Initial Public Offering (IPO) on Permissioned Blockchain using Secure Multiparty Computation

Fabrice Benhamouda , **Angelo De Caro** , Shai Halevi, Tzipora Halevi, Charanjit jutla, Yacov Manevichk, and Qi Zhang

IBM

Outline

- Hyperledger and Fabric
- Fabric Architecture
- Initial Public Offering and Multi-Party Computation

Hyperledger and Fabric

HYPERLEDGER

- **Hyperledger – www.hyperledger.org**
  - Global collaboration hosted by the Linux Foundation
    - Advances blockchain technologies for business, neutral, community-driven
    - Started in 2016: Hyperledger unites industry leaders to advance blockchain technology
    - ca. 230 members in May '18
  - Develops and promotes blockchain technologies for business
  - Hyperledger has 5 frameworks and 5 tools, hundreds of contributors

HYPERLEDGER FABRIC

- **Hyperledger Fabric – github.com/hyperledger/fabric/**
  - A generic blockchain framework, modular, consortium
  - Originally contributed by IBM and DAH
  - Architecture, consensus, and cryptography contributed by IBM Research - Zurich

Hyperledger Overview

Hyperledger Modular Greenhouse Approach

Infrastructure

Technical, Legal, Marketing, Organizational

Ecosystems that accelerate open development and commercial adoption

THE LINUX FOUNDATION

Cloud Foundry | Node.js | HYPERLEDGER | Open Container Initiative

Frameworks

Meaningfully differentiated approaches to business blockchain frameworks developed by a growing community of communities

- HYPERLEDGER FABRIC — Permissioned with channel support
- HYPERLEDGER SAWTOOTH — Permissioned & permissionless support
- HYPERLEDGER IROHA — Mobile application focus
- HYPERLEDGER INDY — Decentralized identity
- HYPERLEDGER BURROW — Permissionable smart contract machine

Tools

Typically built for one framework, and through common license and community of communities approach, ported to other frameworks

- HYPERLEDGER COMPOSER — Model and build blockchain networks
- HYPERLEDGER CELLO — As-a-service deployment
- HYPERLEDGER EXPLORER — View and explore data on the blockchain
- HYPERLEDGER QUILT — Ledger interoperability
- HYPERLEDGER CALIPER — Blockchain framework benchmark platform

Fabric Architecture

HYPERLEDGER FABRIC In a Nutshell

Permissioned

- **Strong identity management**
- Support for **multiple credential** and cryptographic services for identity
- Support for **”bring your own identity”**

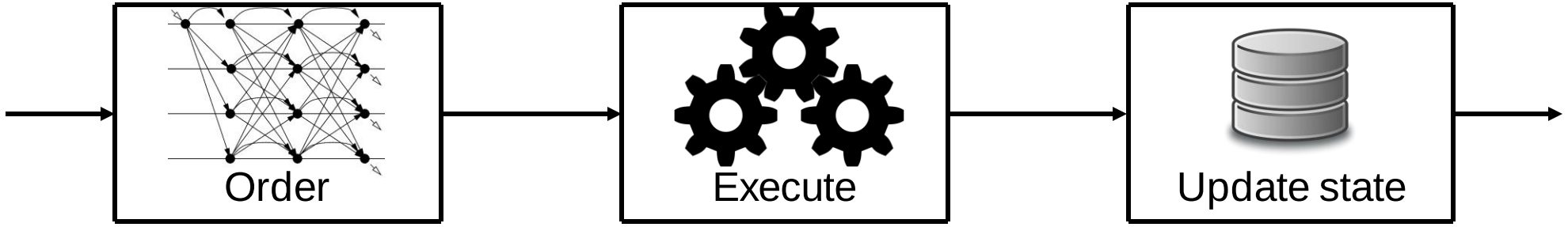
Privacy Friendly

- Support **broader regulatory requirements** for privacy and confidentiality
- **Contract state concealable to unauthorized parties**
- Business Logic is executed only after authorized entity request and only on a subset of the netwrok

Scalable

- **Scale the number of participants and transaction** throughput
- **Eliminate non deterministic transactions**
- **Parallel execution of the business logic**

Traditional design: Replicated State Machine



- Consensus or atomic broadcast
- Deterministic (!) tx execution
- Persist state on all peers

