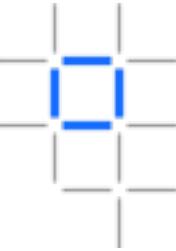




Creating Blockchain applications on Hyperledger Fabric and the IBM Cloud

Lennart Frantzell, IBM Developer Advocate

August 2018



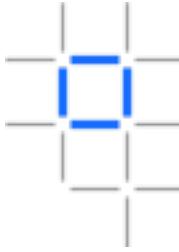
CALL FOR CODE®

Answer the Call for Code by building global solutions for disaster preparedness.

The most impactful project will be implemented with the help of IBM, The Linux Foundation, UN Human Rights, and American Red Cross.

Get started

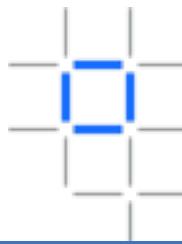
[Answer the call as an organization >](#)



IBM's three most important technologies:

- Watson and AI
- Blockchain with a team of 1500 people
- Internet of Things

IBM and the Blockchain Story



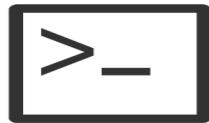
Open source at IBM

IBM's home for open source code, community, and culture.

At IBM, we do open source right. Discover new open source innovation, truly open communities, and continuing IBM leadership in open technologies.

Show me IBM's hot open source projects!

Show me all IBM projects on GitHub



The home of IBM generated innovation

IBMer's are building innovative new open source projects right now! Choose from cloud, analytics, blockchain, IoT — whatever your area of interest, we have the code for you.

Join the innovation



Join a thriving community

Dynamic communities are the cornerstone of open source development. IBM puts its time and resources into communities that support high-impact open source projects. Join us!

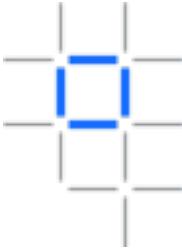
Find a community



The IBM open source way

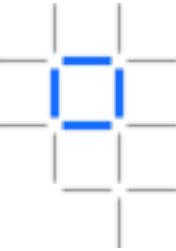
Best practices for training, tooling, automation, and more. We do open source with unmatched scope. Find out how we do it, and how you can too.

How do I do open source?



IBM and Blockchain, the story

1. Our blockchain team of 1,500 has experience with more than 400 engagements around the globe to develop blockchain applications, activate blockchain networks, and get the most value from this disruptive technology. For instance, we are working with:
2. The U.S. Food and Drug Administration to study the use of blockchain technology for secure exchange of healthcare data.
3. Sony and Sony Global Education to develop a new system, built on the IBM Blockchain platform, to manage students' learning data.
4. we.Trade, a group of nine of Europe's largest banks, provides a new trade finance platform designed to simplify domestic and cross-border trade for small and medium enterprises in Europe, while helping to increase overall trade transaction transparency.
5. Also, we recently announced that we are creating a joint venture with shipping giant Maersk that is meant to transform the global shipping industry.



Some key IBM'ers in Blockchain development



A photograph of Chris Ferris, a man with a beard, wearing a black t-shirt with the word "CODE" in white. He is standing on a stage with a large screen behind him displaying the word "CODE" and some text about a competition. To his right is a red banner with the word "TOWN".

Tweets 16.9K Following 1,844 Followers 4,641 Likes 44.5K Lists 12

[Following](#)

Chris Ferris



A screenshot of Christian Cachin's Twitter profile. It features a circular profile picture of a smiling man, a blue header bar, and a light gray sidebar.

Tweets 39 Following 111 Followers 361

[Tweets](#) [Tweets & replies](#) [Media](#)

christian cachin @cczurich Jul 11 Program is out -- Workshop on Blockchain Technology and Theory @podc_conference In two weeks in London! Looking forward to great selection of talks by excellent speakers: 2018.blockchain-workshop.net/program.shtml

christian cachin
@cczurich
cachin.com/cc/
Joined December 2013

Christian Cachin



A screenshot of Arnaud Le Hors's Twitter profile. It features a circular profile picture of a man with glasses, a blue header bar, and a light gray sidebar.

Tweets 517 Following 114 Followers 480 Likes 489

[Following](#)

[Tweets](#) [Tweets & replies](#) [Media](#)

Arnaud J Le Hors @jhors Senior Technical Staff Member, Web and Blockchain Open Technologies, IBM <standard_disclaimer/>

Dr. Angel Luis Diaz @angelluisdiaz Aug 9 Why IBM is offering \$200k to developers to create tech solutions for natural disaster relief tek.io/2LXe9Pg via @techrepublic

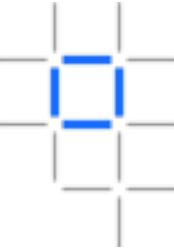
Who to follow • Refresh • View all

Sandro Hawke @sandha... Follow

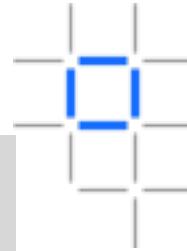
Jerry Cuomo @JerryCuomo

IBM Blockchain Arnaud Le Hors





Hyperledger Applications



TRADELENS: DIGITIZING THE GLOBAL SUPPLY CHAIN

TradeLens is an open and neutral industry platform underpinned by Blockchain technology, supported by major industry players.

•

Legitimize Diamonds and Reduce Fraud

What? Provenance.

- Track diamonds across supply chain from mine to retail

How?

- **Shared ledger for storing digital certification with supporting material**

Benefits

1. Protect against the occurrence of fraud, theft, trafficking and black markets
2. Assist in the identification and reduction of synthetic stones being labelled as authentic
3. Increase speed of transparency for cross border transactions for insurance companies, banks and claimants



Food Safety

What? Provenance

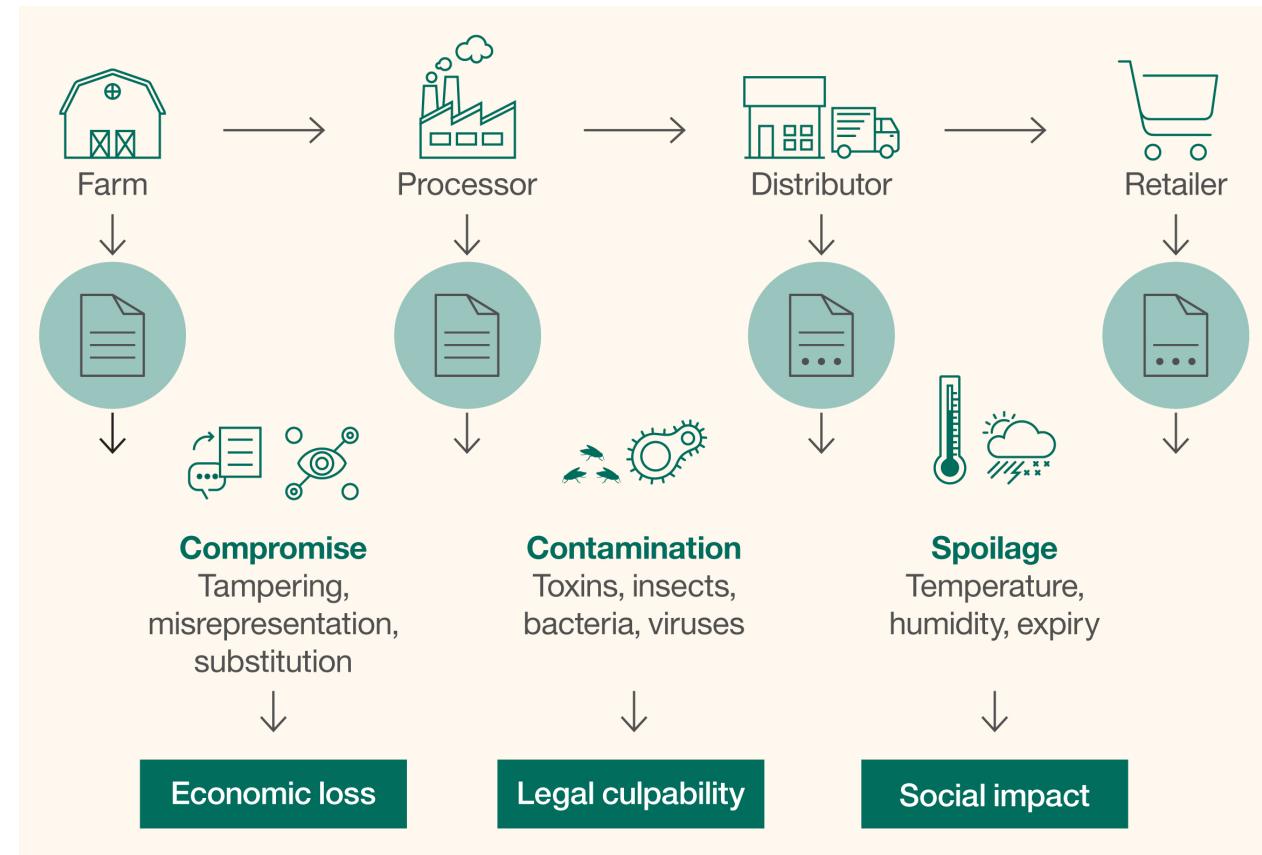
- Provide a trusted source of information and traceability to improve transparency and efficiency across the food network.

How?

- Shared ledger for storing digital compliance documentation, test results and audit certificates

Benefits

- Reduce impact of food recalls through instant access to end-to-end traceability data to verify history in the food network and supply chain.
- Help to address the 1 in 10 people sickened and 400,000 fatalities WW which occur every year from food-born illnesses.





Global Financing: Dispute Resolution

What?

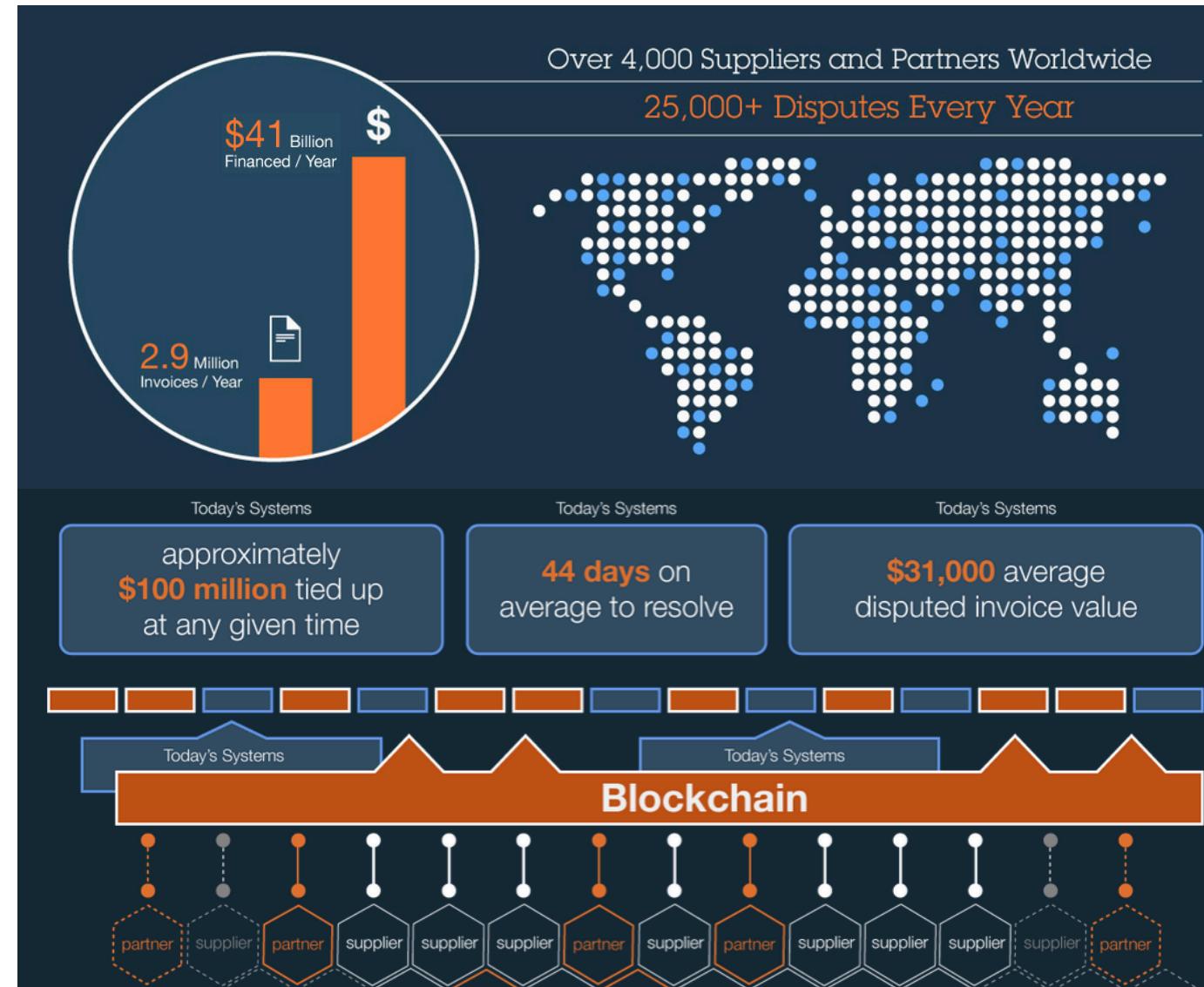
- IBM Global Finance provides a \$41bn channel financing per year. There are a number of disputes that take time to resolve and can lock up transactions costing time and money

How?

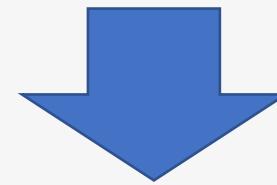
- **Blockchain provides visibility and provenance end-to-end across supply chain**

Benefits

1. Reduced dispute resolution time by 75%
2. Released working capital from \$100m
3. Combine IGF and Supplier info to further expand benefits further
4. In production since Sept 2016



Spectrum of Blockchains



Permissionless Public	Permissionless Private	Permissioned Public	Permissioned Private
Bitcoin, Ethereum	Public Polls	Land Titles, University Degrees	Medical Records Business Networks

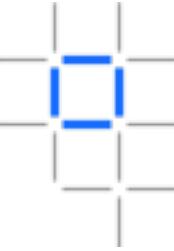
Permissioned vs. Permissionless: Who can write to a Blockchain, i.e. accessibility
Public vs Private: who can read from a Blockchain, i.e. visibility

<https://www.youtube.com/watch?v=-O3ouBdG3Xg>

Permissionless vs Permissioned Blockchains

The technology has recently been divided into two main camps:

- **Permissioned (private) blockchains**, which primarily focus on bringing blockchain-based solutions to enterprises, and
- **Permissionless (public) blockchains**, most notably Bitcoin, which aim to bring a fully decentralized cryptocurrency and distributed applications to the masses.



Hyperledger

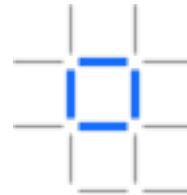
Introducing Hyperledger

**Open source
collaborative effort to
advance **cross-industry**
blockchain
technologies**

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

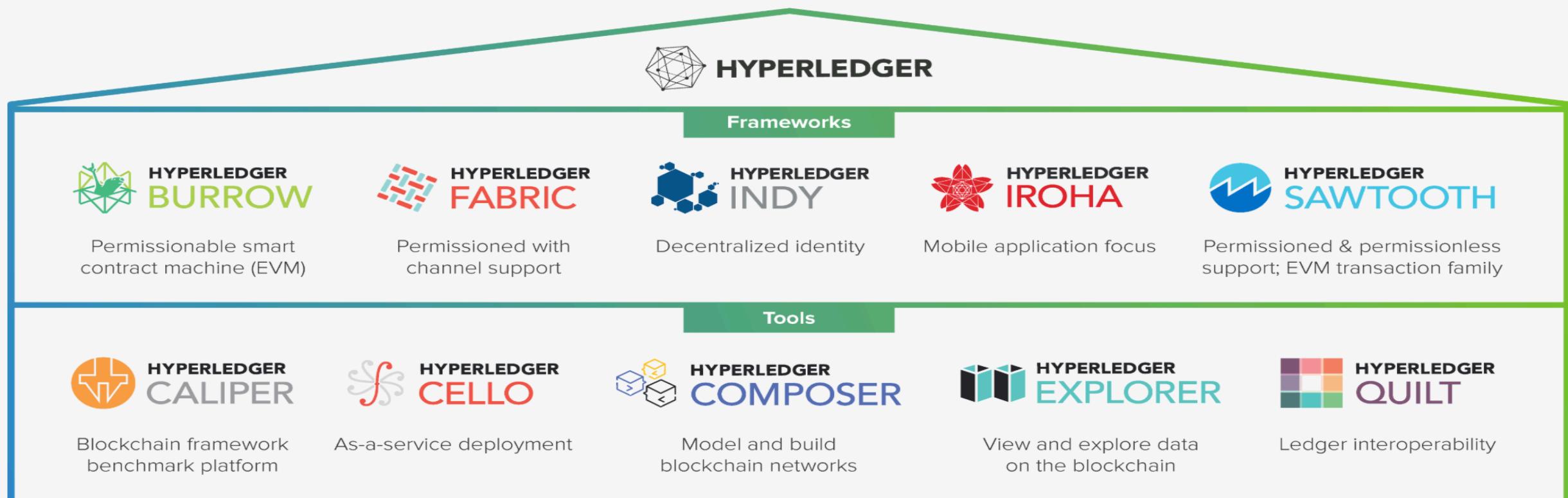
Global collaboration
spanning **finance,**
banking, IoT, supply
chain, healthcare,
manufacturing, govt,
technology and more.

Which Blockchain do we want to use?

