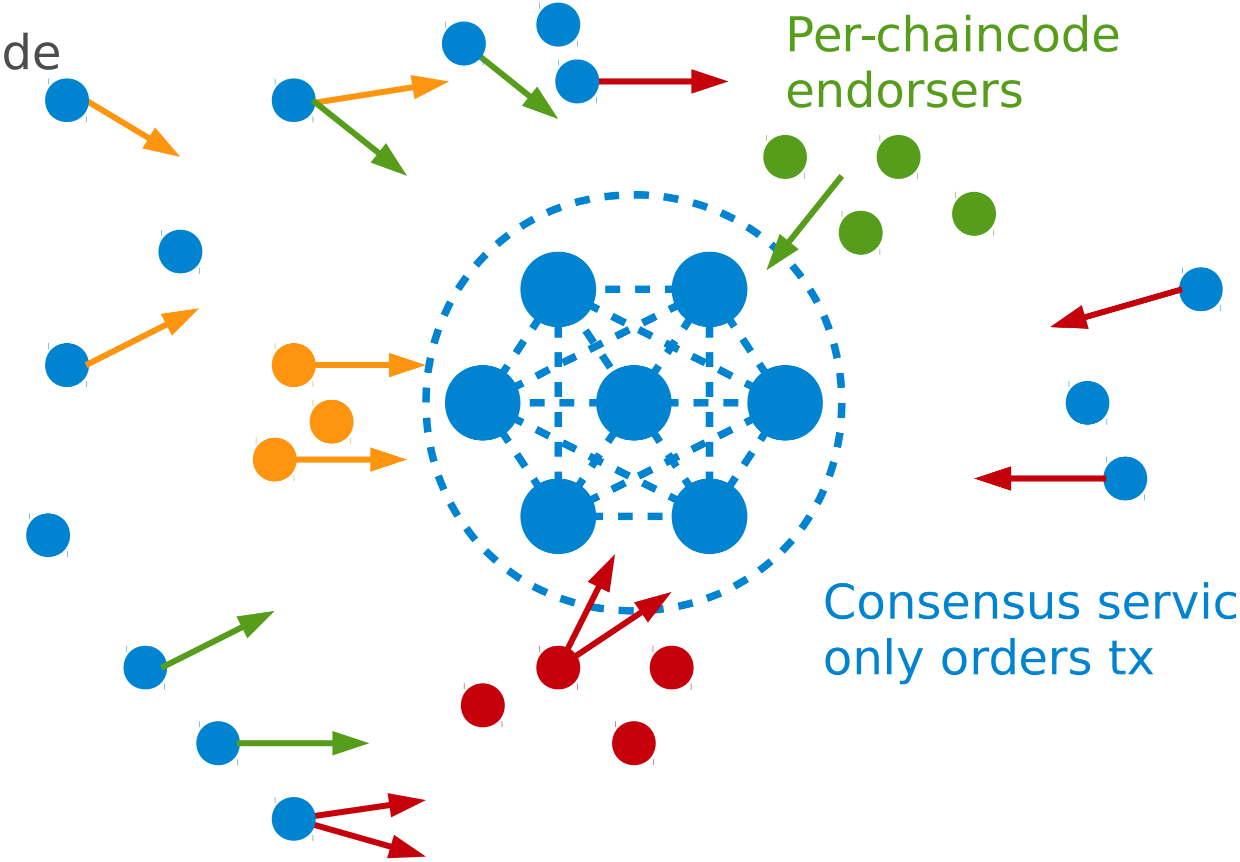
- Order RW-sets
- Stateless consensus service

- Validate RW-sets
- Eliminate conflicting ops.
- State kept by all nodes

- Reminiscent of middleware-replicated databases [Kemme-Jiménez-Patiño, '10]
- Appropriate for BFT model

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



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
  - 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

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
- BFT-SMaRt - Research prototype
  - Tolerates f < n/3 Byzantine faulty nodes among n
  - Demonstration of functionality
- SBFT - Simple implementation of PBFT (currently under development)
  - Tolerates f < n/3 Byzantine faulty nodes among n
  - Focus on resilience

Hyperledger Fabric V1 - Skipped aspects

- Further important components
  - Organizations, Membership service providers (MSP), and Certification Authorities (CA)
  - Chaincode syntax (GO)
  - Gossip protocols for dissemination
  - Channels
  - Data format and ledger design (LevelDB)

- Most important
  - Industrial software engineering
  - Production release V1.0 in July '17

Conclusion

- Blockchain = Distributing trust over the Internet

- Many new models, applications, protocols …
  - Cryptography
  - Distributed computing

- This is only the beginning

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

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