- All prior BFT systems operate like this [S90]
- All prior permissioned blockchains operate like this
  - Including Hyperledger Fabric until V0.6

Issues with the traditional replication design

Sequential execution

- Increased latency – or – complex schemes for parallelism

Operations must be deterministic

- Difficult to enforce with generic programming language (difficult per se!)
- Modular filtering of non-deterministic operations is costly [CSV16]

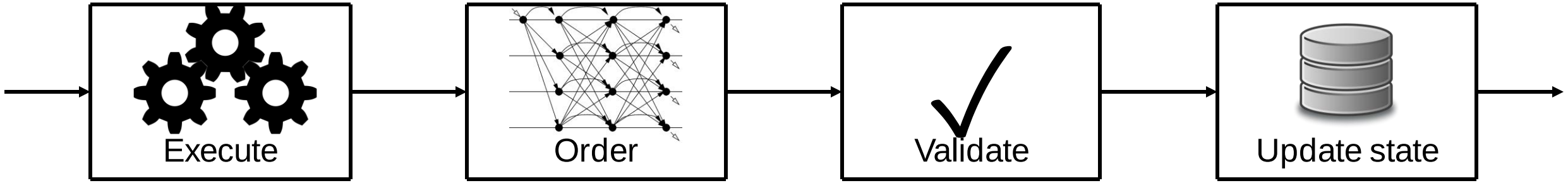
Trust model is fixed for all applications (smart contracts)

- Typically some (F+1) validator nodes must agree to result (at least one correct)
- Fixed to be the same as in consensus protocol

Privacy is difficult, as data spreads to all nodes

- All nodes execute all applications

Fabric Unique Architecture Scales



- Simulate tx and endorse
- Create rw-set
- Collect endorse-ments

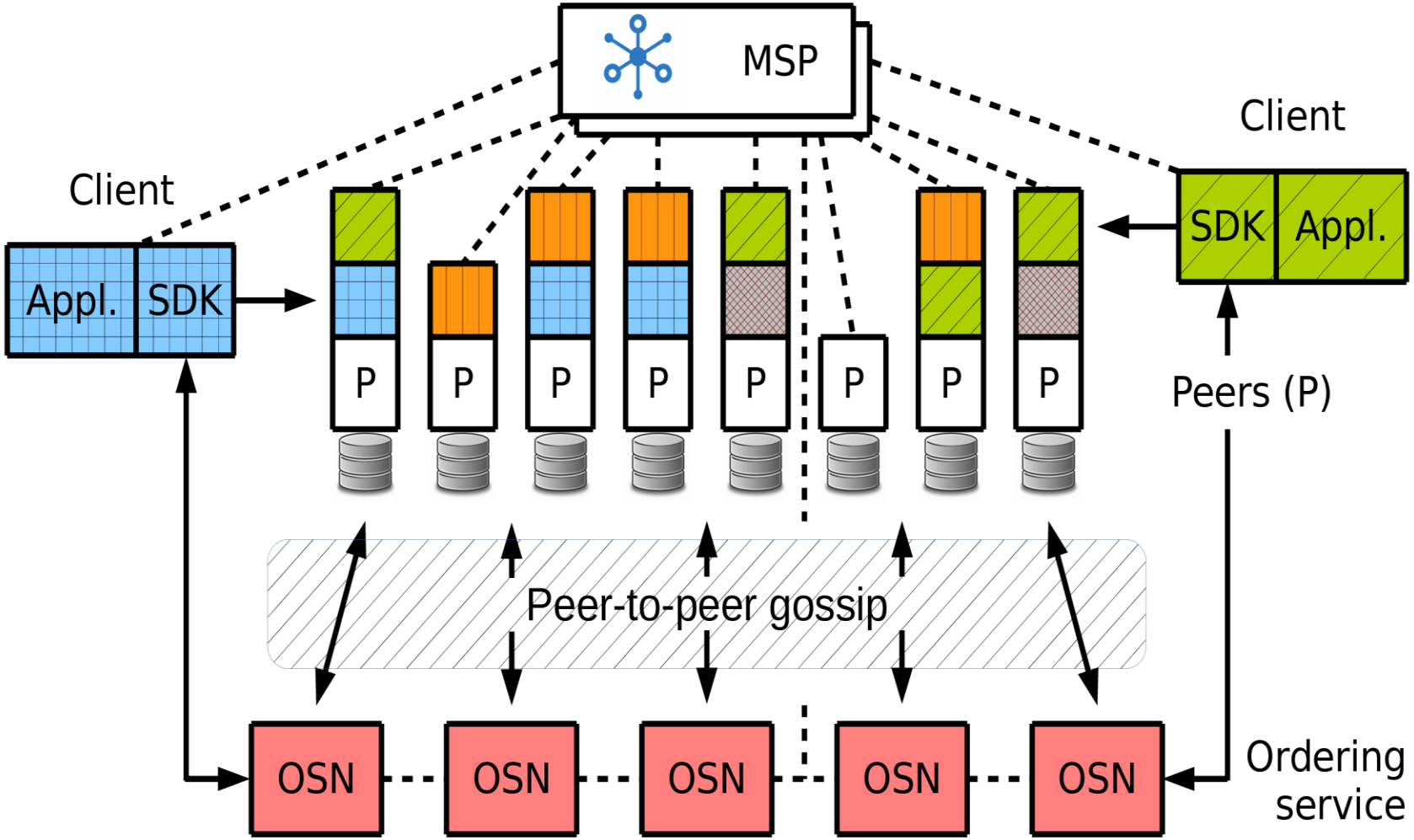
- Order rw-sets
- Atomic broadcast (consensus)
- Stateless ordering service

