

**Introduction to Hyperledger Business Blockchain**

Consensus mechanisms



February 9, 2018

**About**

Hyperledger is an open source collaborative effort created to advance cross-industry blockchain technologies. It is a global collaboration including leaders in finance, banking, Internet of Things, supply chains, manufacturing and Technology. The Linux Foundation hosts Hyperledger under the foundation.

**Why Hyperledger?**

As with any early-stage, Highly-complex technology that demonstrates the ability to change the way we live our lives and conduct Business, Blockchain demands a cross-industry, Open source collaboration to advance the technology for all.

**Key Features**

* Increasing demand for permissioned blockchain
* Some Users/Use cases require validating or non-validating nodes to be controlled or sometimes just sponsored by select whitelisted organizations.
* Authority service on the network
* Need for private networks, as well as public networks
* Performance/scalability
* Channels for sharing confidential information
* Ordering Service delivers transactions consistently to peers in the network
* Endorsement policies for transactions
* Bring-your-own Membership Service Provider (MSP)
* Complete business model/Security features
* Really flexible and useful for the operation.

Fast Transaction For Example, Thousand operation per minute

**Hyperledger Frameworks/Projects**

The [Linux Foundation](https://www.sdxcentral.com/linux-foundation/)’s Hyperledger project, which is focused on [open source](https://www.sdxcentral.com/cloud/open-source/) blockchain technology, divides its work into [five sub projects](https://www.hyperledger.org/projects).

## **1.Fabric**

Fabric is Hyperledger’s most active project to date. The [Fabric 1.0 release](https://www.sdxcentral.com/articles/news/linux-foundation-blockchain-group-debuts-hyperledger-fabric-1-0/2017/07/) was issued in July. [IBM](https://www.sdxcentral.com/listings/ibm/) initiated the Fabric project. It’s intended as a foundation for developing blockchain distributed ledger applications with a modular architecture. It allows components, such as consensus and membership services, to be plug-and-play.

“Fabric is the granddaddy, if you will,” said Behlendorf. “Several companies are already selling products and services based on it.” The core of the platform is written in the Go programming language. A unique characteristic of Fabric is that its distributed ledger and smart contract platform allows for private channels. “If you have a large blockchain network and you want to share data with only certain parties, you can create a private channel with just those participants,” Behlendorf said. “It’s the most distinctive thing about Fabric right now.”

“159 engineers from 28 organizations contributed to project to advance open blockchain products and services”

## **2.Sawtooth**

The Sawtooth project originally came from [Intel](https://www.sdxcentral.com/listings/intel/). It includes a novel consensus algorithm called Proof of Elapsed Time. Consensus is a critical element of all blockchains. Generally, it is the technique by which new information is reviewed and confirmed before being accepted as the next entry in the ledger.

The Sawtooth consensus software targets large distributed validator populations with minimal resource consumption. “It may give us the ability to build very broad and flat networks of hundreds to thousands of nodes,” said Behlendorf. “It’s harder to do with traditional consensus mechanisms without having the CPU burden of cryptocurrencies.”

## **3.Indy**

The Indy project was originally the brainchild of the nonprofit group the [Sovrin Foundation](https://sovrin.org/). The idea is to provide digital identities for individuals and give them the power to share their identity with whom they chose. “Instead of being an entry in a giant data base, you have your data and deal programmatically with different organizations who want to check your identity,” said Behlendorf. “And companies don’t have to store so much personal data. They can store a pointer to the identity.”

Indy’s work looks especially timely, given the recent Experian hack. Behlendorf said Indy’s blockchain software is based on data minimization. When a company is done with your data, it throws it away. “It’s a toxic asset that could present a liability,” he said.

## **4.Burrow**

The Burrow project includes a permissioned, smart-contract interpreter built in part to the specification of the Ethereum Virtual Machine (EVM). The Ethereum platform is used both for cryptocurrency as well as for smart contracts. It’s written with the [Solidity programming language](http://solidity.readthedocs.io/en/latest/). Within the Burrow Project, the EVM is the interpreter for smart contracts (not related to cryptocurrency) that run across the Ethereum network.