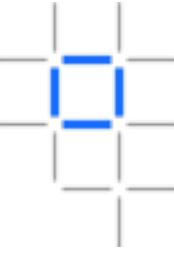


The Hyperledger Greenhouse

Business Blockchain Frameworks & Tools Hosted by Hyperledger



<https://www.hyperledger.org/>



Hyperledger Fabric

Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains

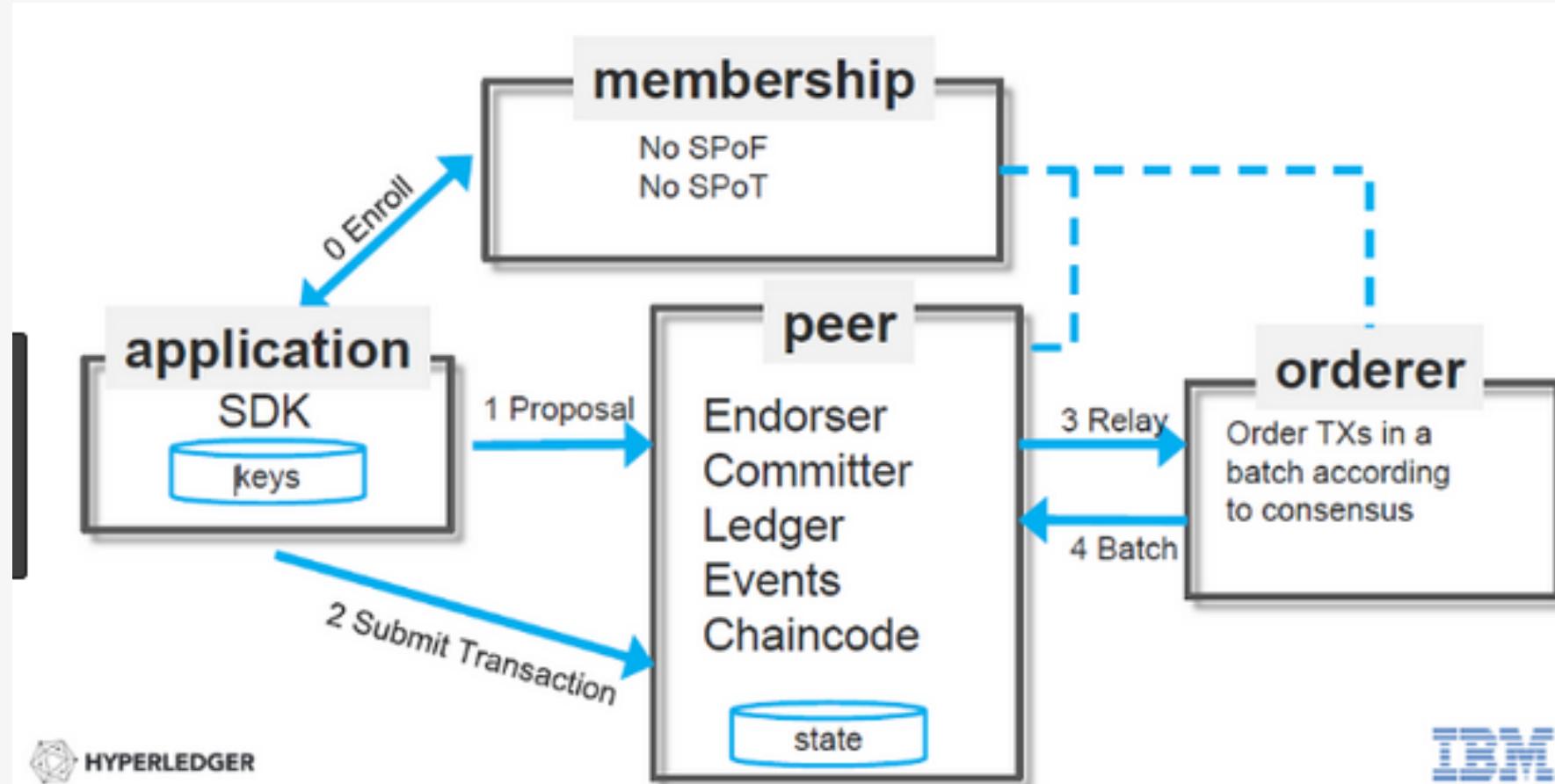
[Elli Androulaki](#), [Artem Barger](#), [Vita Bortnikov](#), [Christian Cachin](#), [Konstantinos Christidis](#), [Angelo De Caro](#), [David Enyeart](#), [Christopher Ferris](#), [Gennady Laventman](#), [Yakov Manevich](#), [Srinivasan Muralidharan](#), [Chet Murthy](#), [Binh Nguyen](#), [Manish Sethi](#), [Gari Singh](#), [Keith Smith](#), [Alessandro Sorniotti](#), [Chrysoula Stathakopoulou](#), [Marko Vukolić](#), [Sharon Weed Cocco](#), [Jason Yellick](#)

Fabric is a modular and extensible open-source system for deploying and operating permissioned blockchains and one of the Hyperledger projects hosted by the Linux Foundation (www.hyperledger.org).

Fabric is the first truly extensible blockchain system for running distributed applications. It supports modular consensus protocols, which allows the system to be tailored to particular use cases and trust models. Fabric is also the first blockchain system that runs distributed applications written in standard, general-purpose programming languages, without systemic dependency on a native cryptocurrency. This stands in sharp contrast to existing blockchain platforms that require "smart-contracts" to be written in domain-specific

This paper describes Fabric, its architecture, the rationale behind various design decisions, its most prominent implementation aspects, as well as its distributed application programming model. We further evaluate Fabric by implementing and benchmarking a Bitcoin-inspired digital currency. We show that Fabric achieves end-to-end throughput of more than 3500 transactions per second in certain popular deployment configurations, with sub-second latency, scaling well to over 100 peers.

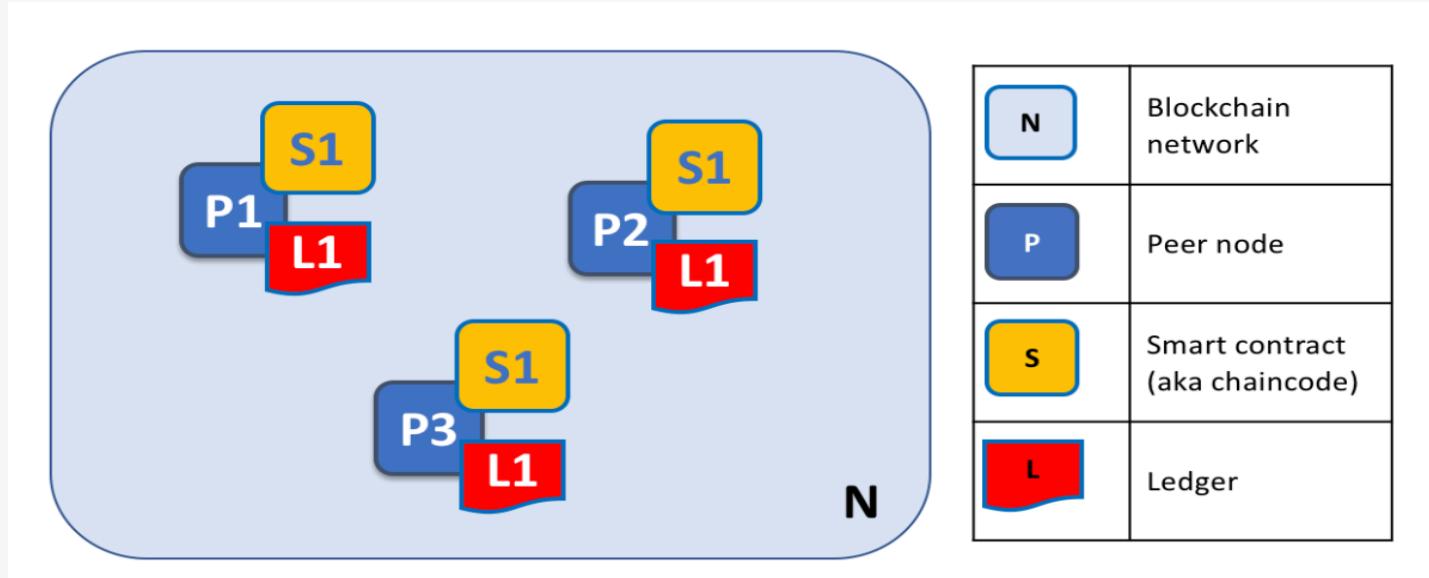
Hyperledger architecture



Hyperledger Fabric architecture

https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

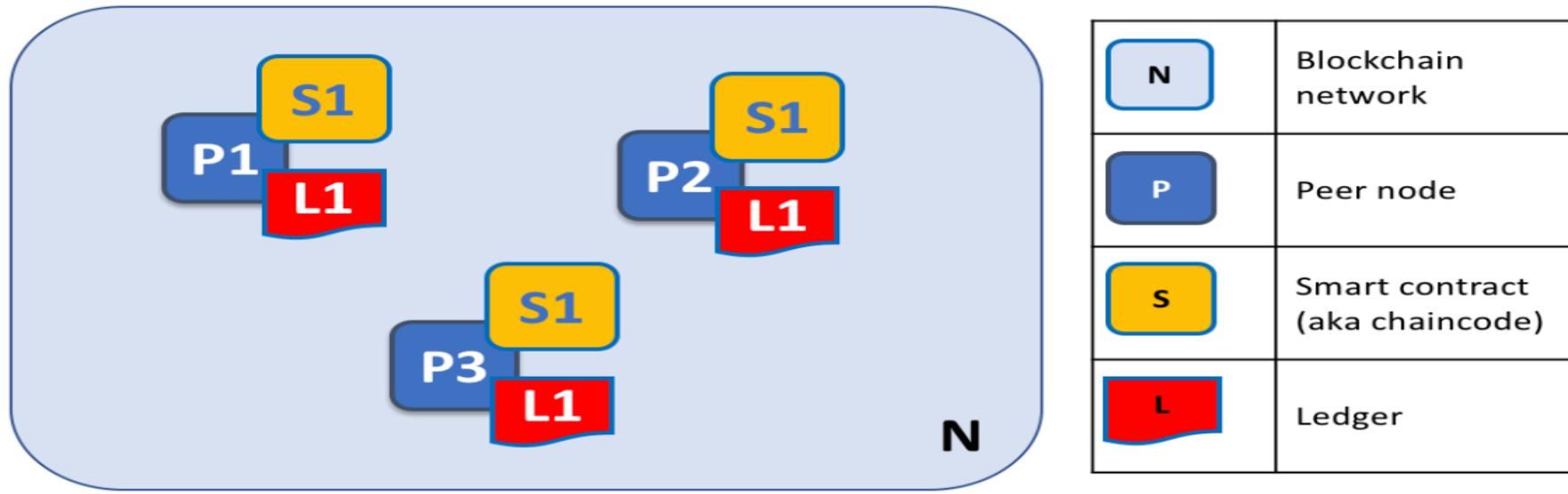
Hyperledger Fabric components



*A blockchain network is comprised of peer nodes, each of which can hold copies of ledgers and copies of smart contracts. In this example, the network **N** consists of peers **P1**, **P2** and **P3**, each of which maintain their own instance of the distributed ledger **L1**. **P1**, **P2** and **P3** use the same chaincode, **S1**, to access their copy of that distributed ledger.*

<https://hyperledger-fabric.readthedocs.io/en/release-1.2/peers/peers.html>

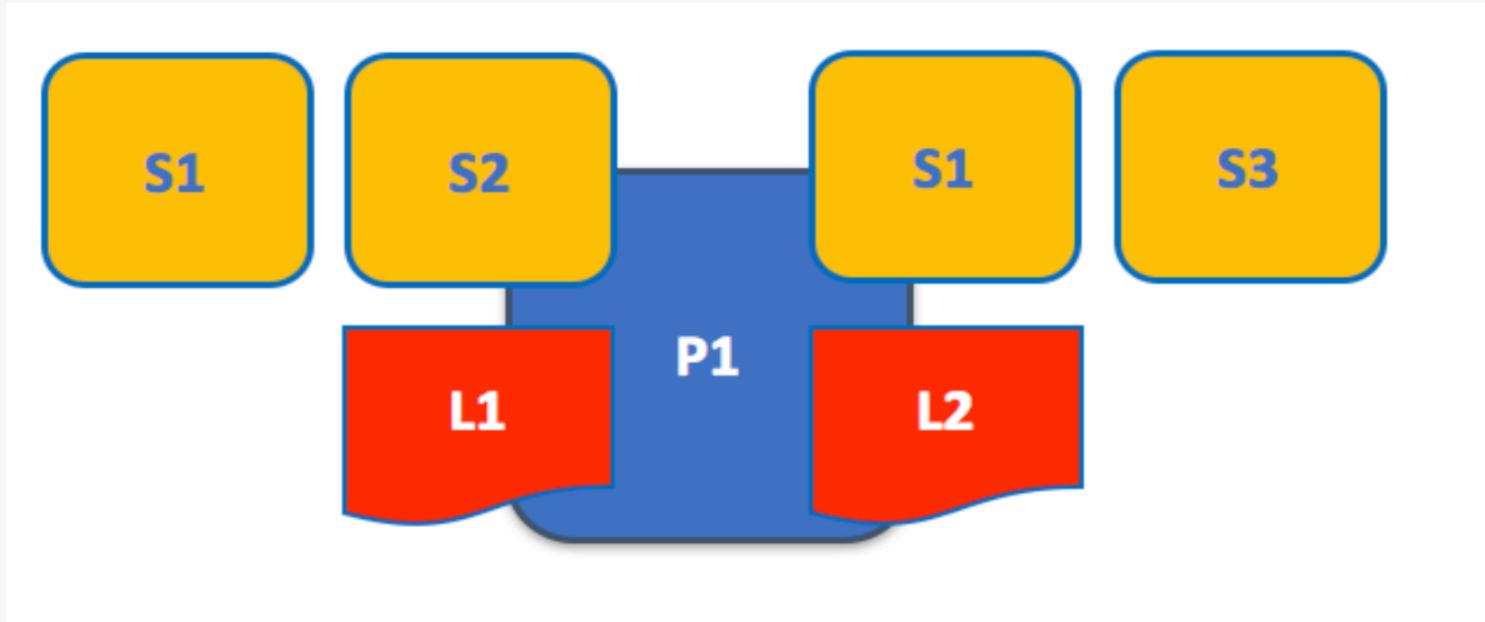
Hyperledger Fabric components



A peer hosting multiple ledgers. Peers host one or more ledgers, and each ledger has zero or more chaincodes that apply to them. In this example, we can see that the peer P1 hosts ledgers L1 and L2. Ledger L1 is accessed using chaincode S1. Ledger L2 on the other hand can be accessed using chaincodes S1 and S2.

<https://hyperledger-fabric.readthedocs.io/en/release-1.2/peers/peers.html>

Hyperledger Fabric components



An example of a peer hosting multiple chaincodes. Each ledger can have many chaincodes which access it. In this example, we can see that peer P1 hosts ledgers L1 and L2, where L1 is accessed by chaincodes S1 and S2, and L2 is accessed by S1 and S3. We can see that S1 can access both L1 and L2.

<https://hyperledger-fabric.readthedocs.io/en/release-1.2/peers/peers.html>

Much more information on Peers and Hyperledger message flow.

<https://hyperledger-fabric.readthedocs.io/en/release-1.2/peers/peers.html>

Consensus algorithms have different strengths and weaknesses



Solo /
No-ops

[Validators apply received transactions without consensus](#)

PROs: Very quick; suited to development

CONS: No consensus; can lead to divergent chains

Example usage: Hyperledger Fabric V1



PBFT-based

[Practical Byzantine Fault Tolerance implementations](#)

PROs: Reasonably efficient and tolerant against malicious peers

CONS: Validators are known and totally connected

Example usage: Hyperledger Fabric V0.6



Kafka /
Zookeeper

[Ordering service distributes blocks to peers](#)

PROs: Efficient and fault tolerant

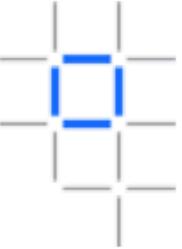
CONS: Does not guard against malicious activity

Example usage: Hyperledger Fabric V1

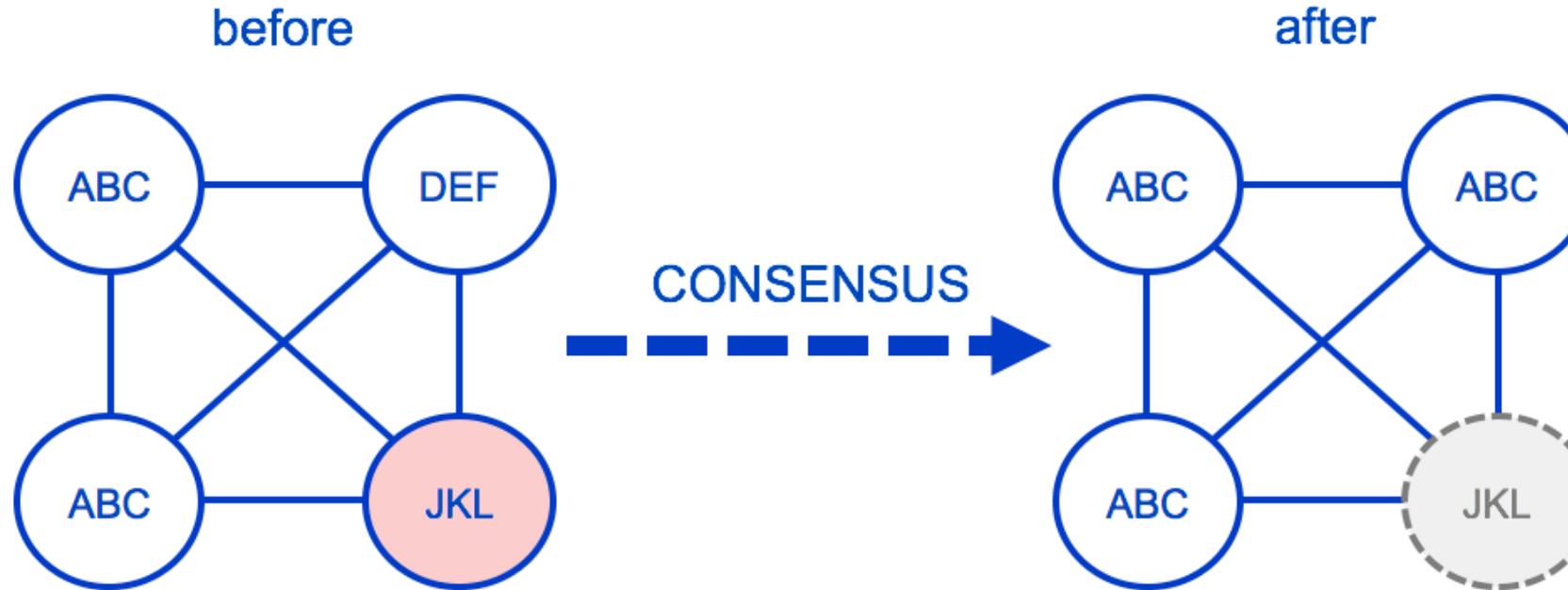


HYPERLEDGER

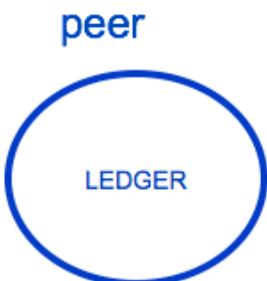
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

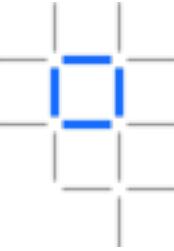


Consensus: The process of maintaining a consistent ledger



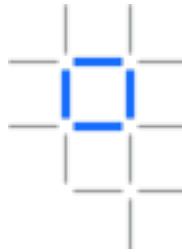
Keep all peers up-to-date
Fix any peers in error
Ignoring all malicious nodes





Hyperledger Composer and the Business Network

Which user roles do you need in a business network?

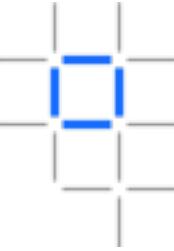


- **The consortium owner:** This is the organization with the vision for the network, its business benefits and how it will disrupt how business is done. Today, for practical purposes, it is best if there is only one consortium owner, since that entity will be responsible for making the network work.
- **Members:** They are part of the Hyperledger network. **They have a peer, which means they can store data, execute chain code (smart contracts), and participate in the consensus process** (if the consortium owner allows it). Members bring substance and validity to the network, and it is their data and intelligence that is shared. As the network matures, members can move up in status to become consortium owners.
- **General users:** These are users of the network and they are sponsored and enabled by the consortium owner and the members. **They can access and input data, but they do not own data.** As with members, as the network matures, general users can become members if they bring value to the network.

<https://www.ibm.com/blogs/systems/blockchain-who-should-be-in-your-consortium/>

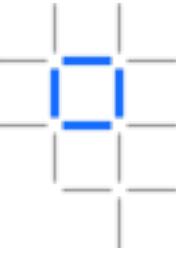
Getting started with Hyperledger Fabric

- **Develop Your First Network – Network administrator**
 - A simple network with 2 organizations running 2 peers, with one channel, a simple chaincode
 - Dockerhub images
 - Uses predefined enrollment certificates and « Solo » Ordering Service
- **Develop Your First Application – Application developer**
 - A simple Node.js application
 - **Start in** devmode (minimal set up), then move to network (several peers), and security (membersvc)
 - **Chaincode: init(), invoke()** – start developing in devmode
 - **Several examples to start from (marbles, car lease demo)**



Satoshi Nakamoto and his vision

Sources



1. <https://medium.com/swlh/hyperledger-chapter-6-hyperledger-fabric-components-technical-context-767985f605dd>

How did it all start?

October 2008 it all started with Satoshi Nakamoto and his paper **Peer-to-Peer Electronic Cash System** <https://bitcoin.org/bitcoin.pdf>

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution.

Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending.

We propose a solution to the double-spending problem using a peer-to-peer network.