- Validate endorse-ments & rw-sets
- Eliminate invalid and conflicting tx

- Persist state on all peers

- Includes techniques from databases
- Extends a middleware-replicated database [KJP10] to BFT model

Fabric Architecture



Security First!

Strong identity management

Selective participation to authorized users

Accountability Non-repudiation

Entities are accounted for the transactions they create, cannot forge others' transactions

Modular, easily extensible, "bring your on provider" membership architecture

Privacy / Access Control

Contract state concealable to unauthorized parties

Authorized Execution

Logic is executed only after authorized entity request

Access Control Enforcement Framework

Privacy / Access Control

User activity & contract logic concealable to unauthorized entities

Secure Chaincode Availability Framework Application Libraries for Privacy

Pluggable Components

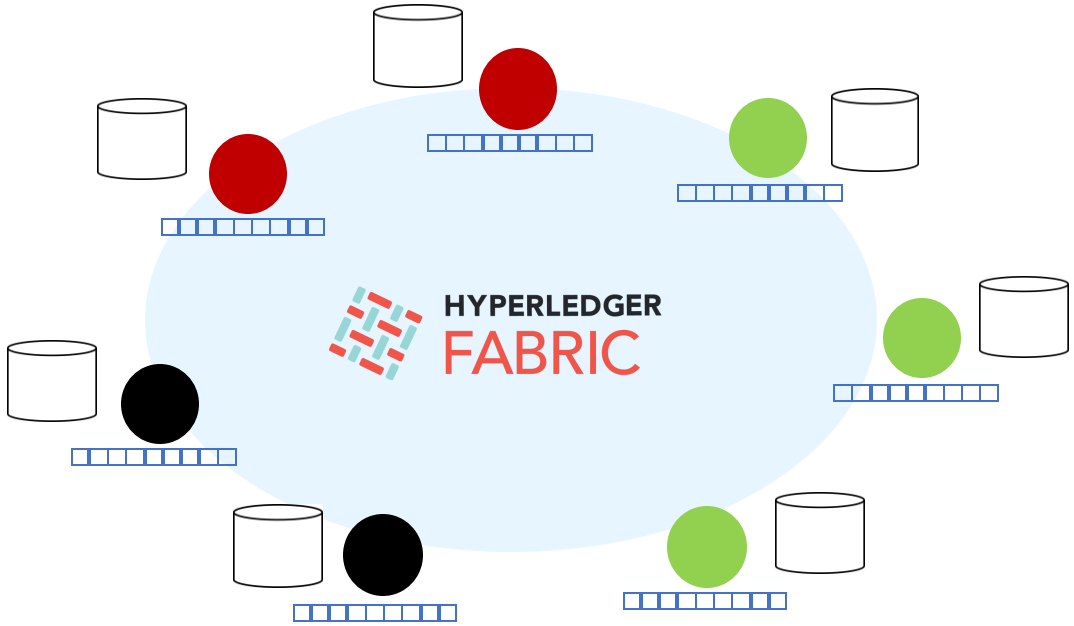
Compatibility with standards

Initial Public Offering (IPO) and Multi-Party Computation

Blockchain Can Revamp Initial Public Offering

- **IPO Trading** is an example of a ***clearing price auction***, where a single seller sells multiple shares at the same price to many buyers.
  - A bank lists it publicly on the ledger, specifying a unique ID.
  - Then, brokerage houses can record IPO orders on the ledger on behalf of investors.
  - Later the listing bank invokes the sell-IPO process, and the peers engage in a protocol to determine the clearing price of this IPO, as well as the share allocation