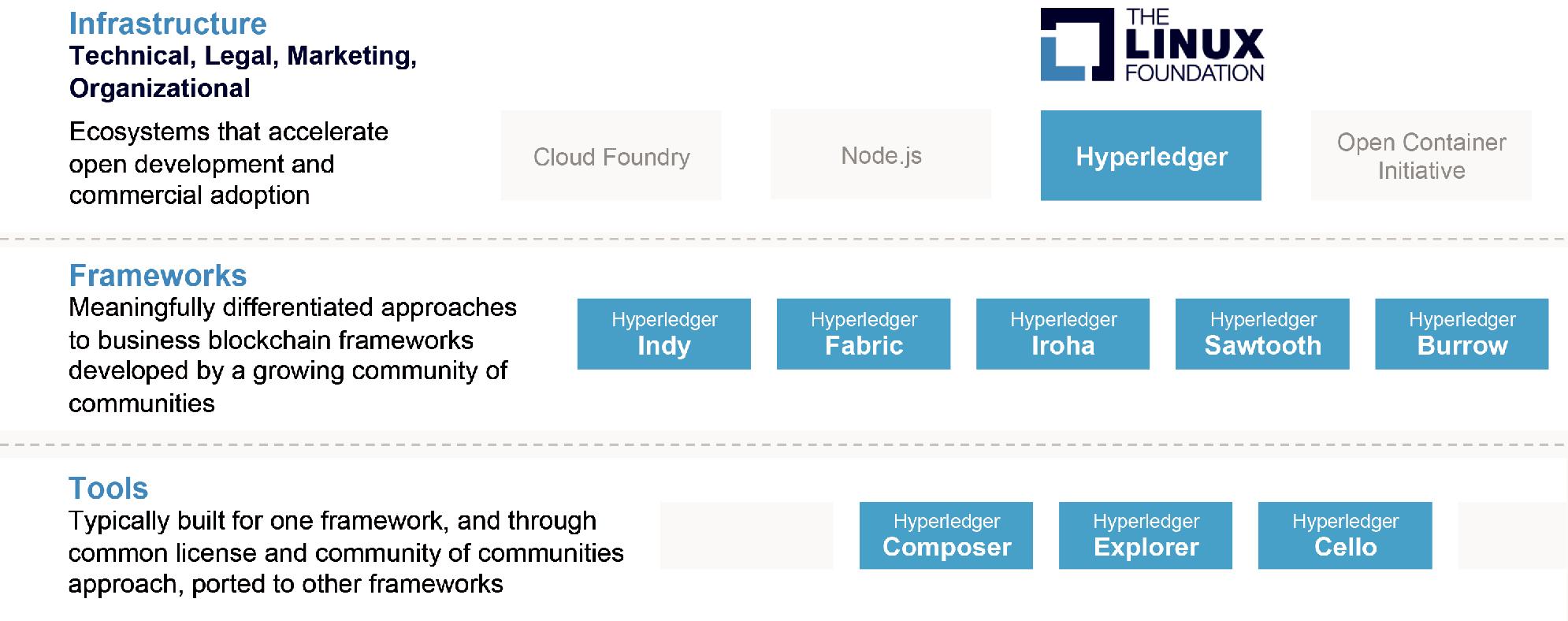
Many well-known companies belong to the [Enterprise Ethereum Alliance](https://entethalliance.org/members/), including JPMorgan, [Microsoft](https://www.sdxcentral.com/listings/microsoft/), [Accenture](https://www.sdxcentral.com/listings/accenture/), BP, and Cisco.

“It’s important to build a relationship with the Ethereum community,” said Behlendorf. “Burrow is the only Apache-licensed Ethereum VM implementation out there.”

## **5.Iroha**

Finally, the Iroha project is a bit of an outlier within Hyperledger. It originated with some developers in Japan who had built their own blockchain technology for a couple of mobile use cases. “It’s implemented in C++ which can be more high performance for small data and focused use cases,” said Behlendorf. “Iroha is still looking for its niche, but it’s a great development team.”

**Hyperledger Modular Design:**



**Architecture Overview**

All Hyperledger projects follow a design philosophy that includes a modular extensible approach, interoperability, an emphasis on highly secure solutions, a token-agnostic approach with no native cryptocurrency, and the development of a rich and easy-to use Application Programming Interface (API). The Hyperledger Architecture Working Group (WG) has distinguished the following business blockchain components:

* **Consensus Layer**

Responsible for generating an agreement on the order and confirming the correctness of the set of transactions that constitute a block.

* **Smart Contract Layer**

Responsible for processing transaction requests and determining if transactions are valid by executing business logic.

* **Communication Layer**

Responsible for peer-to-peer message transport between the nodes that participate in a shared ledger instance.

* **Data Store Abstraction**

Allows different data-stores to be used by other modules. • Crypto Abstraction - Allows different crypto algorithms or modules to be swapped out without affecting other modules.