The Blockchain Revolution

Harvard
Business
Review

INFORMATION & TECHNOLOGY

The Promise of Blockchain Is a World Without Middlemen

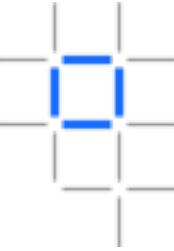
by Vinay Gupta

MARCH 06, 2017

Disintermediation:
Title Companies
Music Industry
Publishers
Agents
Banks
It's a peer-to-peer world

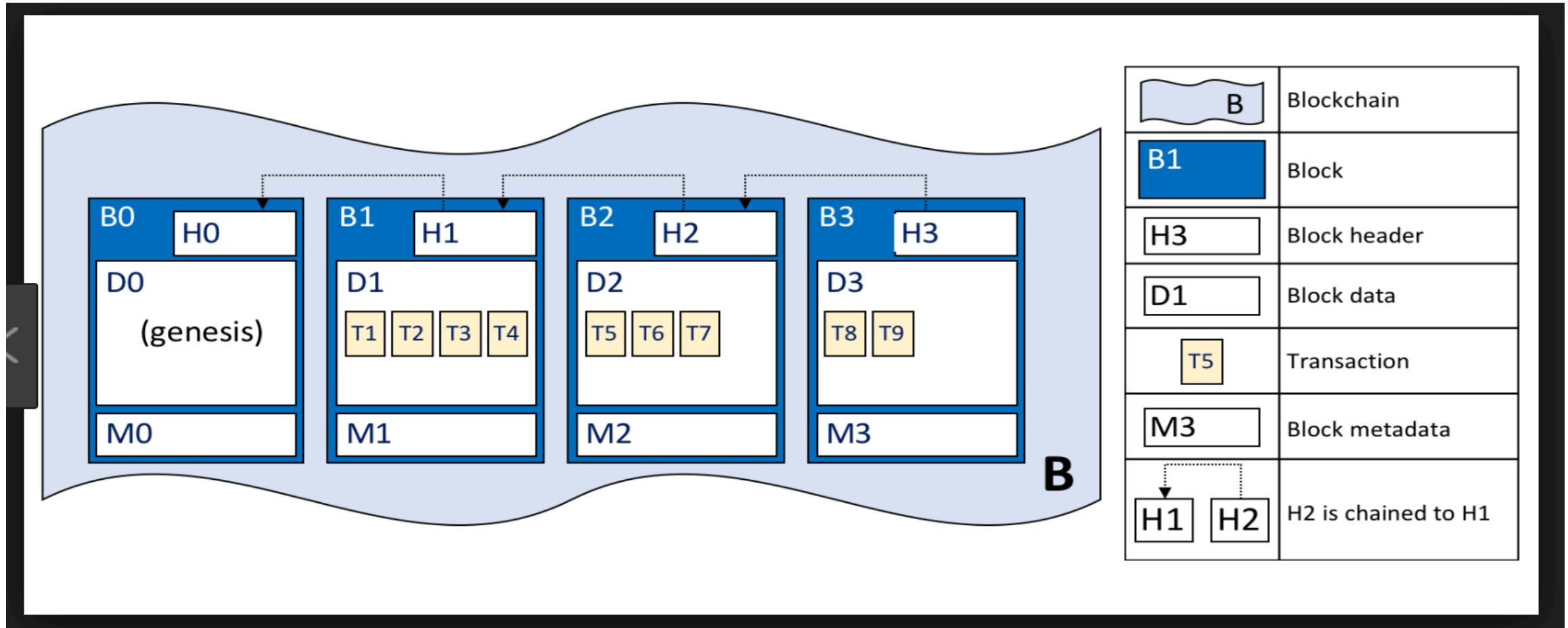
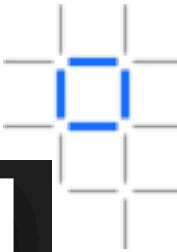


HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

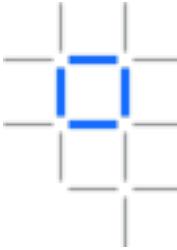


Blockchain Concepts

The chain of transaction blocks in the Blockchain

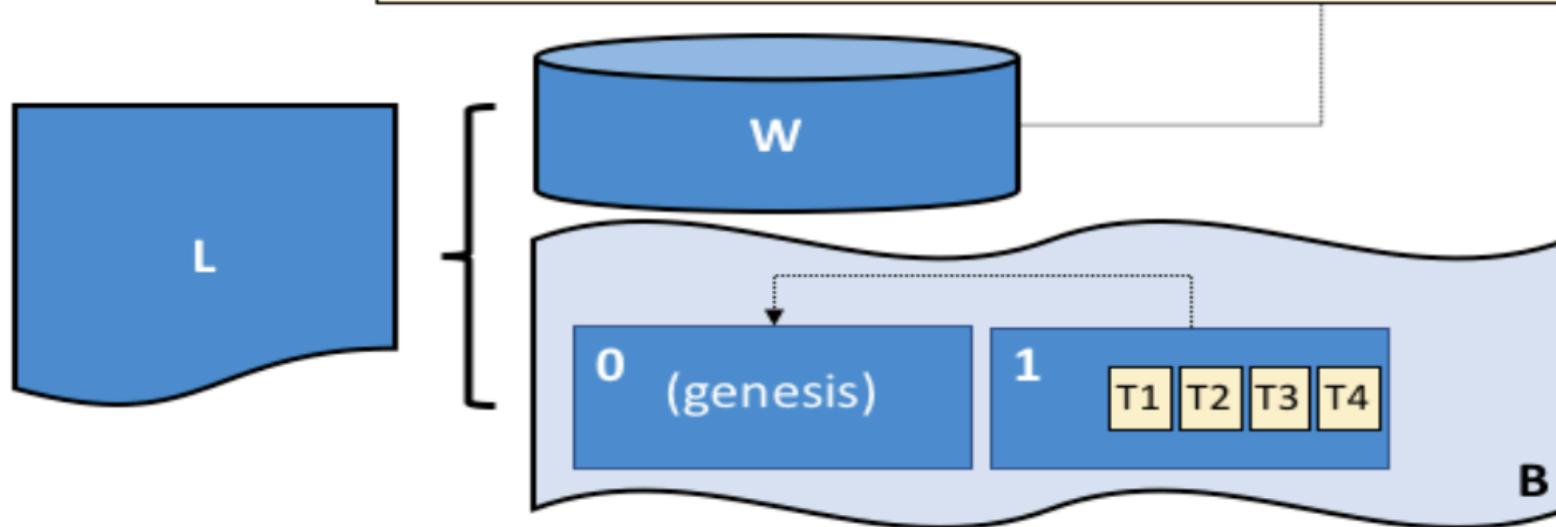


Each block is made up of a number of transactions, the result of chain code invocations

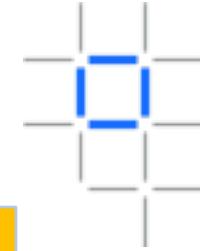


Transactions on the Blockchain from chain code invocations

key=CAR3, value={color: yellow, make: Volkswagen, model: Passat, owner: Max}	version=0
key=CAR2, value={color: green, make: Hyundai, model: Tucson, owner: Jin Soo}	version=0
key=CAR1, value={color: red, make: Ford, model: Mustang, owner: Brad}	version=0
key=CAR0, value={color: blue, make: Toyota, model: Prius, owner: Tomoko}	version=0



<https://hyperledger-fabric.readthedocs.io/en/release-1.2/ledger/ledger.html>

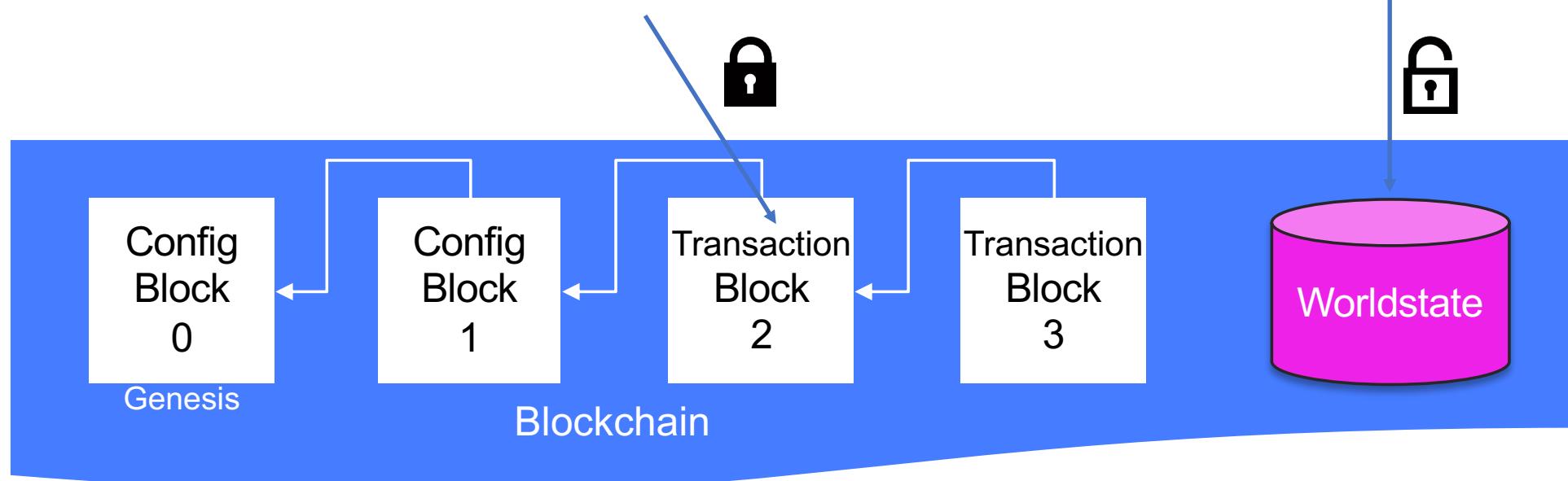


The “Right to Erasure” and blockchain

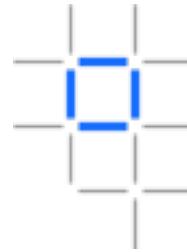
Enables an individual to request the deletion or removal of their personal data

Transactions
in
the
blockchain
are immutable

Data in the world
state is mutable



General Data Protection Regulation (GDPR) Solution – Store data off-chain



- **What**

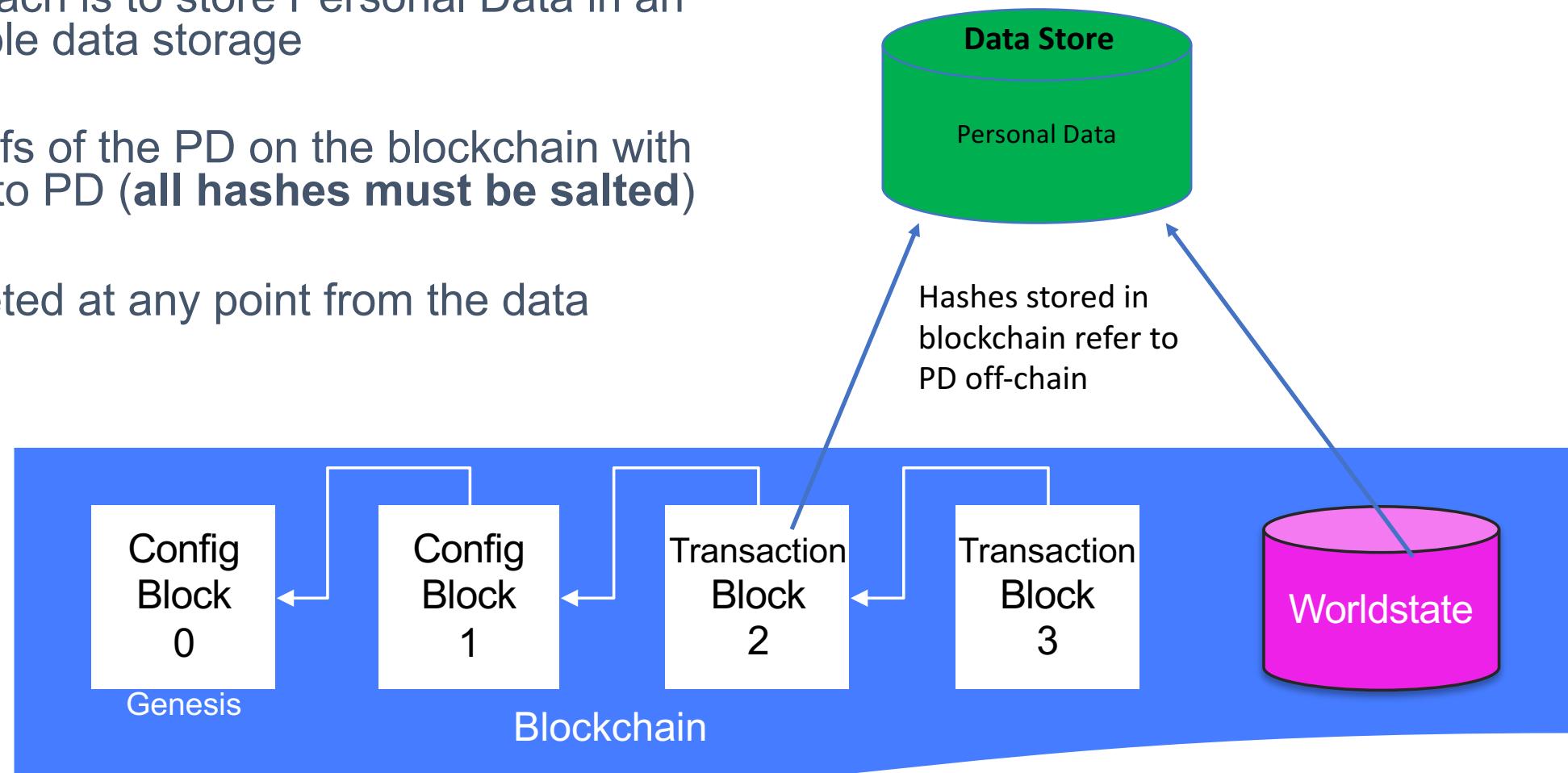
- The only approach is to store Personal Data in an off-chain mutable data storage

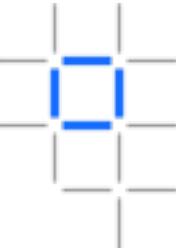
- **How**

- Store only proofs of the PD on the blockchain with hashes linking to PD (**all hashes must be salted**)

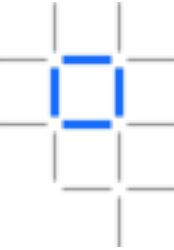
- **Why**

- PD can be deleted at any point from the data store(s)



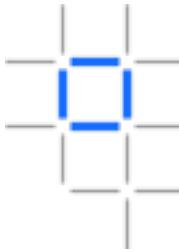


- <https://hyperledger-fabric.readthedocs.io/en/release-1.2/ledger/ledger.html>

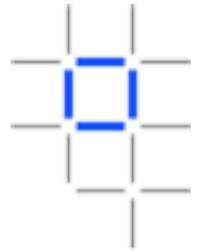


- The Internet made p2p transactions of information possible
- The Blockchain will make p2p transactions of value possible.

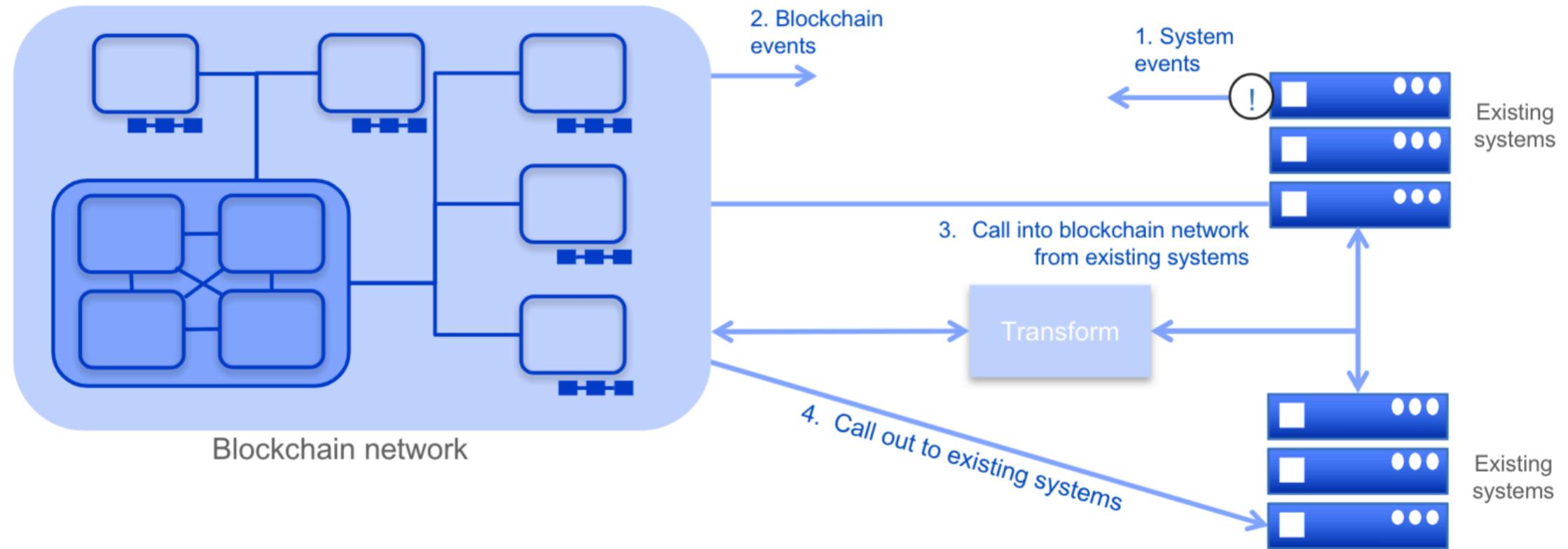
Off chain, the problem with the Systems of Record



- All data can not be stored on the Blockchain
 - Example X-ray images
 - Diamonds
 - Art works
 - Automobiles
 - Aircraft parts
 - Food



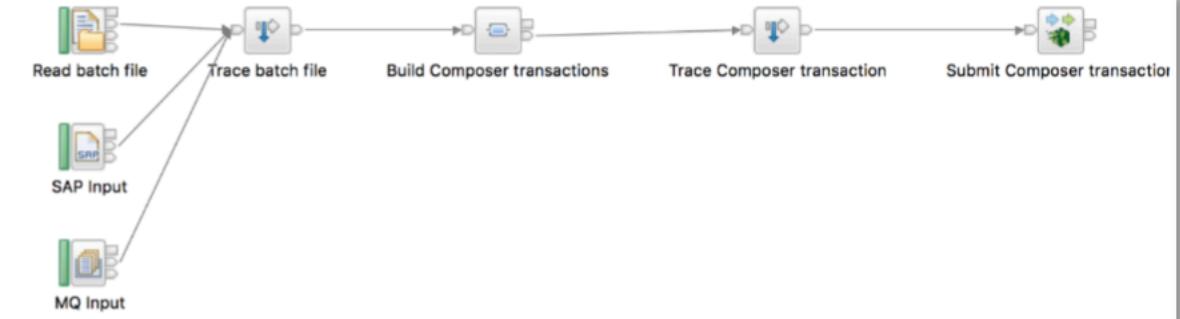
Integrating with existing systems - possibilities





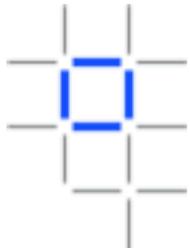
Integrating with existing systems – using middleware

- Blockchain is a network system of record
- Two-way exchange
 - Events from blockchain network create actions in existing systems
 - Cumulative actions in existing systems result in Blockchain interaction



- Transformation between blockchain and existing systems' formats
 - GBO, ASBO is most likely approach
 - Standard approach will be for gateway products to bridge these formats
 - Gateway connects to peer in blockchain network and existing systems
- Smart contracts can call out to existing systems
 - Query is most likely interaction for smart decisions
 - e.g. all payments made before asset transfer?
 - Warning: Take care over predictability... transaction must provide same outputs each time it executes





Non-determinism in blockchain

- Blockchain is a distributed processing system
 - Smart contracts are run multiple times and in multiple places
 - As we will see, smart contracts need to run deterministically in order for consensus to work
 - Particularly when updating the world state
- It's particularly difficult to achieve determinism with off-chain processing
 - Implement services that are guaranteed to be consistent for a given transaction, or
 - Detect duplicates for a transaction in the blockchain, middleware or external system

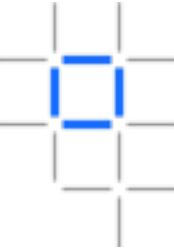
random()

getExchangeRate()

getDateTime()

getTemperature()

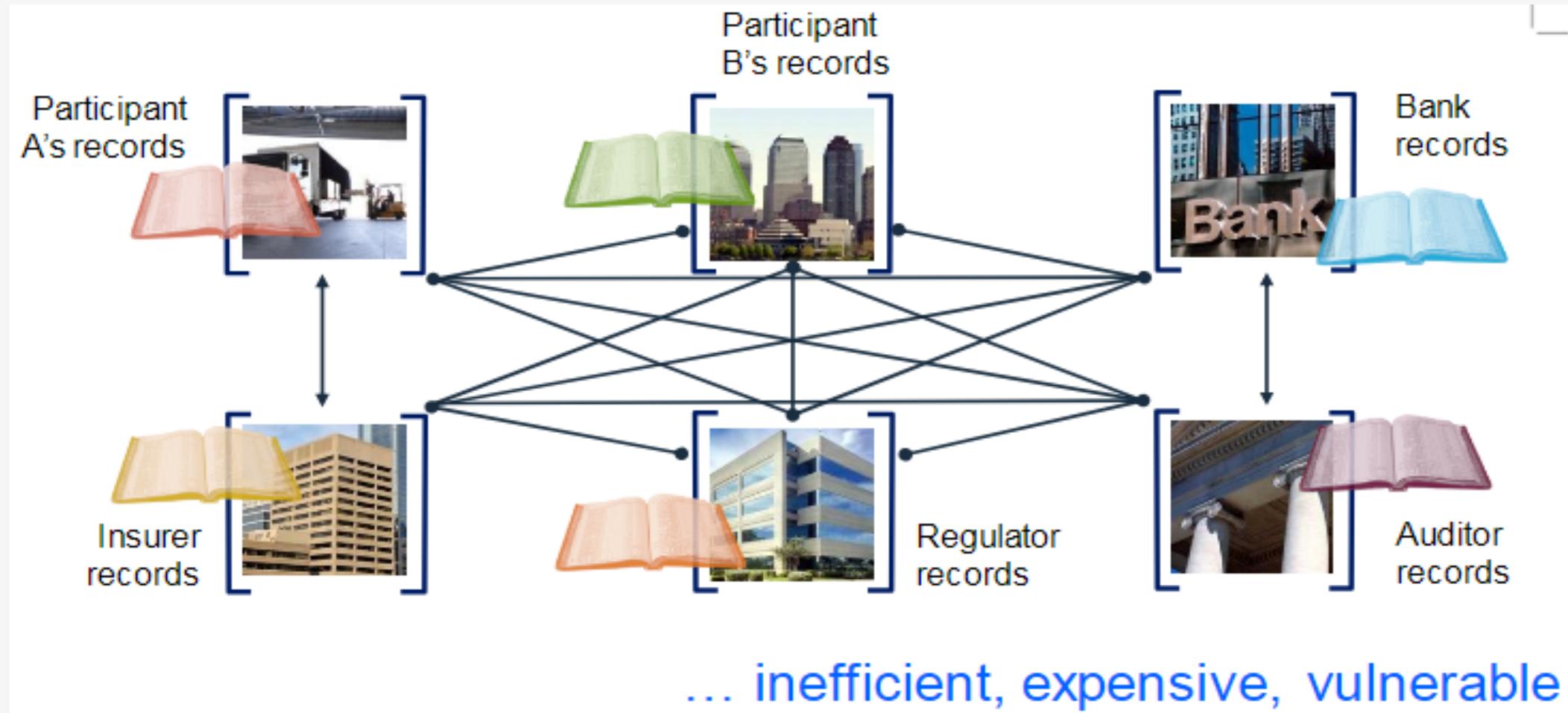
incrementValue
inExternalSystem(...)



Blockchain Types

A view from the Ledger

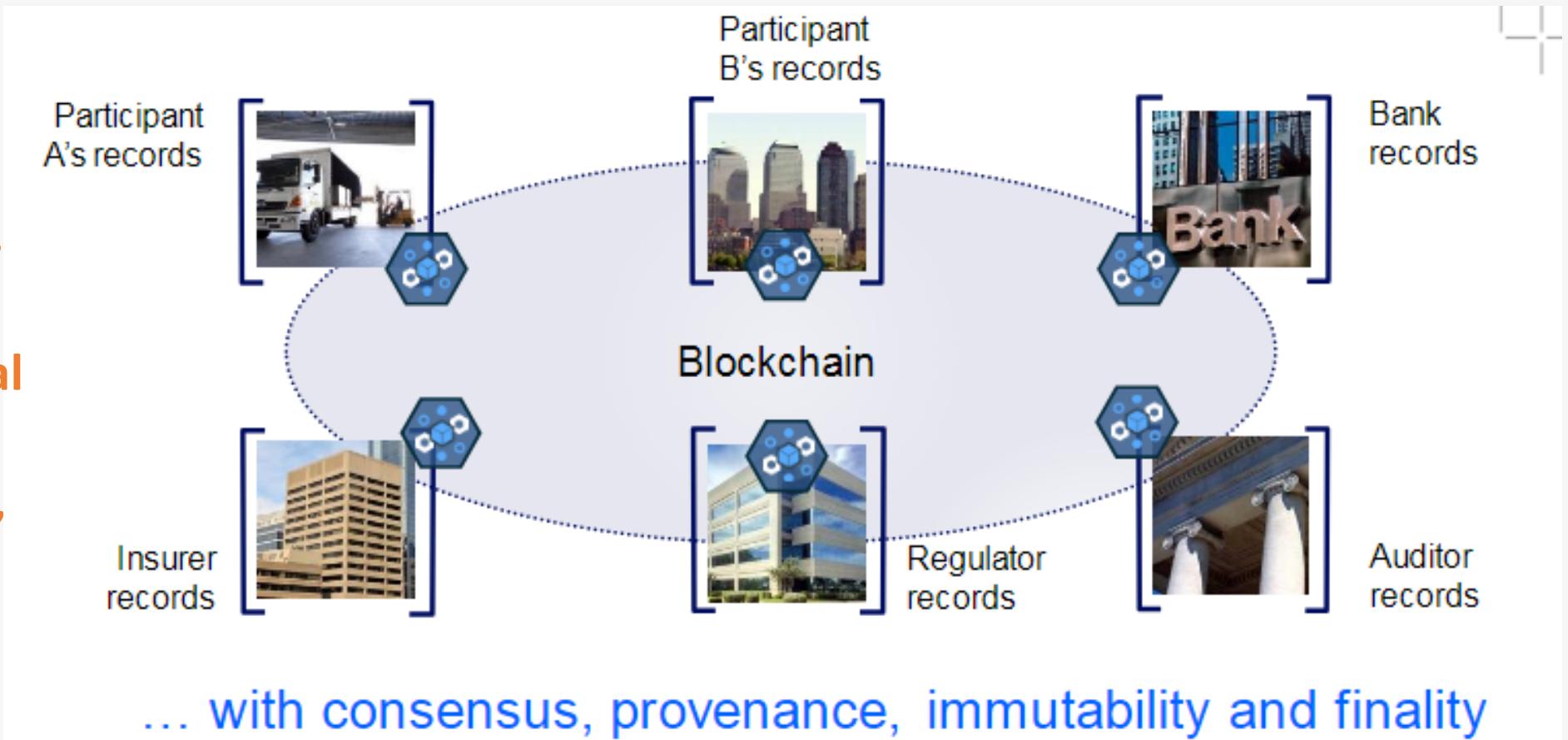
From the multiple Ledger Database



To the shared, replicated, permissioned ledger

No central, third-party mediator, such as a financial institution or clearinghouse, is involved

No reconciliation needed

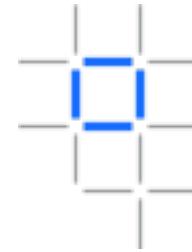


Facets of distributed, shared ledgers

Network nodes both generate their own data and verify data generated by others	Contain historic record of verified transactions that are easily auditable	Distributed Consensus eliminates costly and inefficient reconciliation processes
No central repository – each node stores identical copies of the ledger	Resilient due to network power and cryptographic integrity	Large economic disincentive for malicious actors

<https://composer-playground.mybluemix.net/>

Welcome to Hyperledger Composer Playground!

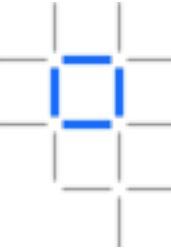


In this web sandbox, you can deploy, edit and test business network definitions. Have a play and learn what Hyperledger Composer Playground is all about.

Let's Blockchain!



Not sure where to start? View our Playground tutorial.



My Business Networks

Connection: Web Browser

The screenshot shows the Hyperledger Composer Playground interface. It features a central panel with a "Hello, Composer!" message and a "basic-sample-network" entry. A "Get Started" button is at the bottom. To the right is a dashed box containing a "Deploy a new business network" section.

Hello, Composer!

Get started with the basic-sample-network, or view our [Playground tutorial](#)

BUSINESS NETWORK

basic-sample-network

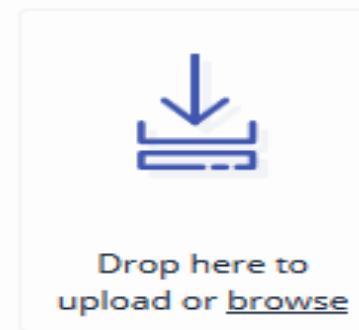
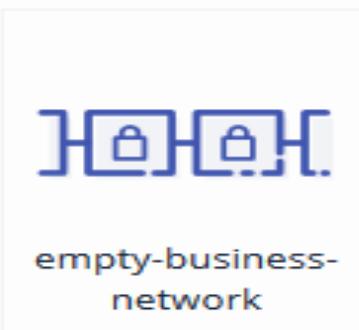
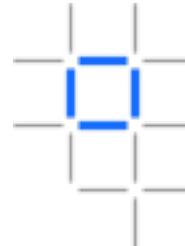
Get Started →



Deploy a new business network

Choose a Business Network Definition to start with:

Choose a sample to play with, start a new project, or import your previous work



Samples on npm



FILES

About

[README.md, package.json](#)

Model File

[models/sample.cto](#)

Script File

[lib/sample.js](#)

Access Control

[permissions.acl](#)[Add a file...](#) [Export](#)

UPDATE NETWORK

From: 0.2.6-20180530153450

To: 0.2.6-deploy.0 [Deploy changes](#)

About File README.md



Basic Sample Business Network

This is the "Hello World" of Hyperledger Composer samples, which demonstrates the core functionality of Hyperledger Composer by changing the value of an asset.

This business network defines:

Participant `SampleParticipant`

Asset `SampleAsset`

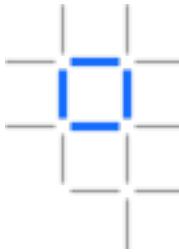
Transaction `SampleTransaction`

Event `SampleEvent`

SampleAssets are owned by a SampleParticipant, and the value property on a SampleAsset can be modified by submitting a SampleTransaction. The SampleTransaction emits a SampleEvent that notifies applications of the old and new values for each modified SampleAsset.

To test this Business Network Definition in the **Test** tab:

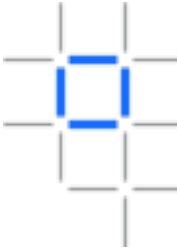
Hyperledger Composer Modeling Language



Hyperledger Composer includes an object-oriented modeling language that is used to define the domain model for a business network definition

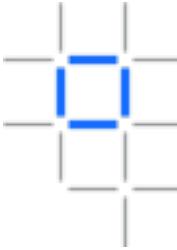
1. A single namespace. All resource declarations within the file are implicitly in this namespace.
2. A set of resource definitions, encompassing assets, transactions, participants, and events.
3. Optional import declarations that import resources from other namespaces.

https://hyperledger.github.io/composer/latest/reference/cto_language



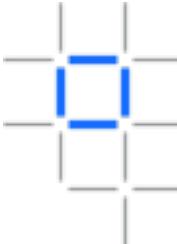
Model File models/sample.cto

```
13  /*
14
15  /**
16   * Sample business network definition.
17  */
18 namespace org.example.basic
19
20 asset SampleAsset identified by assetId {
21   o String assetId
22   --> SampleParticipant owner
23   o String value
24 }
25
26 participant SampleParticipant identified by participantId {
27   o String participantId
28   o String firstName
29   o String lastName
30 }
31
32 transaction SampleTransaction {
33   --> SampleAsset asset
34   o String newValue
35 }
36
37 event SampleEvent {
38   --> SampleAsset asset
39   o String oldValue
40   o String newValue
41 }
42
```

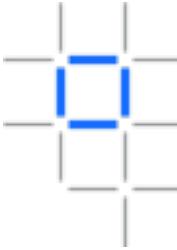


Script File

```
13  /**
14
15  /* global getAssetRegistry getFactory emit */
16
17  /**
18   * Sample transaction processor function.
19   * @param {org.example.basic.SampleTransaction} tx The sample transaction instance.
20   * @transaction
21  */
22  async function sampleTransaction(tx) { // eslint-disable-line no-unused-vars
23
24      // Save the old value of the asset.
25      const oldValue = tx.asset.value;
26
27      // Update the asset with the new value.
28      tx.asset.value = tx.newValue;
29
30      // Get the asset registry for the asset.
31      const assetRegistry = await getAssetRegistry('org.example.basic.SampleAsset');
32      // Update the asset in the asset registry.
33      await assetRegistry.update(tx.asset);
34
35      // Emit an event for the modified asset.
36      let event = getFactory().newEvent('org.example.basic', 'SampleEvent');
37      event.asset = tx.asset;
38      event.oldValue = oldValue;
39      event.newValue = tx.newValue;
40      emit(event);
41  }
42 }
```



```
* Sample access control list.  
*/  
rule EverybodyCanReadEverything {  
    description: "Allow all participants read access to all resources"  
    participant: "org.example.basic.SampleParticipant"  
    operation: READ  
    resource: "org.example.basic.*"  
    action: ALLOW  
}  
  
rule EverybodyCanSubmitTransactions {  
    description: "Allow all participants to submit transactions"  
    participant: "org.example.basic.SampleParticipant"  
    operation: CREATE  
    resource: "org.example.basic.SampleTransaction"  
    action: ALLOW  
}  
  
rule OwnerHasFullAccessToTheirAssets {  
    description: "Allow all participants full access to their assets"  
    participant(p): "org.example.basic.SampleParticipant"  
    operation: ALL  
    resource(r): "org.example.basic.SampleAsset"  
    condition: (r.owner.getIdentifier() === p.getIdentifier())  
    action: ALLOW  
}
```



```
rule SystemACL {
    description: "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Participant"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}

rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}

rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

FILES

About
README.md, package.json

Model File
models/sample.cto

Script File
lib/sample.js

Access Control
permissions.acl

Add a file... Export

UPDATE NETWORK

From: 0.2.6-20180530153450

To: 0.2.6-deploy.0

Deploy changes

About File README.md



Basic Sample Business Network

This is the "Hello World" of Hyperledger Composer samples, which demonstrates the core functionality of Hyperledger Composer by changing the value of an asset.

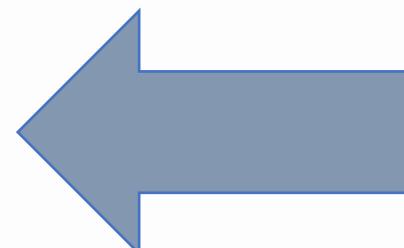
This business network defines:

Participant SampleParticipant

Asset SampleAsset

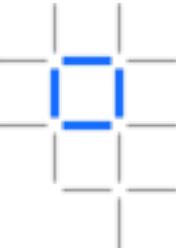
Transaction SampleTransaction

Event SampleEvent



SampleAssets are owned by a SampleParticipant, and the value property on a SampleAsset can be modified by submitting a SampleTransaction. The SampleTransaction emits a SampleEvent that notifies applications of the old and new values for each modified SampleAsset.

To test this Business Network Definition in the **Test** tab:



Create a `SampleParticipant` participant:

```
{  
  "$class": "org.example.basic.SampleParticipant",  
  "participantId": "Toby",  
  "firstName": "Tobias",  
  "lastName": "Hunter"  
}
```

Create a `SampleAsset` asset:

```
{  
  "$class": "org.example.basic.SampleAsset",  
  "assetId": "assetId:1",  
  "owner": "resource:org.example.basic.SampleParticipant#Toby",  
  "value": "original value"  
}
```

Submit a `SampleTransaction` transaction:

```
{  
  "$class": "org.example.basic.SampleTransaction",  
  "asset": "resource:org.example.basic.SampleAsset#assetId:1",  
  "newValue": "new value"  
}
```

After submitting this transaction, you should now see the transaction in the Transaction Registry and that a `SampleEvent` has been emitted. As a result, the value of the `assetId:1` should now be `new value` in the Asset Registry.

Participant registry for org.example.basic.SampleParticipant

[+ Create New Participant](#)

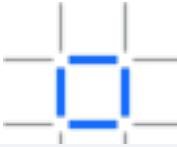
ID

Data

Toby

```
{  
    "$class": "org.example.basic.SampleParticipant",  
    "participantId": "Toby",  
    "firstName": "Tobias",  
    "lastName": "Hunter"  
}
```





Asset registry for org.example.basic.SampleAsset

+ Create New Asset

ID

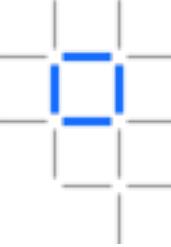
Data

assetId:1

```
{  
    "$class": "org.example.basic.SampleAsset",  
    "assetId": "assetId:1",  
    "owner": "resource:org.example.basic.SampleParticipant#Toby",  
    "value": "original value"  
}
```



Submit Transaction



Transaction Type

SampleTransaction

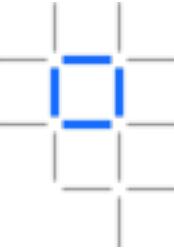


JSON Data Preview

```
1  {
2    "$class": "org.example.basic.SampleTransaction",
3    "asset": "resource:org.example.basic.SampleAsset#assetId:1",
4    "newValue": "new value"
5 }
```



Optional Properties



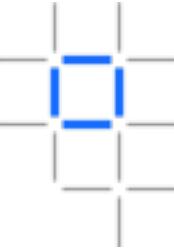
Historian Record

X

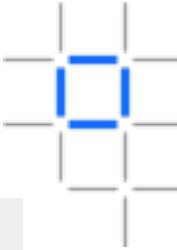
Transaction

Events (1)

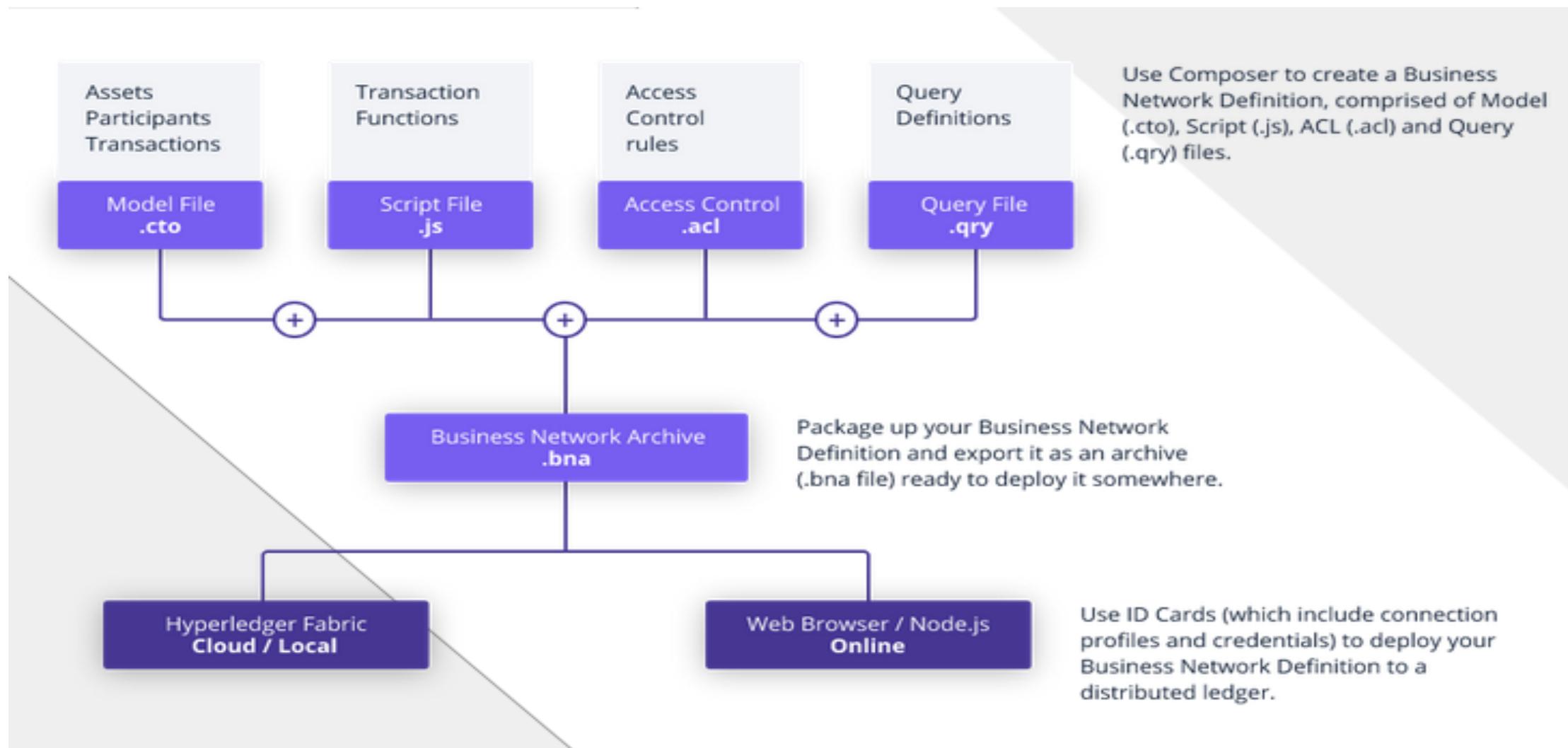
```
1  {
2    "$class": "org.example.basic.SampleTransaction",
3    "asset": "resource:org.example.basic.SampleAsset#assetId:1",
4    "newValue": "new value",
5    "transactionId": "a100b354-4ae3-47c5-a341-d255a724fdee",
6    "timestamp": "2018-06-23T19:26:21.501Z"
7 }
```



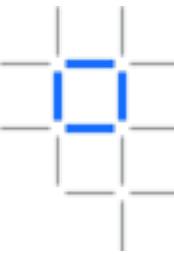
Deploying the BNA file to Hyperledger Fabric and the Cloud



Deploying the BNA file to Hyperledger Fabric and the Cloud



Extensive, Familiar, Open Development Toolset

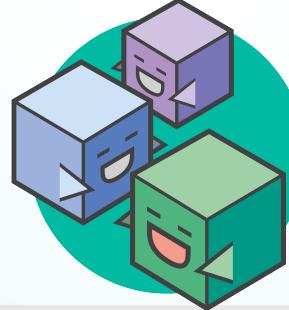


```
asset Animal identi  
  o String animal  
  o AnimalType sp  
  o MovementStatus  
  o ProductionTyp
```

Data modelling



JavaScript
business logic



Web playground

composer-client
composer-admin



Client libraries



Editor support

\$ composer

CLI utilities



Code generation

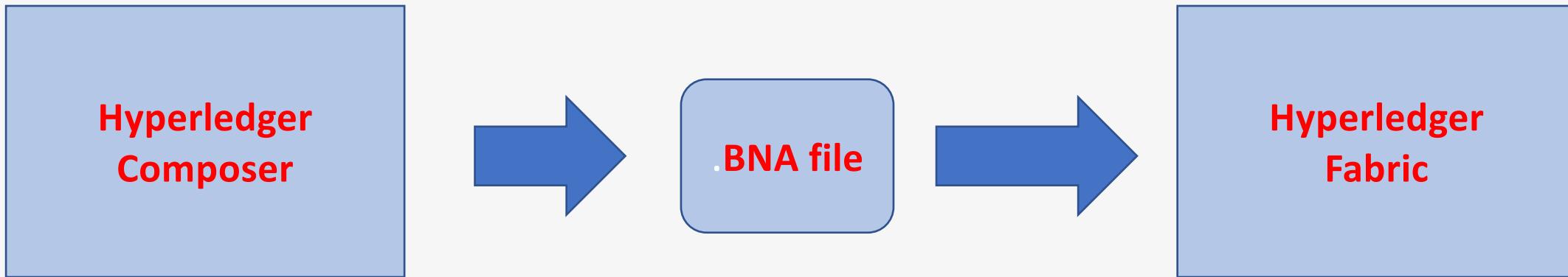
Powered by
 LoopBack
Node.js Framework

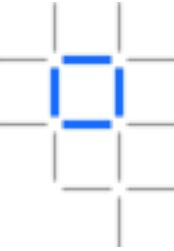


Swagger

Existing systems and
data

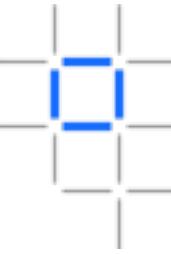
Hyperledger Composer and Ledger





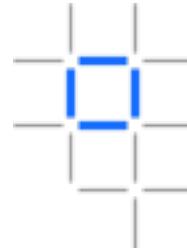
Blockchain Use Cases

What makes a good Blockchain Use Case?



- A Business problem that cannot easily be solved with existing techniques
- An identifiable business network
 - With a shared ledger
 - With Participants, Assets and Transactions
- A need for trust
 - Consensus, Immutability, Finality and Provenance

Re-use existing Blockchain Use Cases



Supply chain, asset registration, identity services, fraud prevention and compliance

IBM Blockchain government point of view

[PDF See the infographic \(50.6KB\)](#)



Payment and digital currency

IBM's universal blockchain payments solution

[PDF See the infographic \(1.4MB\)](#)



Payment and digital currency

IBM announces major blockchain solution to speed up global payments

[→ Read the press release](#)

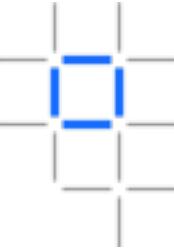


European trade finance network: we.trade

Spreading opportunity to small and medium sized businesses in traditionally underserved markets.

[▶ Watch the video \(02:48\)](#)

<https://www.ibm.com/blockchain/use-cases/>



IBM Code Patterns for Blockchain

Blockchain

Distributed database maintaining a continuously growing list of secured records or blocks.

Read more

Related code patterns



Create and deploy a blockchain network using Hyperledger Fabric SDK Java



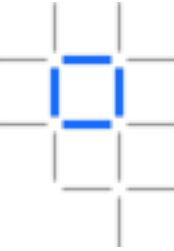
Implement asset securitization on a blockchain ledger

Deploy a React UI that has can submit blockchain transactions and view the state of a blockchain ledger on



Interacting with a blockchain network

Execute transactions and requests against a blockchain network by creating an app to test the network and its

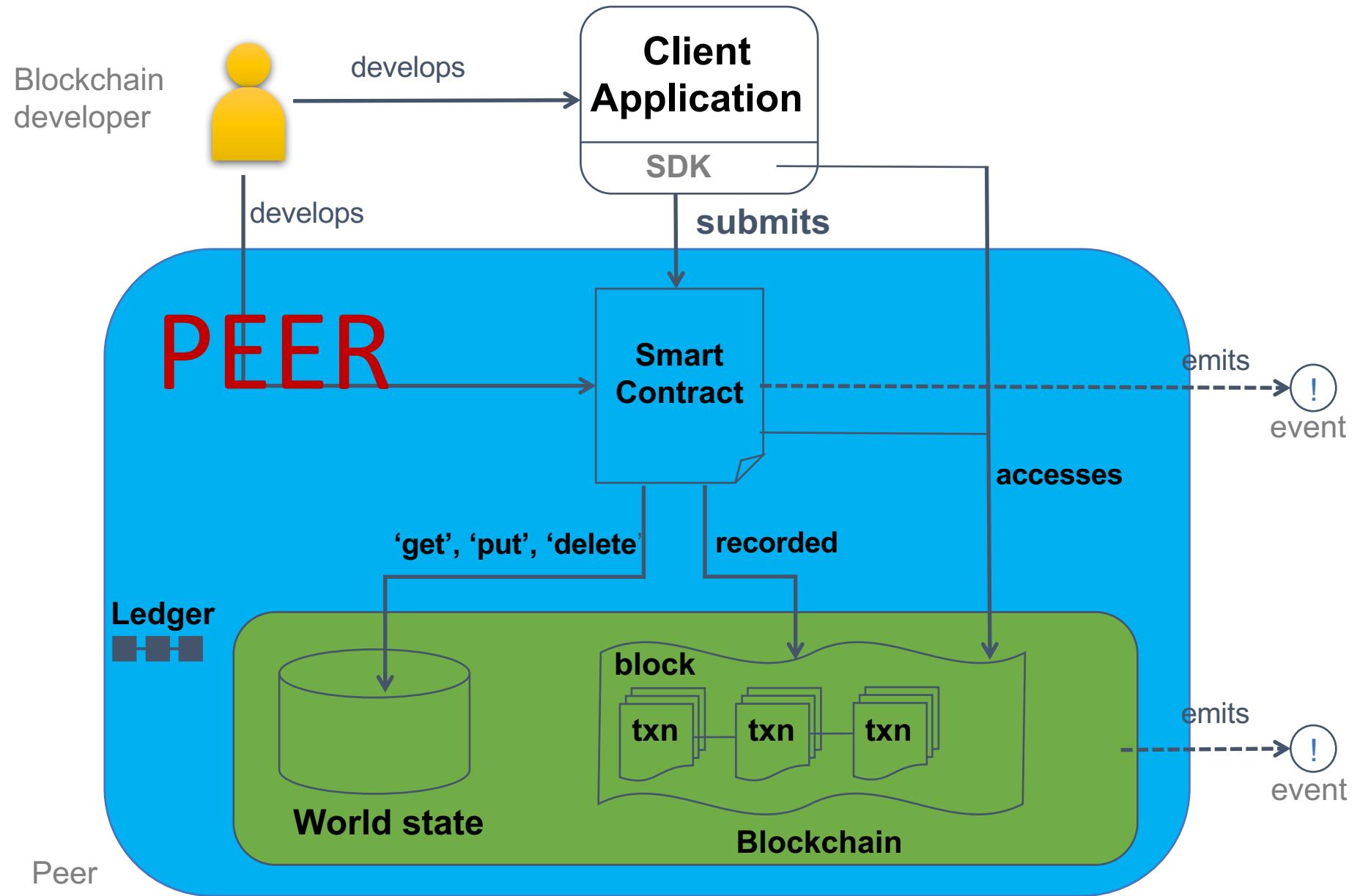
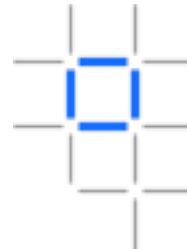


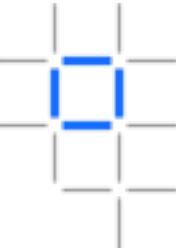
Programming the ledger with Chain Code alias Smart Contracts

Chaincode and Smart Contracts

- Chaincode is a program, written in Go, node.js, and eventually in other programming languages such as Java, that implements a prescribed interface. Chaincode runs in a secured Docker container isolated from the endorsing peer process.
- Similar to smart Contracts and Stored Procedures

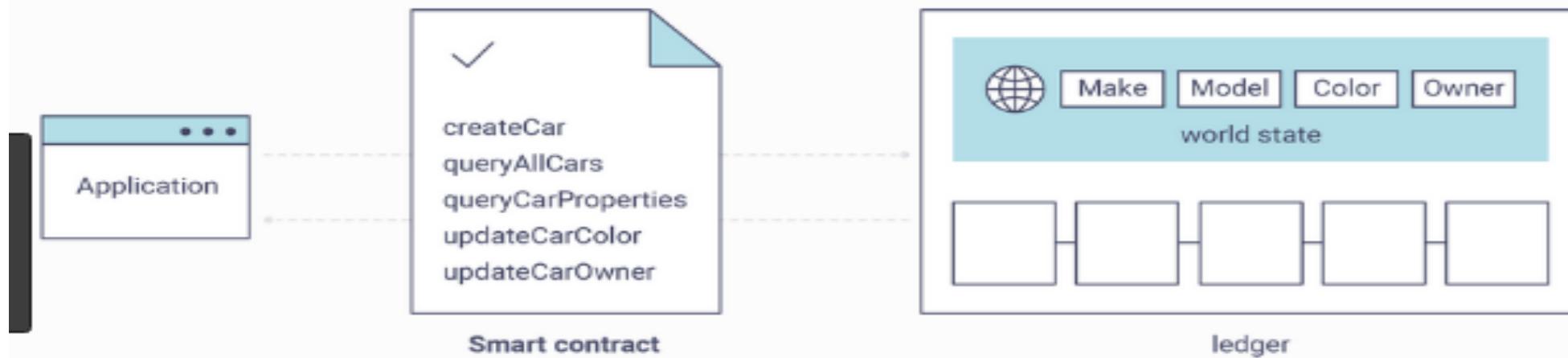
Programming the ledger





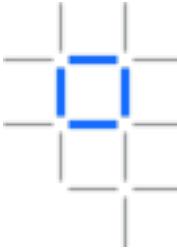
Chaincode, aka Smart Contracts

Below is a representation of how an app would call different functions in chaincode. Each function must be coded against an available API in the chaincode shim interface, which in turn allows the smart contract container to properly interface with the peer ledger.



We can see our `queryAllCars` function, as well as one called `createCar`, that will allow us to update the ledger and ultimately append a new block to the chain in a moment.

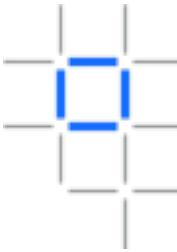
<https://hyperledger-fabric.readthedocs.io/en/release-1.2/chaincode.html>



Two Chaincode Personas

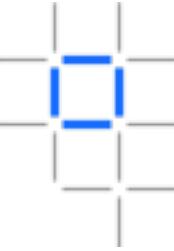
- One, from the perspective of an application developer developing a blockchain application/solution entitled **Chaincode for Developers**, and the other,
- **Chaincode for Operators** oriented to the blockchain network operator who is responsible for managing a blockchain network/
<https://hyperledger-fabric.readthedocs.io/en/release-1.2/chaincode.html>

More on Chaincode

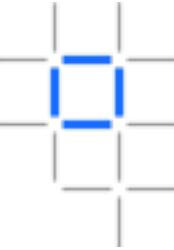


- A chaincode typically handles business logic agreed to by members of the network, so it may be considered as a “smart contract”.
- State created by a chaincode is scoped exclusively to that chaincode and can't be accessed directly by another chaincode.
- However, within the same network, given the appropriate permission a chaincode may invoke another chaincode to access its state.

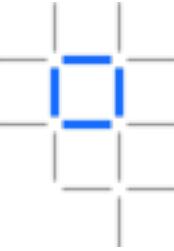
<https://hyperledger-fabric.readthedocs.io/en/release-1.2/chaincode.html>



Your first Blockchain Application

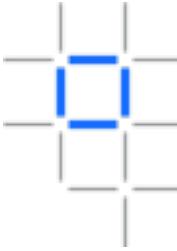


Blockchain in the Cloud

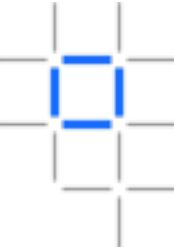


Blockchain Demos

Blockchain Demos

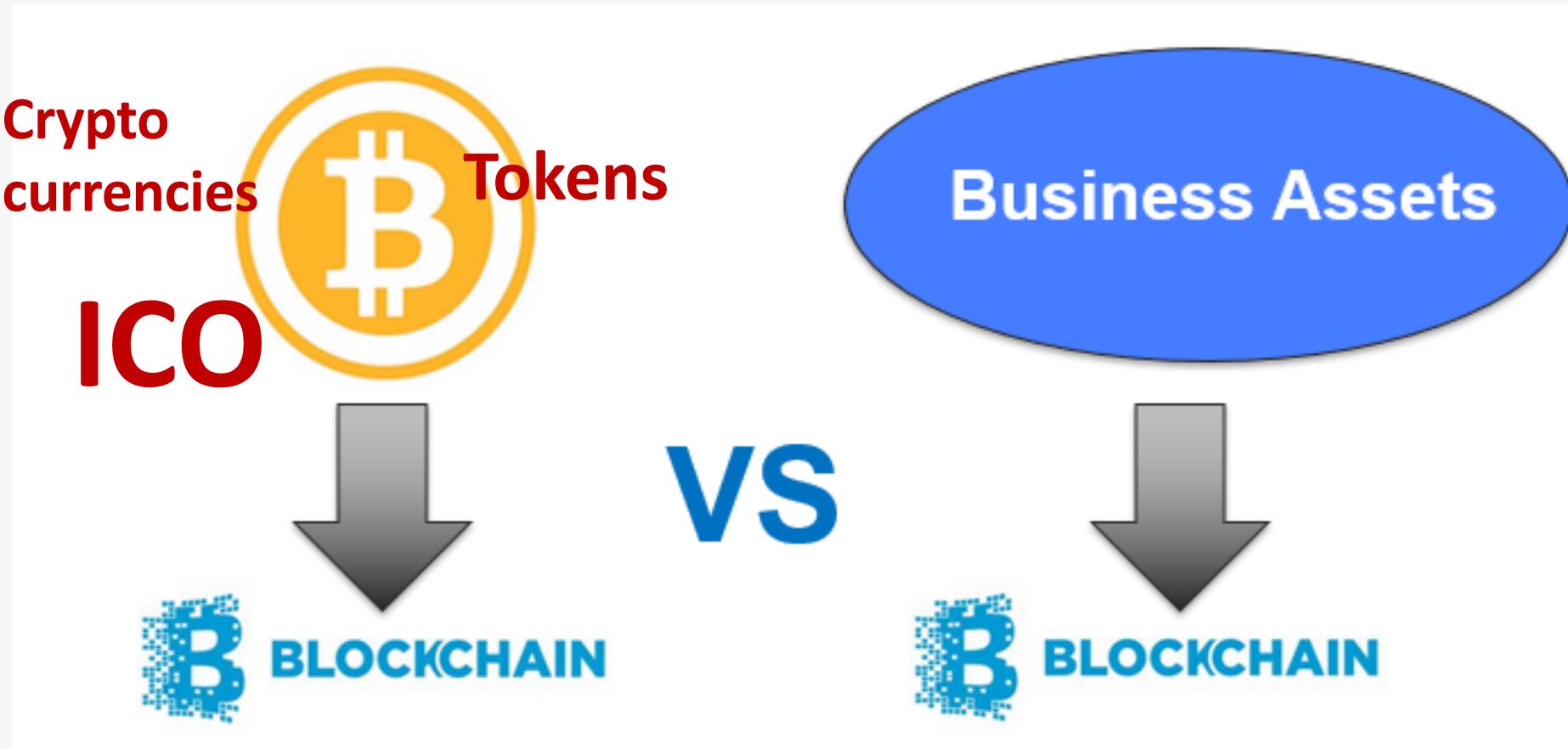


- <http://blockchain-coffee-web.mybluemix.net/>
- Network Name: altnetwork Password: loesenord
- <https://www.ibm.com/developerworks/cloud/library/cl-coffee-with-blockchain-introduction-to-how-blockchain-works/index.html>

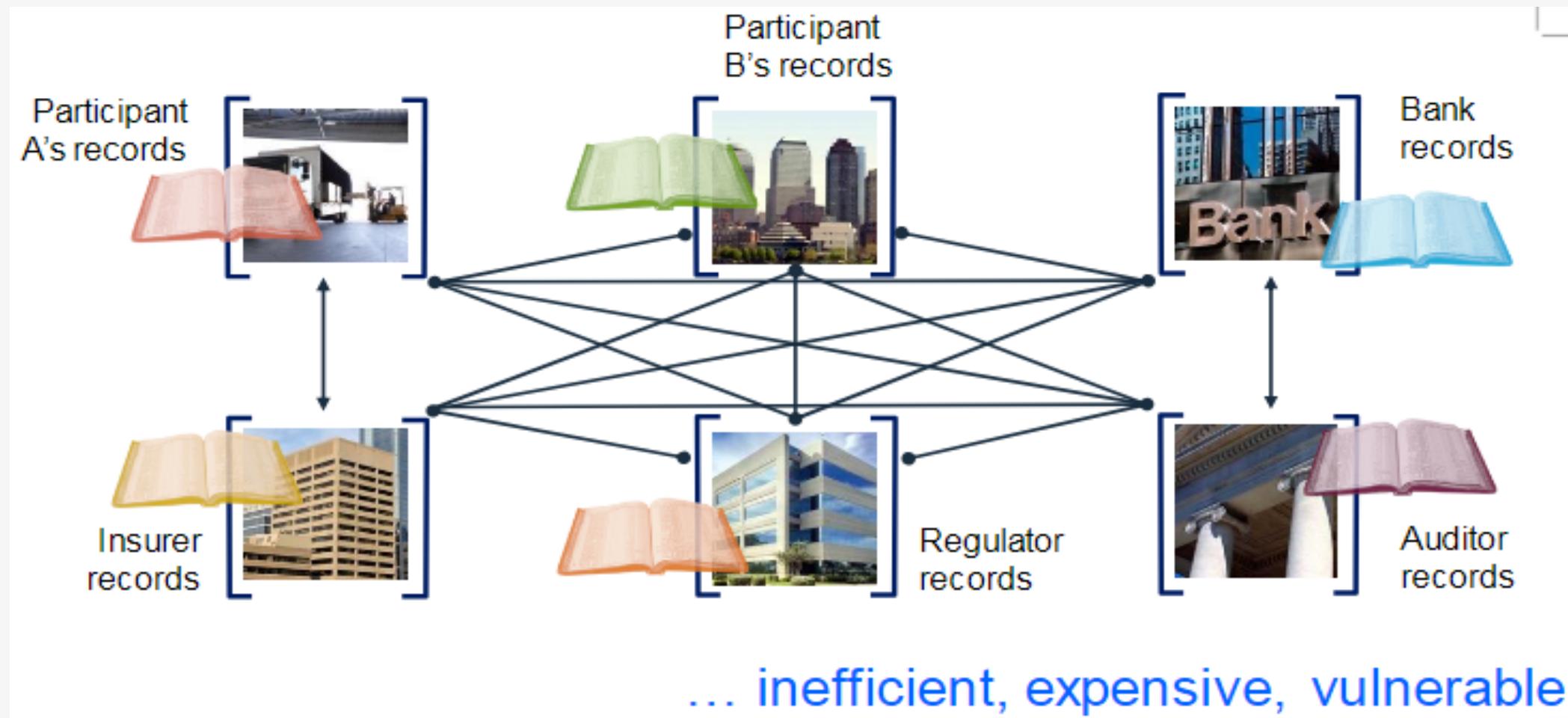


Blockchain Sources

The two versions of Blockchain



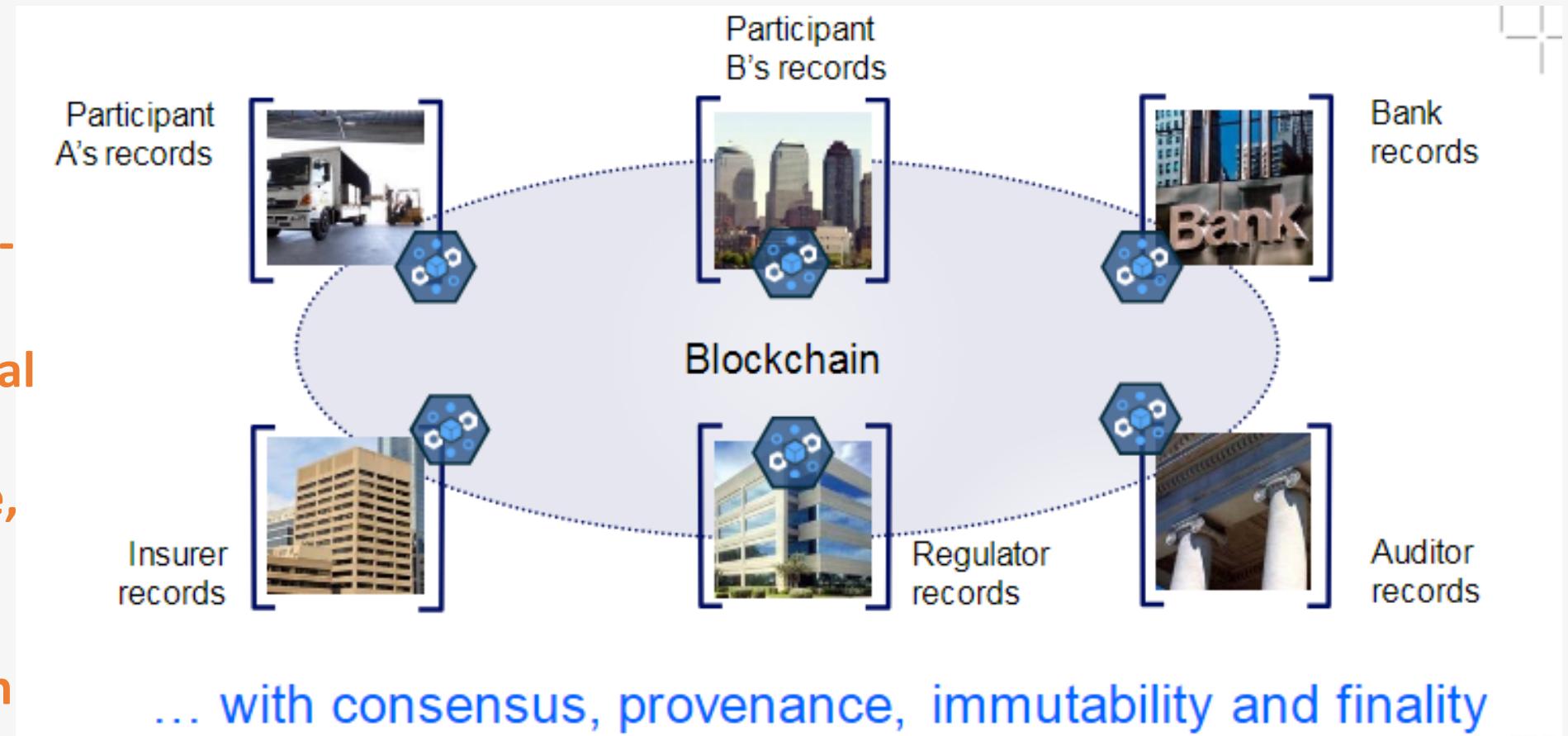
From the multiple Ledger Database

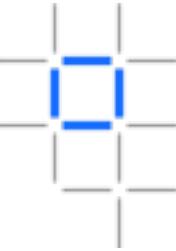


To the shared, replicated, permissioned ledger

No central, third-party mediator, such as a financial institution or clearinghouse, is involved

No reconciliation needed

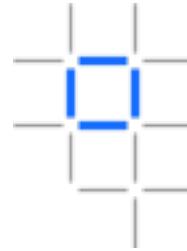




What makes a good Blockchain Use Case?

- A Business problem that cannot easily be solved with existing techniques
- An identifiable business network
 - With a shared ledger
 - With Participants, Assets and Transactions
- A need for trust
 - Consensus, Immutability, Finality and Provenance

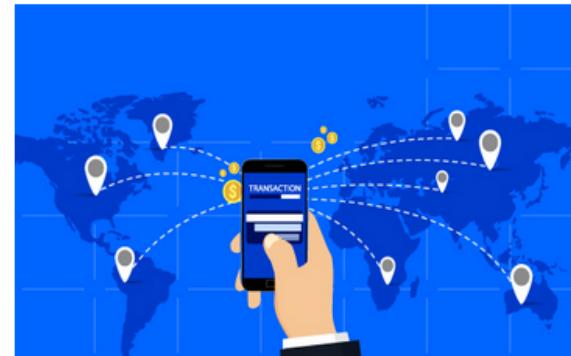
Re-use existing Blockchain Use Cases



Supply chain, asset registration, identity services, fraud prevention and compliance

IBM Blockchain government point of view

[PDF See the infographic \(50.6KB\)](#)



Payment and digital currency

IBM's universal blockchain payments solution

[PDF See the infographic \(1.4MB\)](#)



Payment and digital currency

IBM announces major blockchain solution to speed up global payments

[→ Read the press release](#)



European trade finance network: we.trade

Spreading opportunity to small and medium sized businesses in traditionally underserved markets.

[▶ Watch the video \(02:48\)](#)

<https://www.ibm.com/blockchain/use-cases/>

Facets of distributed, shared ledgers

Network nodes both generate their own data and verify data generated by others	Contain historic record of verified transactions that are easily auditable	Distributed Consensus eliminates costly and inefficient reconciliation processes
No central repository – each node stores identical copies of the ledger	Resilient due to network power and cryptographic integrity	Large economic disincentive for malicious actors

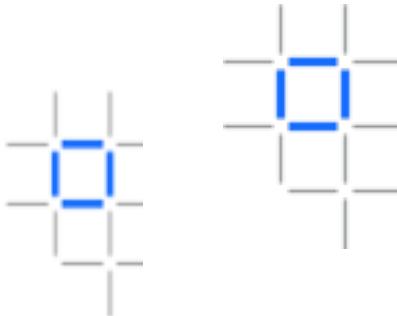
Spectrum of Blockchains

Permissionless Public	Permissionless Private	Permissioned Public	Permissioned Private
Bitcoin, Ethereum	Public Polls	Land Titles, University Degrees	Medical Records

Permissioned vs. Permissionless: Who can write to a Blockchain, i.e. accessibility
Public vs Private: who can read from a Blockchain, i.e. visibility

<https://www.youtube.com/watch?v=-O3ouBdG3Xg>

Why not use a Database?



- A traditional database is centralized
- Everyone needs to trust the administrator managing the database
- There's typically no immutability or provenance



- Distributed databases do not alleviate the trust issue
- There are now more copies to worry about and more administrators



- Blockchain allows the concept of a distributed database to be deployed across an untrusted network
- Something a traditional database cannot handle

Traditional databases cannot be used in untrusted networks

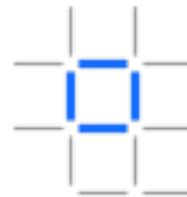
Introducing Hyperledger

**Open source
collaborative effort to
advance **cross-industry**
blockchain
technologies**

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

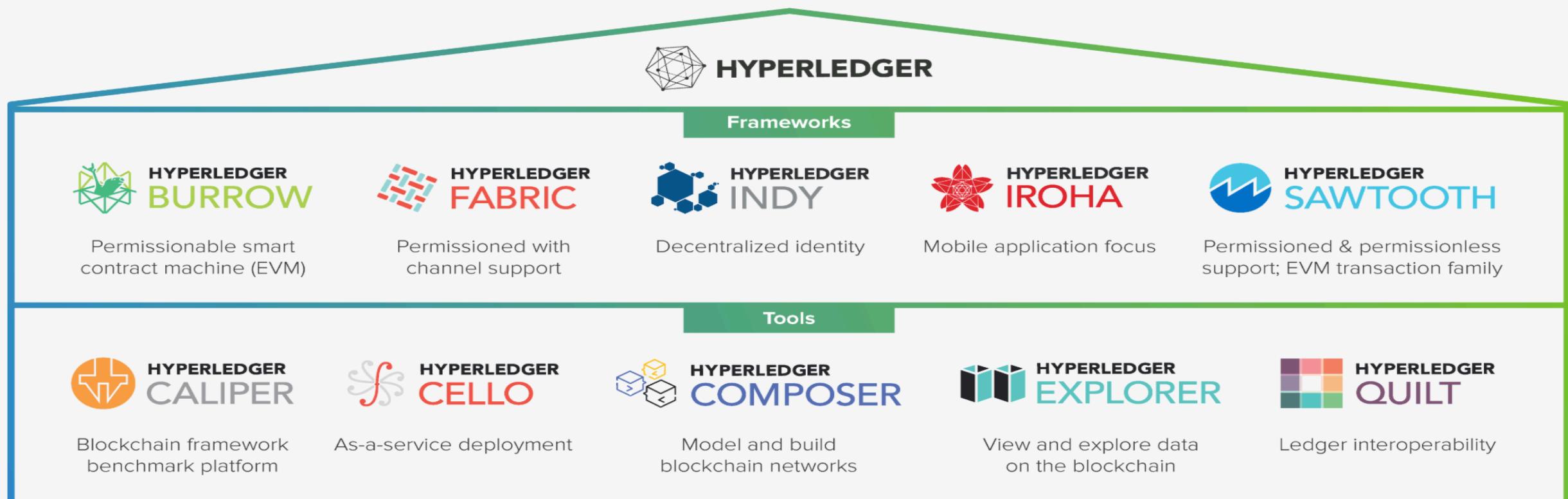
Global collaboration
spanning **finance, banking, IoT, supply chain, healthcare, manufacturing, govt, technology** and more.

Which Blockchain do we want to use?



The Hyperledger Greenhouse

Business Blockchain Frameworks & Tools Hosted by Hyperledger



<https://www.hyperledger.org/>

Consensus algorithms have different strengths and weaknesses



Solo /
No-ops

[Validators apply received transactions without consensus](#)

PROs: Very quick; suited to development

CONS: No consensus; can lead to divergent chains

Example usage: Hyperledger Fabric V1



PBFT-based

[Practical Byzantine Fault Tolerance implementations](#)

PROs: Reasonably efficient and tolerant against malicious peers

CONS: Validators are known and totally connected

Example usage: Hyperledger Fabric V0.6



Kafka /
Zookeeper

[Ordering service distributes blocks to peers](#)

PROs: Efficient and fault tolerant

CONS: Does not guard against malicious activity

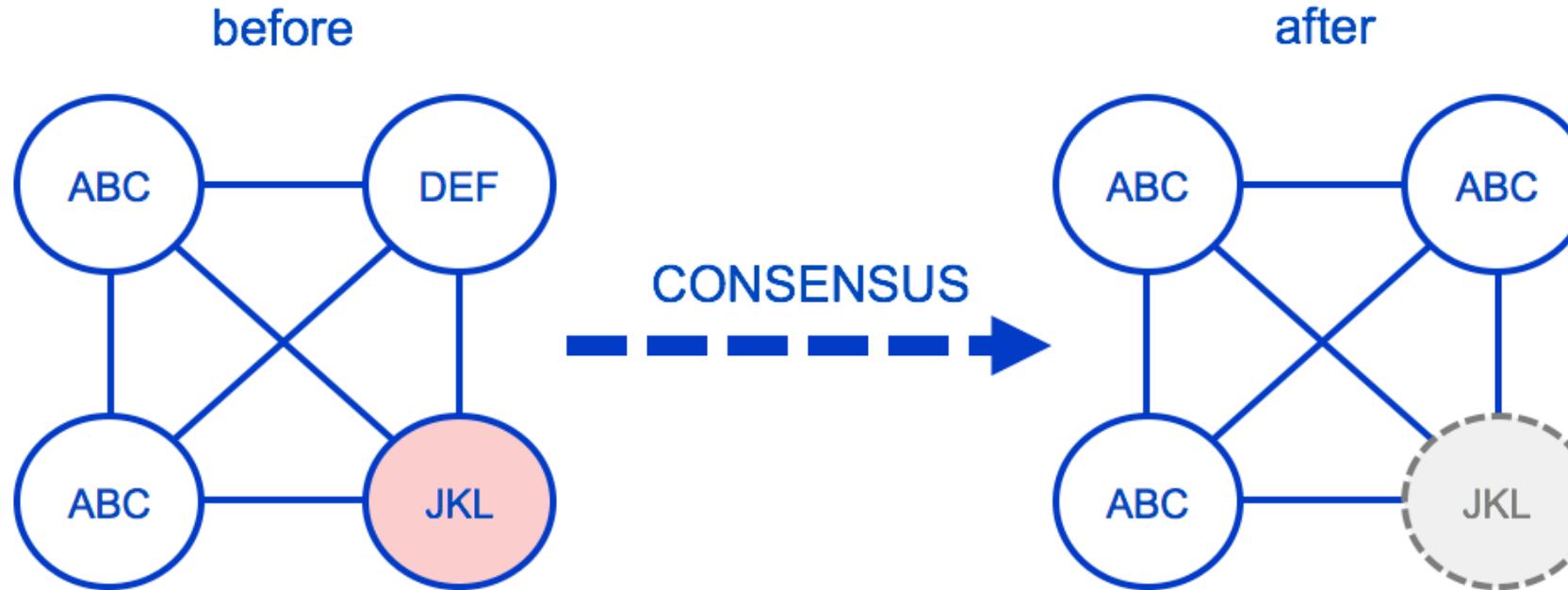
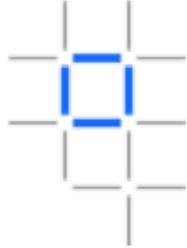
Example usage: Hyperledger Fabric V1



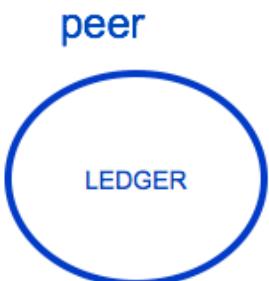
HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

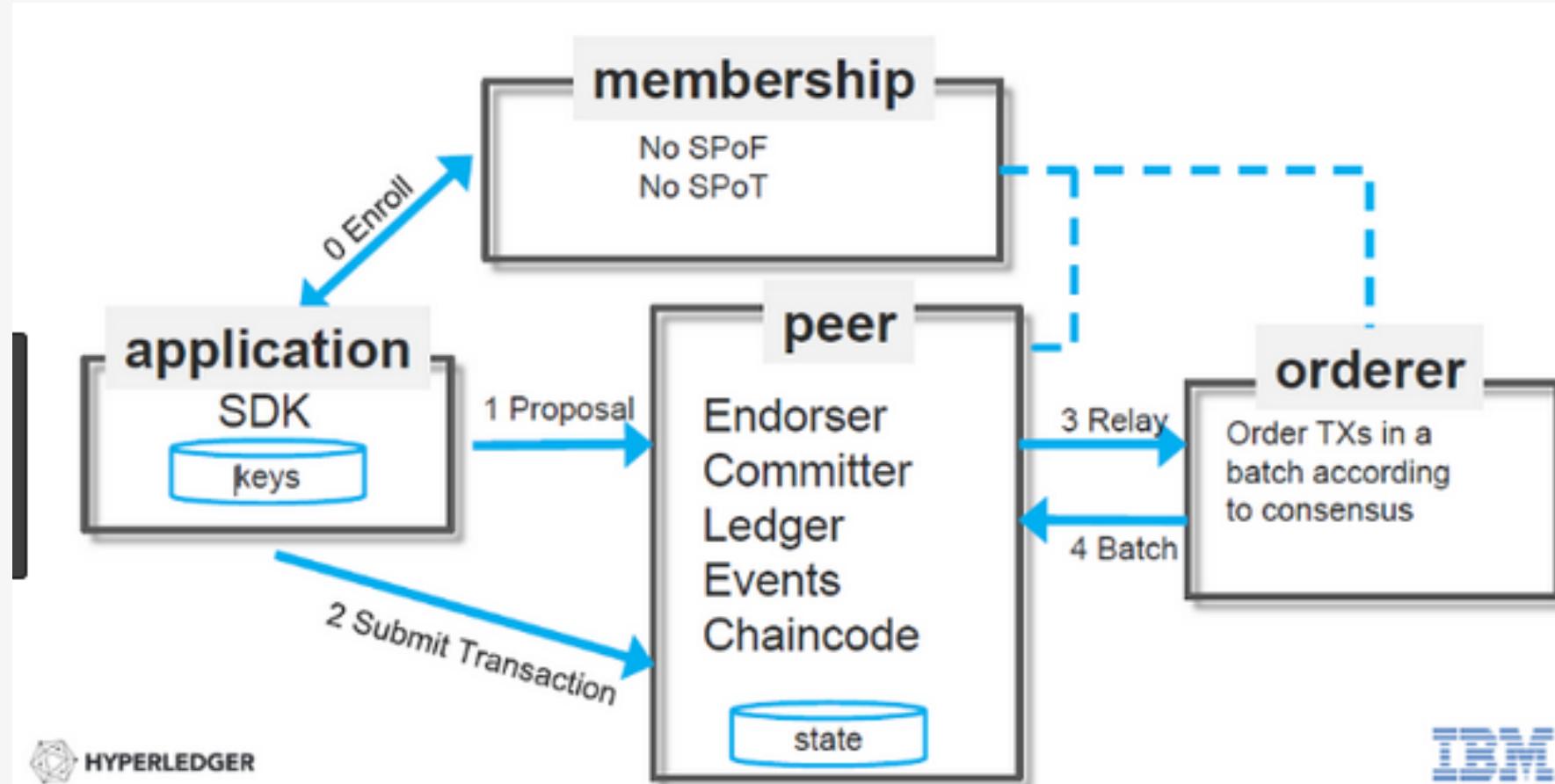
Consensus: The process of maintaining a consistent ledger



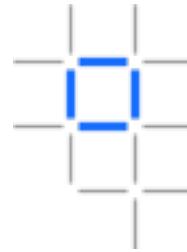
Keep all peers up-to-date
Fix any peers in error
Ignoring all malicious nodes



Hyperledger architecture



Hyperledger Fabric has three components

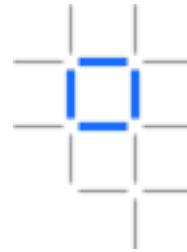


Membership Services – which manages digital certificates and access,

- **Ordering service** – which keeps the peers in alignment,
- and **peers** – which hold the ledger.
 - A peer is also where **chaincode**, also known as smart contracts, are executed.
 - Membership and ordering services align well to being centralized.
- **Peers**, on the other hand, can be distributed and placed anywhere you desire in the world.
- Each member of the Hyperledger network should have their own peer.
- The channels (databases) that a peer subscribes to, and transacts against, determines the member's role and capabilities in the blockchain network

<https://www.ibm.com/blogs/systems/blockchain-how-should-you-organize-your-peers/>

Hyperledger Fabric Architecture

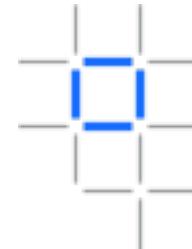


Architecturally, the fabric is extremely flexible, but in practice consider two recommendations.

- **First, you only need one or two peers.** Per a given channel, peers are identical. They execute the same transactions and receive the same code updates in the exact same order as all other peers on that channel. **There is no reason to have lots of peers.**
 - Get two peers if you need the availability – which I would recommend.
- **Secondly, the front-end systems that hold the Hyperledger SDKs (those systems that provide RESTful interfaces) should be aligned to a member's peers.**
 - Which means you want your front-end systems talking to your peers and you want other member's front-end systems talking to their peers.
 - The idea of an SDK talking to all of the peers for consensus purposes makes little to no sense when they will all return the same results.
 - One SDK only needs to talk to one peer or two, for high availability, to get the optimal performance and functional capabilities of the fabric.

<https://composer-playground.mybluemix.net/>

Welcome to Hyperledger Composer Playground!



In this web sandbox, you can deploy, edit and test business network definitions. Have a play and learn what Hyperledger Composer Playground is all about.

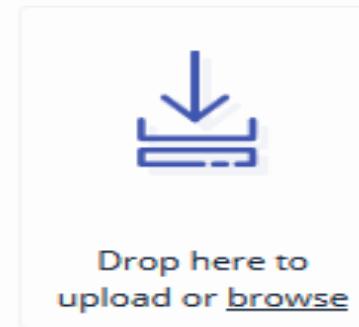
Let's Blockchain!



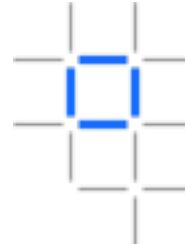
Not sure where to start? View our Playground tutorial.

Choose a Business Network Definition to start with:

Choose a sample to play with, start a new project, or import your previous work



Samples on npm



Blockchain

Distributed database maintaining a continuously growing list of secured records or blocks.

Read more

Related code patterns



Create and deploy a blockchain network using Hyperledger Fabric SDK Java



Implement asset securitization on a blockchain ledger

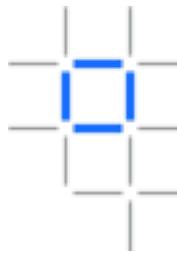
Deploy a React UI that has can submit blockchain transactions and view the state of a blockchain ledger on



Interacting with a blockchain network

Execute transactions and requests against a blockchain network by creating an app to test the network and its

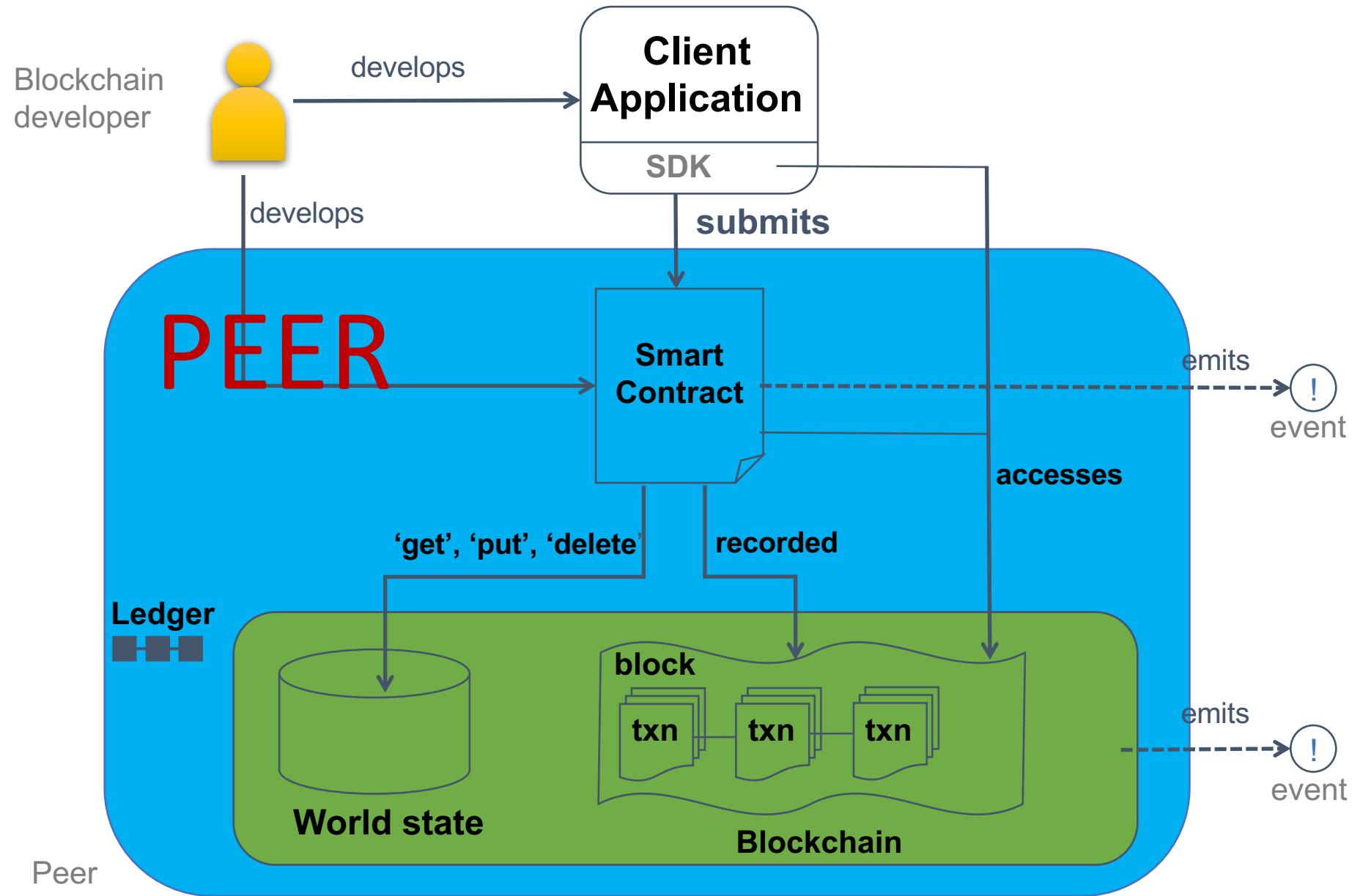
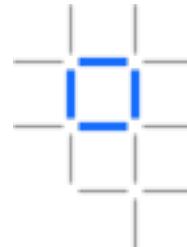
Which user roles do you need in a business network?



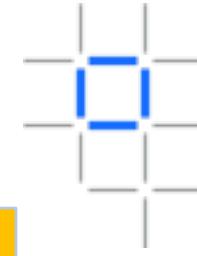
- **The consortium owner:** This is the organization with the vision for the network, its business benefits and how it will disrupt how business is done. Today, for practical purposes, it is best if there is only one consortium owner, since that entity will be responsible for making the network work.
- **Members:** They are part of the Hyperledger network. **They have a peer, which means they can store data, execute chain code (smart contracts), and participate in the consensus process** (if the consortium owner allows it). Members bring substance and validity to the network, and it is their data and intelligence that is shared. As the network matures, members can move up in status to become consortium owners.
- **General users:** These are users of the network and they are sponsored and enabled by the consortium owner and the members. **They can access and input data, but they do not own data.** As with members, as the network matures, general users can become members if they bring value to the network.

<https://www.ibm.com/blogs/systems/blockchain-who-should-be-in-your-consortium/>

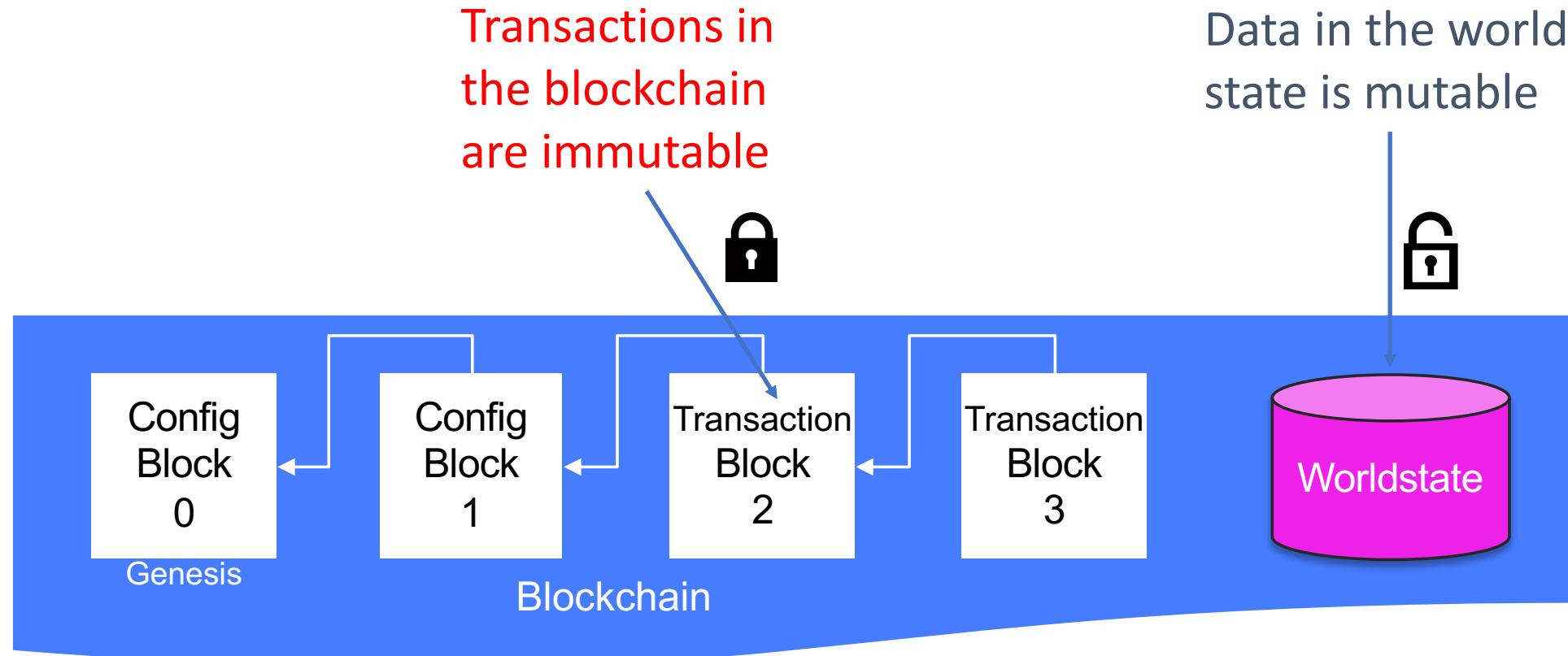
Programming the ledger



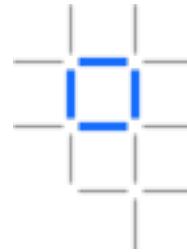
The “Right to Erasure” and blockchain



Enables an individual to request the deletion or removal of their personal data



General Data Protection Regulation (GDPR) Solution – Store data off-chain



- **What**

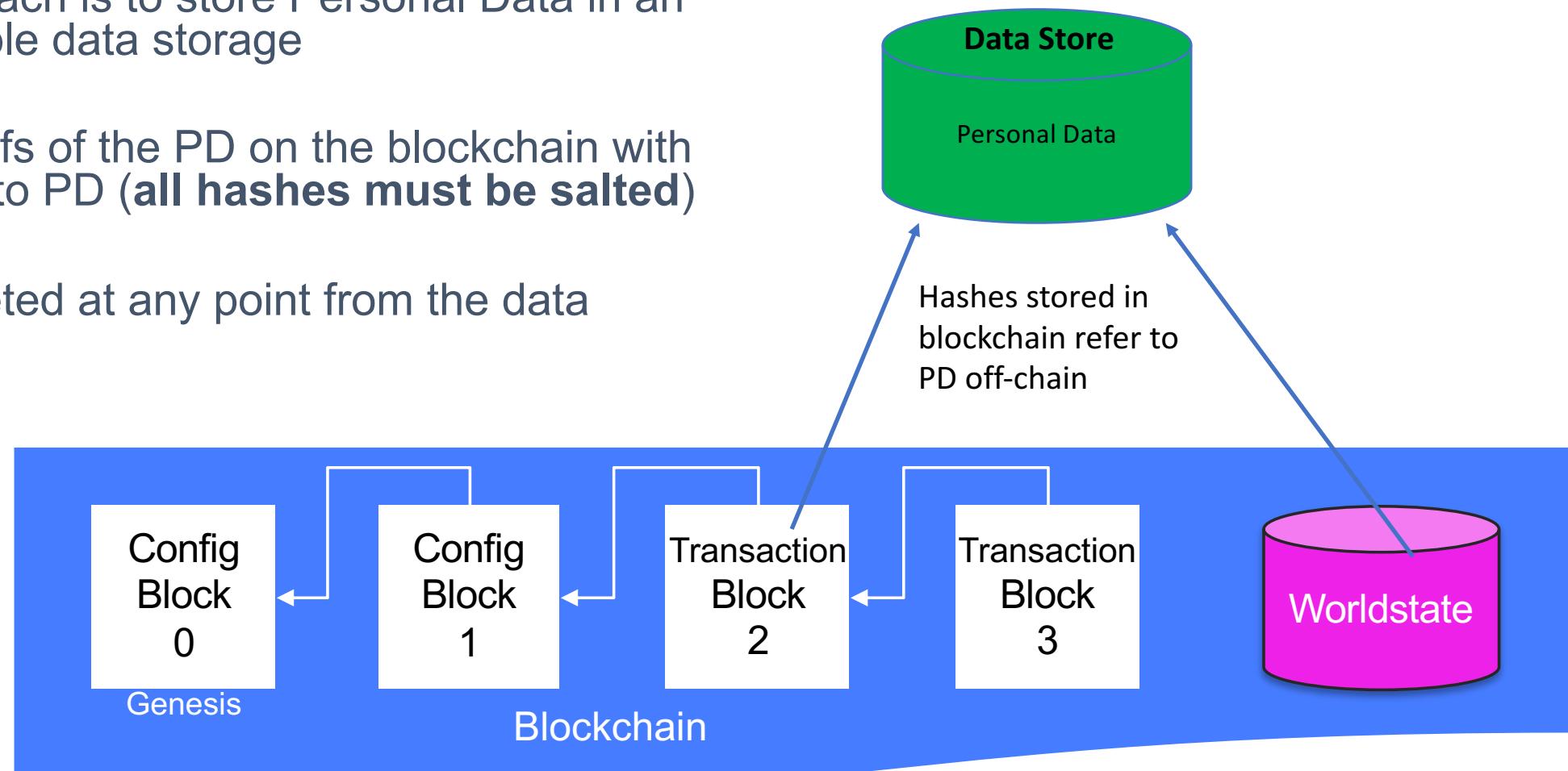
- The only approach is to store Personal Data in an off-chain mutable data storage

- **How**

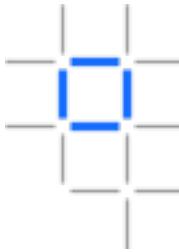
- Store only proofs of the PD on the blockchain with hashes linking to PD (**all hashes must be salted**)

- **Why**

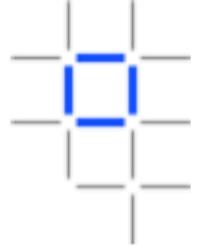
- PD can be deleted at any point from the data store(s)



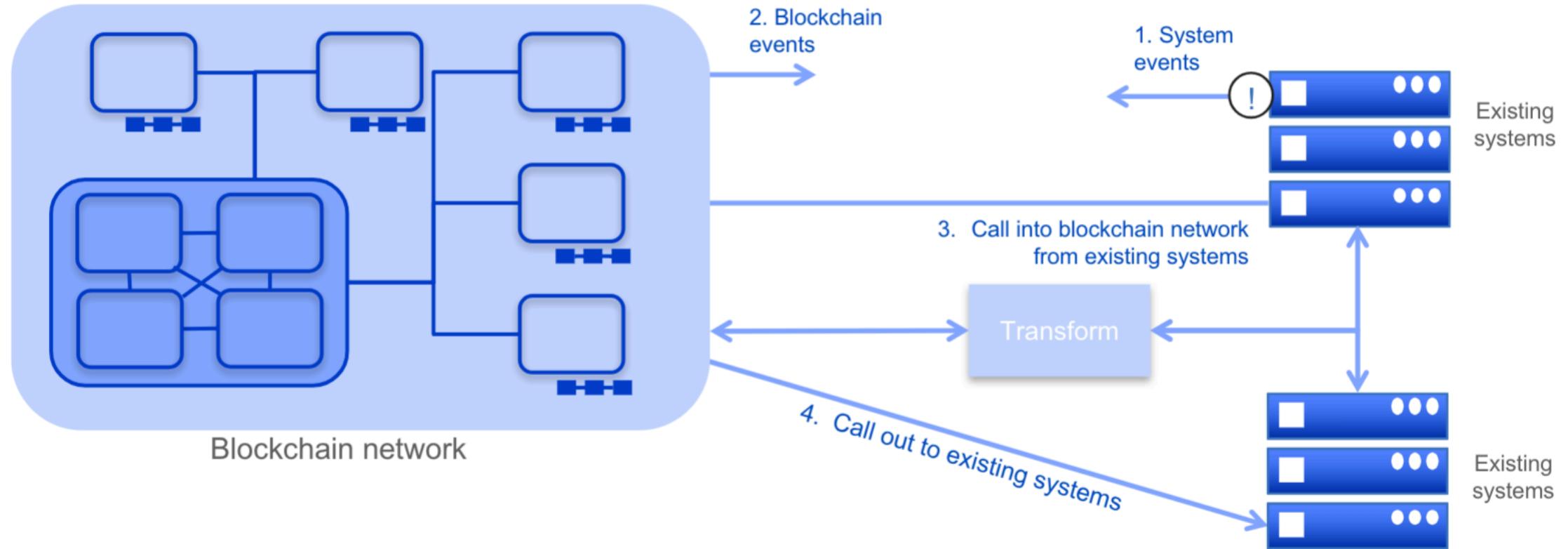
Off chain, the problem with the Systems of Record



- All data can not be stored on the Blockchain
 - Example X-ray images
 - Diamonds
 - Art works
 - Automobiles
 - Aircraft parts
 - Food



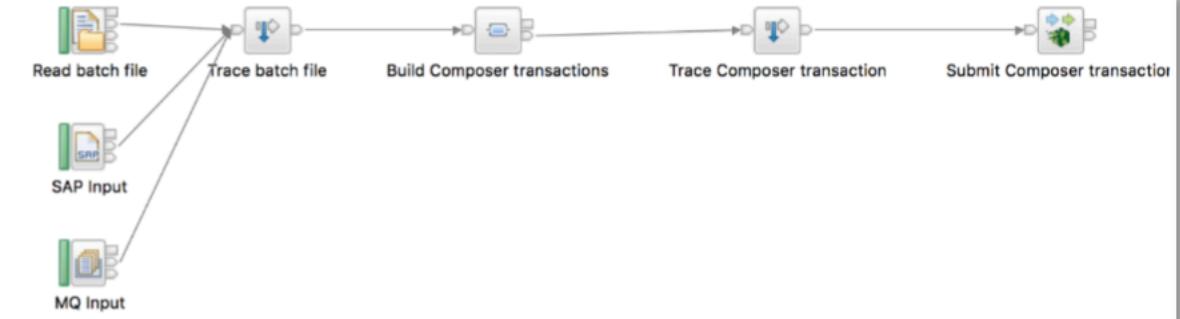
Integrating with existing systems - possibilities





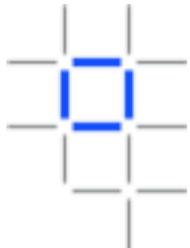
Integrating with existing systems – using middleware

- Blockchain is a network system of record
- Two-way exchange
 - Events from blockchain network create actions in existing systems
 - Cumulative actions in existing systems result in Blockchain interaction



- Transformation between blockchain and existing systems' formats
 - GBO, ASBO is most likely approach
 - Standard approach will be for gateway products to bridge these formats
 - Gateway connects to peer in blockchain network and existing systems
- Smart contracts can call out to existing systems
 - Query is most likely interaction for smart decisions
 - e.g. all payments made before asset transfer?
 - Warning: Take care over predictability... transaction must provide same outputs each time it executes





Non-determinism in blockchain

- Blockchain is a distributed processing system
 - Smart contracts are run multiple times and in multiple places
 - As we will see, smart contracts need to run deterministically in order for consensus to work
 - Particularly when updating the world state
- It's particularly difficult to achieve determinism with off-chain processing
 - Implement services that are guaranteed to be consistent for a given transaction, or