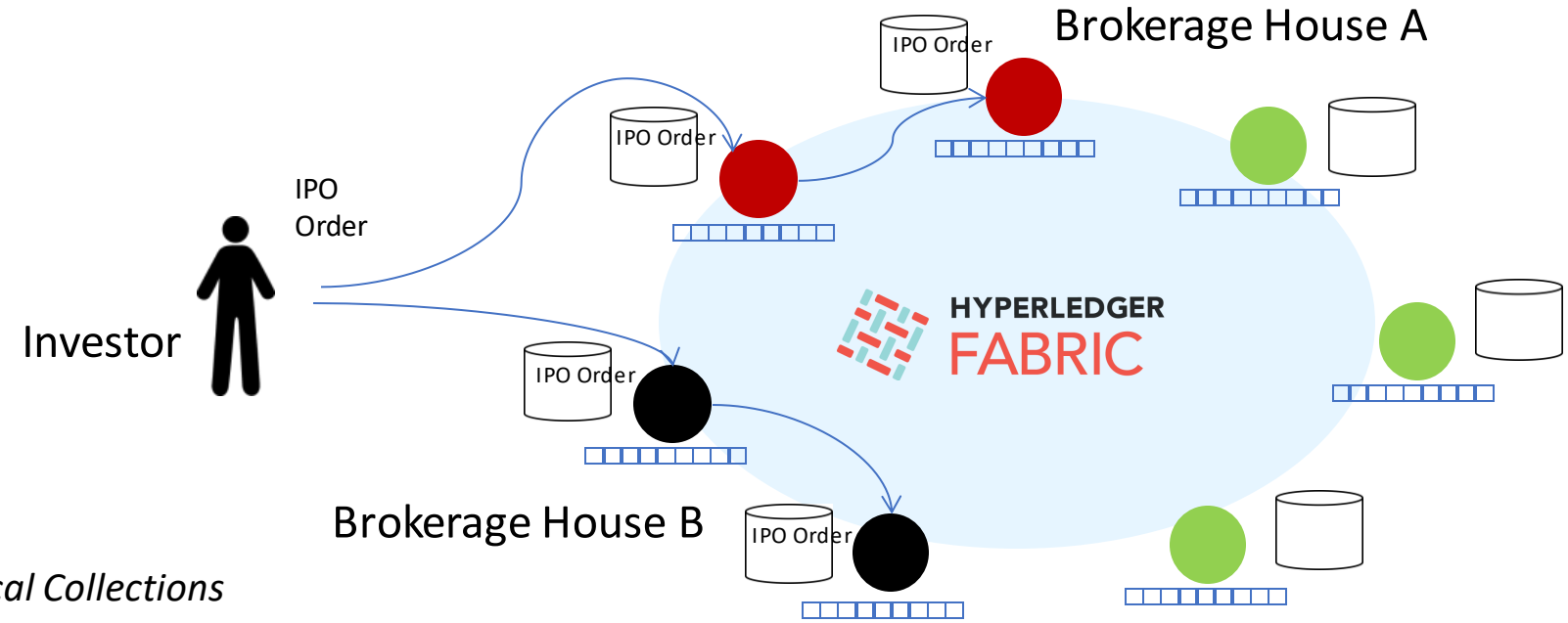
- The use of a **blockchain is highly beneficial**:
  - It provides **strong traceability and auditability**
  - **confidential orders** without having to rely on a trusted party.



IPO – A First Attempt using Fabric

IPO Trading is an example of a *clearing price auction*, where a single seller sells multiple shares at the same price to many buyers.

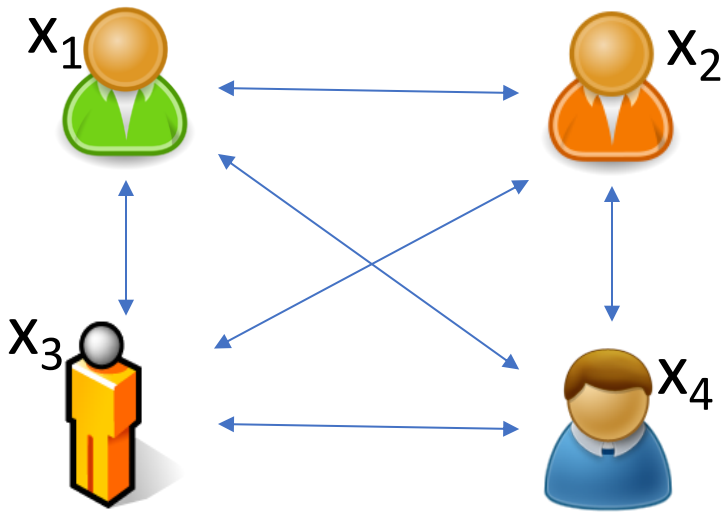
- A bank lists shares publicly.
- Then, brokerage houses records the **IPO orders** on behalf of investors. (**Confidentiality required**)
- Later the listing bank determines the clearing price of this IPO, as well as the share allocation. (**Settlement)**



(Confidentiality) *Local Collections*
(**Settlement**) *Tokens*

Secure Multi Party Computation(MPC)

- Cryptographic protocol for emulating a trusted party
  - In a system with no trusted parties
- $P_1, P_2, \ldots, P_n$ are mutually suspicious
  - Each with its own secret input $x_1, x_2, \ldots, x_n$
  - Want to compute a joint function $y=f(x_1, x_2, \ldots, x_n)$



Goal:

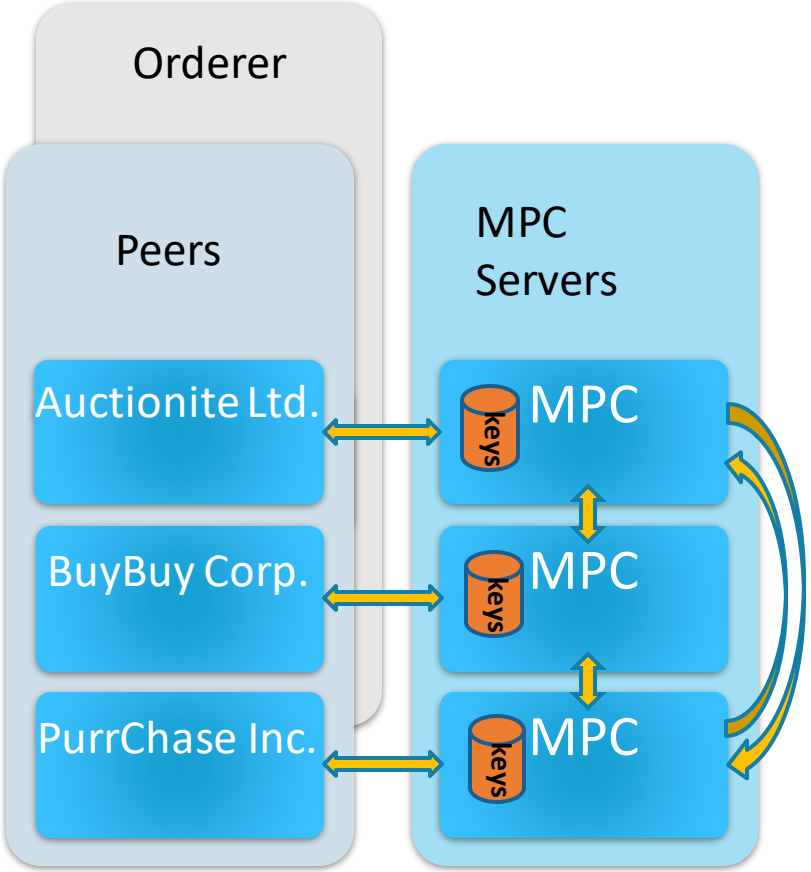
Correctness: Everyone computes $y=f(x_1,\ldots,x_n)$

Security: Nothing but the output is revealed

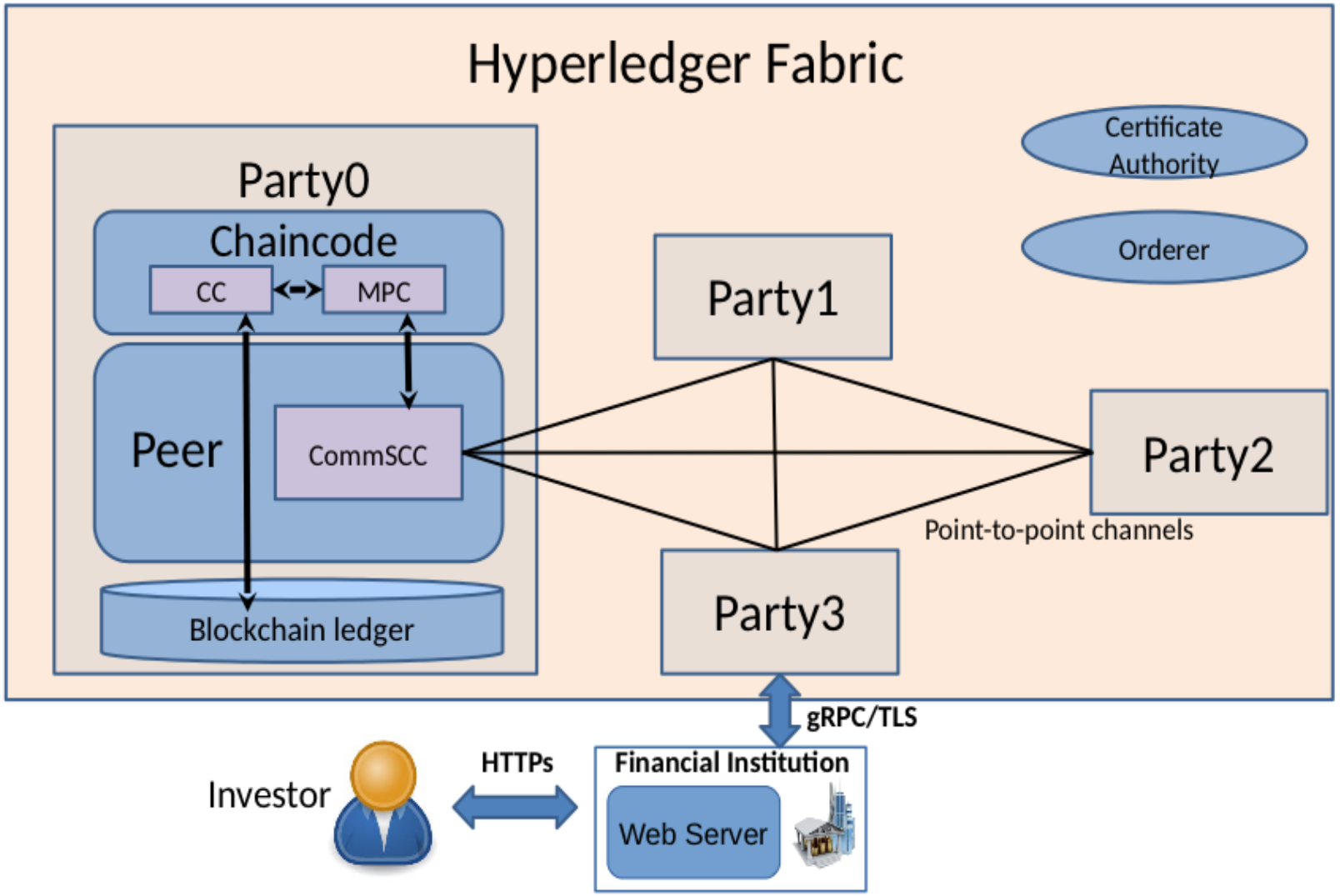
Multi-Party Computation Enables Decentralization and Privacy

- **Goal**: Enable private data that impacts transactions
  - *In current Fabric, transaction data is seen by everyone*
    - *At least, everyone who needs to endorse the transaction*
  - Private data support opens a whole new level of applications
    - Commerce: Purchase goes through if buyer has enough money
    - Shipping: Bidding on space for containers in a ship
    - Medical: Drug dispensed if client's condition warrants it
    - IoT: Aggregate recorded w/o revealing individual data
    - Audit: Action recorded when departments align their books
      - Without them having to share confidential data (e.g., Chinese wall)

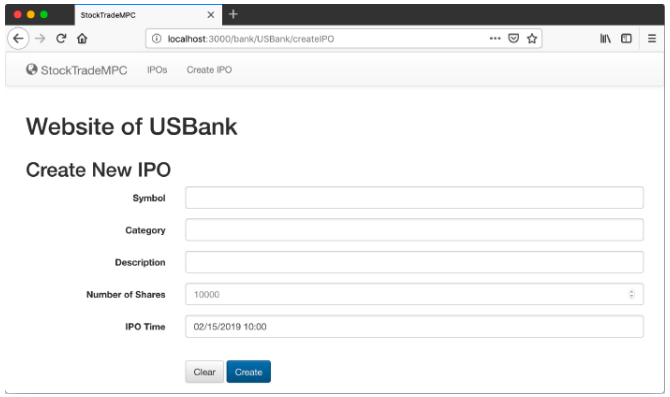
- **Solution**: Use secure Multi-Party Computation (MPC). An interactive protocol with multiple parties, each with private input. Computing the correct output, learning nothing more, audit later when needed.



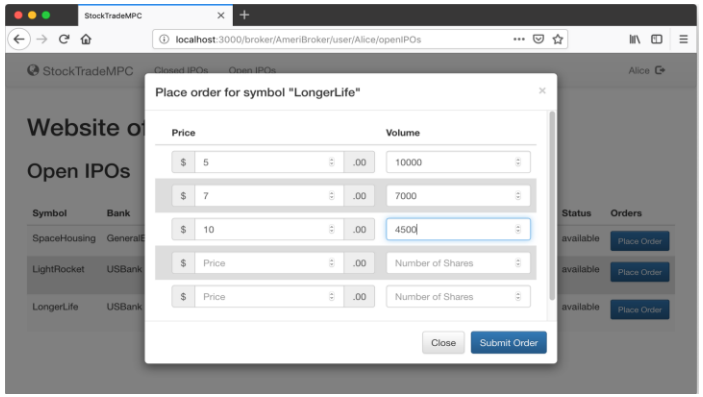
Fabric and MPC deliver **Auditable Privacy**



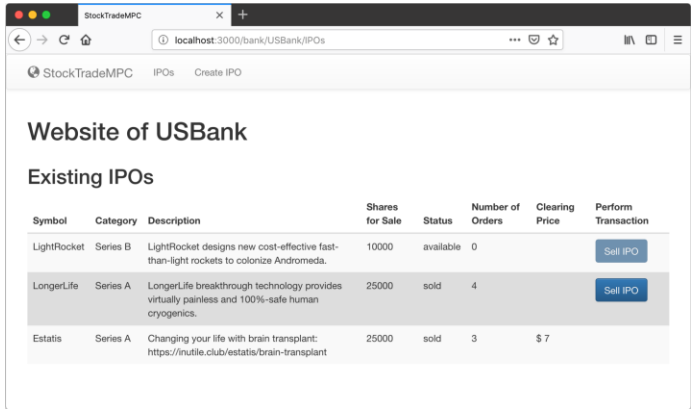
Demo: MPC based IPO on Blockchain



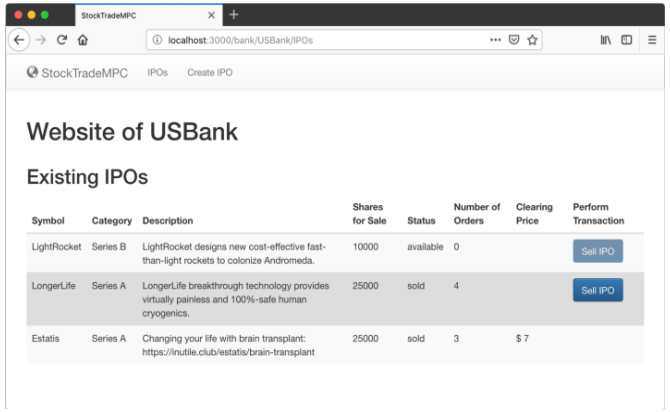
1. Bank create IPO



2. Buyers place orderers



3. Banks sell IPO



4. Bank list the closed IPO