* **Identity Services**

Enables the establishment of a root of trust during setup of a blockchain instance, the enrollment and registration of identities or system entities during network operation, and the management of changes like drops, adds, and revocations. Also, provides authentication and authorization.

* **Policy Services**

Responsible for policy management of various policies specified in the system, such as the endorsement policy, consensus policy, or group management policy. It interfaces and depends on other modules to enforce the various policies.

* **APIs**

Enables clients and applications to interface to blockchains.

* **Interoperation**

Supports the interoperation between different blockchain instances.

**Consensus**

Consensus is the process by which a network of nodes provides a guaranteed ordering of transactions and validates the block of transactions. Consensus must provide the following core functionality:

* Confirms the correctness of all transactions in a proposed block, according to endorsement and consensus policies.
* Agrees on order and correctness and hence on results of execution (implies agreement on global state).
* Interfaces and depends on smart-contract layer to verify correctness of an ordered set of transactions in a block.

**Comparison of Consensus Types**

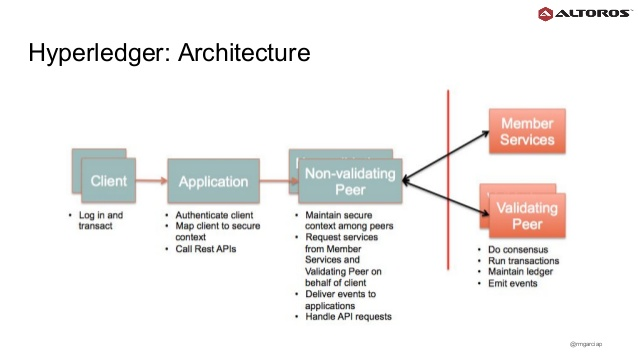
Consensus may be implemented in Two different ways:

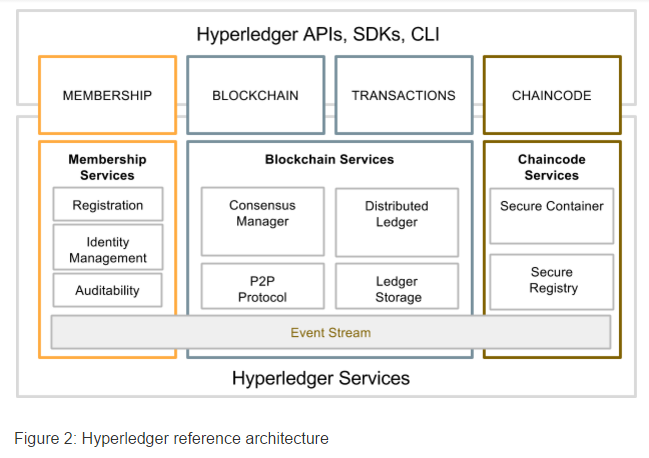
* **Lotterybased algorithms including Proof of Elapsed Time (PoET)** –

are advantageous in that they can scale to a large number of nodes since the winner of the lottery proposes a block and transmits it to the rest of the network for validation. On the other hand, these algorithms may lead to forking when two “winners” propose a block. Each fork must be resolved, which results in a longer time to finality.

* **Proof of Work (PoW) orthrough the use of voting-based methods including Redundant Byzantine Fault Tolerance(RBFT) and Paxos**.

The voting-based algorithms are advantageous in that they provide low-latency finality.When a majority of nodes validates a transaction or block, consensus exists and finality occurs. Because voting-based algorithms typically require nodes to transfer messages to each of the other nodes on the network, the more nodes that exist on the network,the more time it takes to reach consensus. This results in a trade-off between scalability and speed.

****



References

* <http://hyperledger-fabric.readthedocs.io/en/release/capabilities.html>
* **Sample Hyperledger fabric:**

<http://hyperledger-fabric.readthedocs.io/en/release/samples.html>

**Github file huperledger fabric** <https://github.com/hyperledger/fabric/blob/release/docs/source/samples.rst>

* <https://www.hyperledger.org/projects>
* <https://www.ibm.com/blockchain/hyperledger.html>
* <https://www.sdxcentral.com/articles/news/whats-the-difference-between-the-5-hyperledger-blockchain-projects/2017/09/>
* **Introduction to Blockchain and the Hyperledger Project**

https://pt.slideshare.net/ManuelGarcia122/introduction-to-blockchain-and-the-hyperledger-project/30?smtNoRedir=1

* **A Blockchain Application in Hyperledger Fabric** <http://www.blogsaays.com/learn-chaincode-tutorial-ibm-bluemix/>
* **Refer the Pdf file for your reference:**

<https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf>