Hyperledger and Fabric v1

Arnaud Le Hors
Senior Technical Staff Member Web & Blockchain Open Technologies, IBM
Member of the Hyperledger Technical Steering Committee
Contributor to Hyperledger Fabric
May 23rd, 2017
Shared Ledger Database

Blockchain allows different parties to securely interact with the same universal source of truth

**Finance**
Streamlined settlement, improved liquidity, increased transparency and new products/markets

**Healthcare**
Unite disparate processes, increase data flow and liquidity, reduce costs and improve patient experience and outcomes

**Supply Chain**
Track parts and service provenance, ensure authenticity of goods, block counterfeits, reduce conflicts
A World of Many Chains

There will not be only one blockchain, or a chain-of-all-chains.

There will be many public chains and millions of private chains, potentially each with a different consensus mechanism, preferred smart contract language/mecanism, and other characteristics.

The more common code underlying these chains, the better for everyone.

This is still early days – perhaps like 1994 and the Web?
Open source collaborative effort to advance cross-industry blockchain technologies

Hosted by The Linux Foundation, fastest-growing project in LF history

Global collaboration spanning finance, banking, IoT, supply chains, manufacturing and technology
Together with the global technology community, The Linux Foundation® is solving the world’s hardest problems through open source and **creating the largest shared technology investment in history**.

With 16 years experience providing **governance structure, IT infrastructure and ecosystem development**, The Linux Foundation is the umbrella organization for **more than 50 open source projects** accelerating open technology development and commercial adoption.

Some of the game-changing initiatives hosted by The Linux Foundation include:
300% Growth in year one!
Hyperledger’s Modular Umbrella Approach

**Infrastructure**
Technical, Legal, Marketing, Organizational

Ecosystems that accelerate open development and commercial adoption

**Frameworks**
Meaningfully differentiated approaches to business blockchain frameworks developed by a growing community of communities from the entire industry

**Modules**
Typically built for one framework, and through common license and community of communities approach, ported to other frameworks
Community Working Groups

Working Groups are open to the public

Architecture Working Group

Requirements Working Group

Identity Working Group

Whitepaper Working Group

Blockchain Protocol Working Group

Technical Working Group, China (TWG - China)

+ New Working Group on Performance and Scalability starting!
Community and Ecosystem Engagement

Regular participation and Hyperledger exhibits at cross-industry events.

Active engagement with technology and finance journalists and analysts to continue educating the market on Hyperledger. hyperledger.org/news

Regular online and face-to-face hackfests, hackathons, and meetups. Join our mailing lists to learn about these and other technical activities. hyperledger.org/community
Hyperledger Fabric

- Graduated to « Active » status
- Stable release is on branch v0.6
- Focus now shifted to 1.0 on master branch
  - 1.0.0-alpha released on March 17th
  - 1.0.0-alpha2 released on May 19th
Hyperledger Fabric v1.0 Architecture

Based on Source: https://jira.hyperledger.org/browse/FAB-37
Key characteristics of Hyperledger Fabric v1.0

- Better reflect business processes by specifying who endorses transactions
- Support broader regulatory requirements for privacy and confidentiality
- Scale the number of participants and transaction throughput
- Eliminate non-deterministic transactions
- Support rich data queries of the ledger
- Dynamically upgrade fabric and chaincode
- Support for multiple credential and cryptographic services for identity
- Support for "bring your own identity"
Ordering Service

The ordering service packages transactions into blocks to be delivered to peers. Communication with the service is via channels.

Different configuration options for the ordering service include:

- **SOLO**
  - Single node for development

- **Kafka**: Crash fault tolerant consensus
  - 3:n nodes minimum
  - Odd number of nodes recommended

- **SBFT**: Byzantine fault tolerant consensus
  - 4:n nodes minimum
Channels

Separate channels isolate transactions on different ledgers

- Chaincode is installed on peers that need to access the worldstate
- Chaincode is instantiated on specific channels for specific peers
- Ledgers exist in the scope of a channel
  - Ledgers can be shared across an entire network of peers
  - Ledgers can be included only on a specific set of participants
- Peers can participate in multiple channels
- Concurrent execution for performance and scalability
Single Channel Network

- Similar to 0.6 PBFT model
- All peers connect to the same system channel (blue).
- All peers have the same chaincode and maintain the same ledger
- Endorsement by peers $E_0$, $E_1$, $E_2$, and $E_3$
Multi Channel Network

- Peers $E_0$ and $E_3$ connect to the red channel for chaincodes $Y$ and $Z$
- Peers $E_1$ and $E_2$ connect to the blue channel for chaincodes $A$ and $B$
Installing and instantiating chaincode

Operator installs then instantiates

Operator installs smart contracts with endorsement policies to appropriate peers: E₀, E₁, E₂, P₃, and not P₄

Operator instantiates smart contract on given channel. One-time initialization

Policy subsequently available to all peers on channel, e.g. including P₄

Key:

<table>
<thead>
<tr>
<th>Endorser</th>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Committing Peer</td>
<td>Application</td>
</tr>
<tr>
<td>Ordering Node</td>
<td></td>
</tr>
<tr>
<td>Smart Contract (Chaincode)</td>
<td>Endorsement Policy</td>
</tr>
</tbody>
</table>
Endorsement Policies

Describe the conditions by which a transaction can be endorsed. A transaction can only be considered valid if it has been endorsed according to its policy.

- Each chaincode is associated with an Endorsement Policy
- Default implementation: Simple declarative language for the policy
- ESCC (Endorsement System ChainCode) signs the proposal response on the endorsing peer
- VSCC (Validation System ChainCode) validates the endorsements
Endorsement Policy Examples

Examples of policies:

• Request 1 signature from all three principals
  –  \text{AND('Org1.member', 'Org2.member', 'Org3.member')} \\

• Request 1 signature from either one of the two principals
  –  \text{OR('Org1.member', 'Org2.member')} \\

• Request either one signature from a member of the Org1 MSP or (1 signature from a member of the Org2 MSP and 1 signature from a member of the Org3 MSP)
  –  \text{OR('Org1.member', AND('Org2.member', 'Org3.member'))}
Application Developer’s Focus: client + chaincode

1. Client Application in using Hyperledger Fabric Client (HFC) SDK

2. Smart Contract implemented using chaincode – managing the World state

Client Application

SDK

2nd Smart Contract

Ledger

World state

Blockchain

 record

Ledger 'get', 'put', 'delete' recorded

emits

accesses

Blockchain

emit event

emit event

emit event
Hyperledger Fabric Roadmap

**Hack Fest docker images**
- 60 participants tested
- Basic v1 architecture in place
- Add / Remove Peers
- Channels
- Node SDK
- Go Chaincode
- Ordering Solo
- Fabric CA

**V1 Alpha ***
- Docker images
- Tooling to bootstrap network
- Fabric CA or bring your own
- Java and Node SDKs
- Ordering Services - Solo and Kafka
- Endorsement policy
- Level DB and Couch DB
- Block dissemination across peers via Gossip

**V1 GA ***
- Hardening, usability, serviceability, load, operability and stress test
- Java Chaincode
- Chaincode ACL
- Chaincode packaging & LCI
- Pluggable crypto
- HSM support
- Consumability of configuration
- Next gen bootstrap tool (config update)
- Config transaction lifecycle
- Eventing security
- Cross Channel Query
- Peer management APIs
- Documentation

**V Next ***
- SBFT
- Archive and pruning
- System Chaincode extensions
- Side DB for private data
- Application crypto library
- Dynamic service discovery
- REST wrapper
- Python SDK
- Identity Mixer (Stretch)
- Tcerts

* Dates for Alpha, Beta, and GA are determined by Hyperledger community and are currently proposals.

**2016 / 17 December**

**Connect-a-thon**
- 11 companies in Australia, Hungary, UK, US East Coast, US West Coast, Canada dynamically added peers and traded assets

**Connect-a-cloud**
- Dynamically connecting OEM hosted cloud environments to trade assets

Proposed Alpha detailed content: https://wiki.hyperledger.org/projects/proposedv1alphacontent
Getting started with Hyperledger Fabric

- Starter kit:
  - Start a simple network with 2 organizations running 2 peers
  - Docker images
  - Uses predefined enrollment certificates and « Solo » Ordering Service
- Start in devmode (minimal set up), then move to network (several peers), and security (membersrvvc)
- Chaincode: init, invoke, *query (0.6 only)*
- Several examples to start from (marbles, car lease demo)
More on using Hyperledger Fabric

- Application integration via:
  - APIs: gRPC, REST (0.6)
  - 4 SDKs: Node.js, Python, Java, Go

- CLI: launch + interact with peers and interact with membersrvcfabric-ca
  - Enroll / login
  - Peer start + stop
  - Channel create, join
  - Chaincode deploy, invoke, query

- Other images available: (fabric-couchdb, fabric-javaenv, etc.)

- Docker Images also available on Bluemix
Contributing to Hyperledger Fabric

• Contributor’s focus: Framework development
  Several areas to choose from: core, chaincode, consensus, ledger, SDKs…

• Development environment:
  Linux Foundation ID
  In Vagrant or natively on Linux or Mac
  Repository and Code review on Gerrit
  Project management on JIRA
Getting Help

• Documentation: http://hyperledger-fabric.readthedocs.io


• RocketChat: https://chat.hyperledger.org/channel/fabric

• Fabric mailing list: https://lists.hyperledger.org/pipermail/hyperledger-fabric/

• IBM Blockchain for developers: https://developer.ibm.com/courses/all-courses/blockchain-for-developers/

• Stackoverflow
Thank you!

www.ibm.com/blockchain
developer.ibm.com/blockchain
www.hyperledger.org