 - Detect duplicates for a transaction in the blockchain, middleware or external system

random()

getExchangeRate()

getDateTime()

getTemperature()

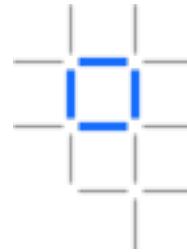
incrementValue
inExternalSystem(...)

Hyperledger adoption

- Change Healthcare's Intelligent Healthcare Network deploys enterprise-scale, production-ready blockchain powered by Hyperledger Fabric 1.0.
<https://www.hyperledger.org/event/himss-2018>
- Illinois Launches Blockchain Pilot to Digitize Birth Certificates
<https://www.coindesk.com/illinois-launches-blockchain-pilot-digitize-birth-certificates/>

Getting started with Hyperledger

Blockchain Use Cases



Supply chain, asset registration, identity services, fraud prevention and compliance

IBM Blockchain government point of view

[PDF See the infographic \(50.6KB\)](#)



Payment and digital currency

IBM's universal blockchain payments solution

[PDF See the infographic \(1.4MB\)](#)



Payment and digital currency

IBM announces major blockchain solution to speed up global payments

[→ Read the press release](#)

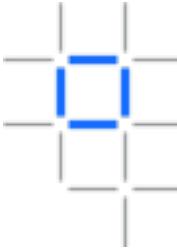


European trade finance network: we.trade

Spreading opportunity to small and medium sized businesses in traditionally underserved markets.

[▶ Watch the video \(02:48\)](#)

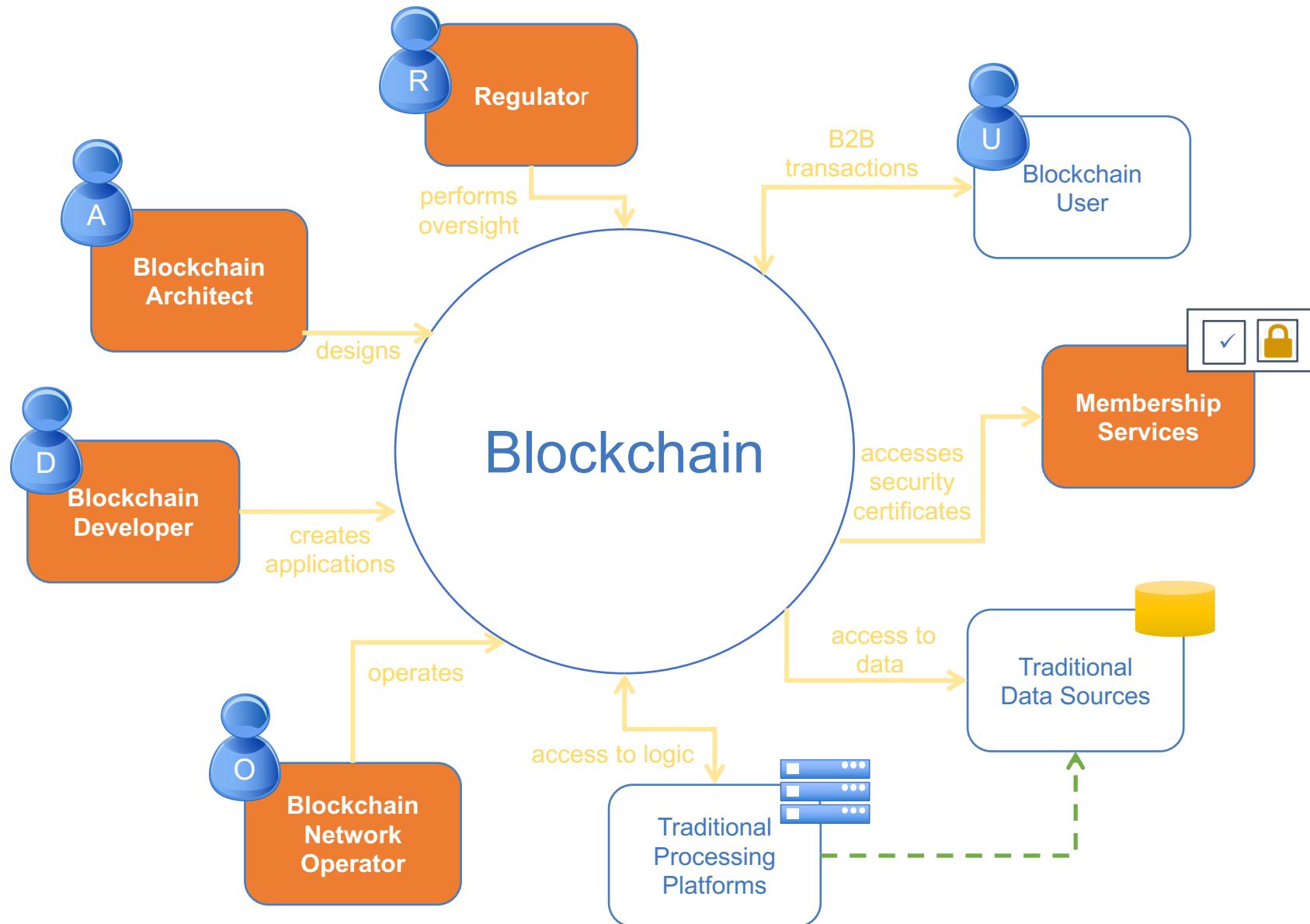
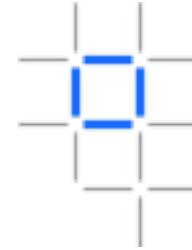
<https://www.ibm.com/blockchain/use-cases/>



What makes a good Blockchain Use Case?

- - Identifying a good blockchain use-case is **not always easy!**
 - However there should always be:
 1. A **business problem** to be solved
 - That cannot be more efficiently solved with other technologies
 2. An identifiable **business network**
 - With Participants, Assets and Transactions
 3. A need for **trust**
 - Consensus, Immutability, Finality or Provenance

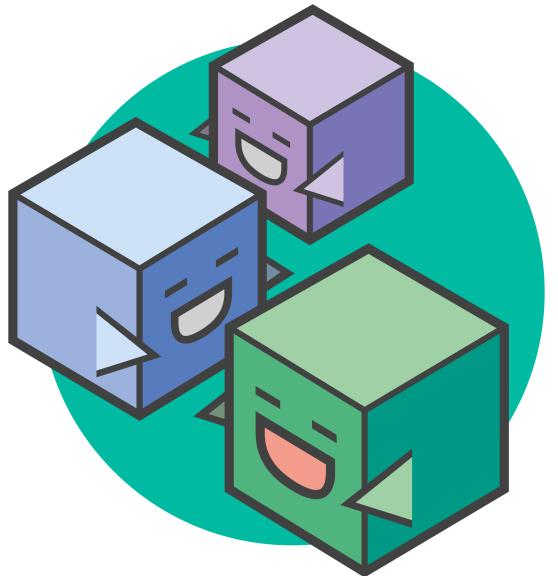
Actors in a Blockchain solution



Hyperledger Composer: Accelerating time to value creating chaincode

<http://composer-playground.mybluemix.net>

- A suite of high level application **abstractions** for business networks
- Emphasis on business-centric vocabulary for quick



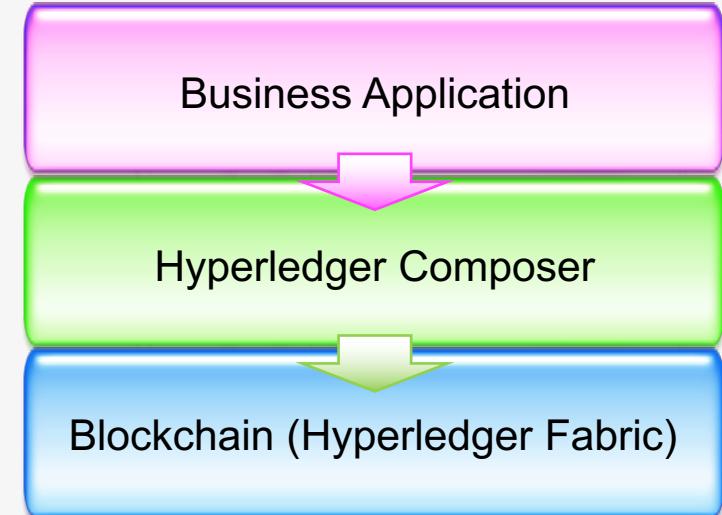
- Features increase understanding and flexibility
 - Model your business networks, test and expose via APIs
 - Applications invoke APIs transactions to interact with business network
 - Integrate existing systems of record using JSON/REST
 - Try it in your web browser now: <http://composer-playground.mybluemix.net/>

Saves time

Reduces risk

Increases flexibility

Increases understanding



Chaincode and Smart Contracts

- Chaincode is a program, written in Go, node.js, and eventually in other programming languages such as Java, that implements a prescribed interface. Chaincode runs in a secured Docker container isolated from the endorsing peer process.
- Similar to smart Contracts and Stored Procedures

Hyperledger Composer Playground

Web my-basic-sample

Define Test admin ▾

FILES

About
README.md, package.json

Model File
models/sample.cto

Script File
lib/sample.js

Access Control
permissions.acl

Add a file... Export

UPDATE NETWORK

From: 0.2.6-20180530153450

To: 0.2.6-deploy.0

Deploy changes

© 2017 IBM Corporation

Define Test admin ▾

About File README.md

Basic Sample Business Network

This is the "Hello World" of Hyperledger Composer samples, which demonstrates the core functionality of Hyperledger Composer by changing the value of an asset.

This business network defines:

Participant SampleParticipant

Asset SampleAsset

Transaction SampleTransaction

Event SampleEvent

SampleAssets are owned by a SampleParticipant, and the value property on a

Hyperledger Composer and Ledger



Hyperledger Fabric – Blockchain for Business



Open Governance

Anyone can join or contribute



Built from the ground up for enterprise

With a maturity model to help companies move to production



Performance

Designed for speed and scalability



Confidentiality and privacy

Built-in channels for isolation and membership services for signing and encryption. Supports IBM High Security Business Network.



Modularity and flexibility

Choice of consensus algorithms and programming languages

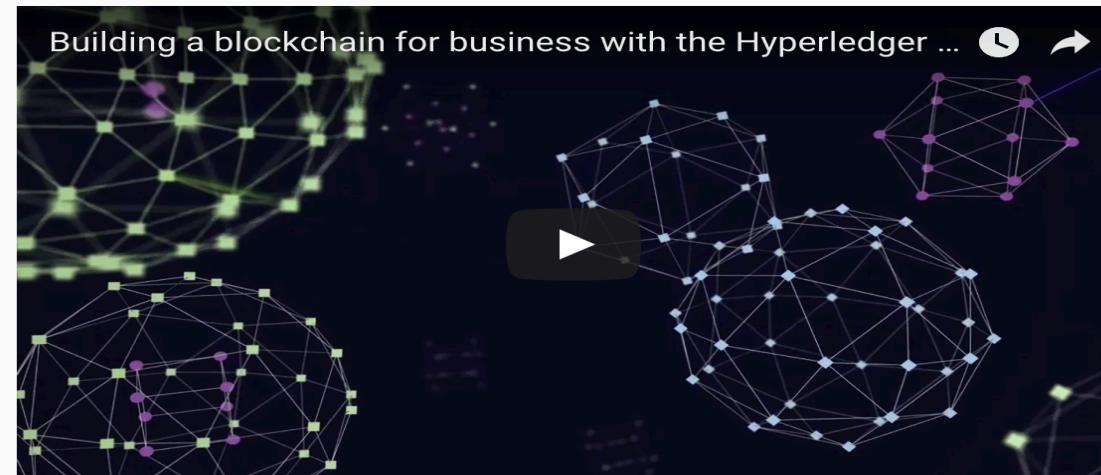
<https://hyperledger-fabric.readthedocs.io/en/release-1.2/>

A screenshot of the Hyperledger Fabric documentation website. The header features the "HYPERLEDGER FABRIC" logo with a red and teal geometric pattern. Below the header is a navigation bar with links to GitHub, Docker, Docker Compose, and YouTube. The main content area contains a sidebar with links to "Introduction", "What's new in v1.2", "Getting Started", "Key Concepts", "Tutorials", "Operations Guides", "Commands Reference", "Architecture Reference", "Frequently Asked Questions", "Contributions Welcome!", "Glossary", and "Releases". At the bottom of the sidebar are "Read the Docs" and "v: release-1.2" buttons.

A Blockchain Platform for the Enterprise



Enterprise grade permissioned distributed ledger platform that offers modularity and versatility for a broad set of industry use cases.



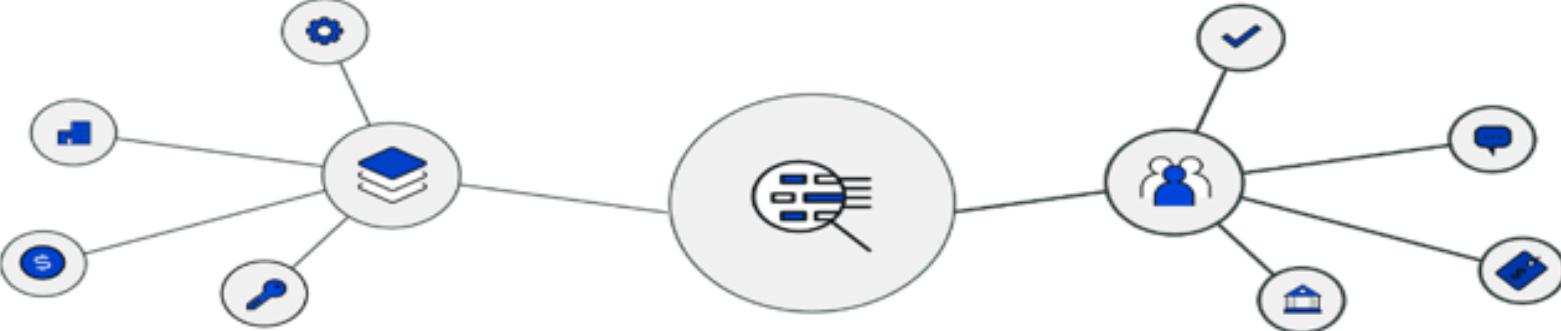
Deploy Hyperledger apps in the cloud

Blockchain /

 Blockchain-BostonWed

Location: US South Org: Developer Advocacy Space: dev

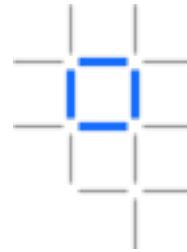
Welcome back, alf



Use the Network Monitor to view and add network nodes, create channels, install chaincode applications, and test confidential transactions.

[Enter Monitor](#)

How do we host our Blockchain application?



IBM Cloud



IBM Cloud Catalog

 blockchain

Filter

Platform

Blockchain

The service enables the creation of blockchain business networks with ownership and control distributed across different organizations.

Blockchain



Utilize IBM's Blockchain
Technology within IBM Cloud

IBM

IBM Blockchain Platform is a flexible software-as-a-service offering that is delivered via the IBM Cloud. It enables network members to quickly get started developing and easily move to a collaborative environment. The platform simplifies your blockchain journey of developing, governing, and operating a network. Choose a membership plan based on your ecosystem needs.

IBM

[View Docs](#) [Terms](#)

AUTHOR IBM

PUBLISHED 07/05/2018

Features

- Use the IBM Blockchain Platform to simplify the developmental, governmental, and operational aspects of creating a blockchain solution. The following plans enable you to easily migrate from POC to pilot, all the way through to production on a secure, high performance, and fully scalable production network that you can't outgrow.
- For more information on developing, governing, and operating your blockchain network, see <https://ibm.biz/bcdocumentation>.
- Get \$500 towards your first network with Starter Plan, featuring an easy-to-use UI to reduce network administration and governance time, an iterative development platform and basic service levels for pilot evaluation or pre-production POCs.*
- Enterprise Plan offers a secure production environment and advanced service levels for production grade deployment, application development, and production testing.

Images

Click on images to enlarge and view screen captures, slides, or videos. Screenshots show the user interface for the

Need Help?

[Contact IBM Cloud Sales](#) 

Already have an account?

[Log in](#)

[Sign up to Create](#)

Overview

View and manage network resources for your organization. You can stop or start your resources and view the log file of the resource by selecting View Logs under "Actions". [Learn more](#)

MY NETWORK

Overview

Members

Channels

Notifications

APIs

MY CODE

Develop code

Install code

Try samples

[Connection Profile](#)[Add Peers](#)

Type	Name	Status	Actions
Orderer	orderer	Running	⋮
CA	org1-ca	Running	▶ ⏺ ⋮
Peer	org1-peer1	Running	▶ ⏺ ⋮

Members

MEMBERS (2/15)	MSP ID	REQUESTER	STATUS
Company A alf@us.ibm.com	org1	--	● Joined
Company B alf@us.ibm.com	org2	--	● Joined

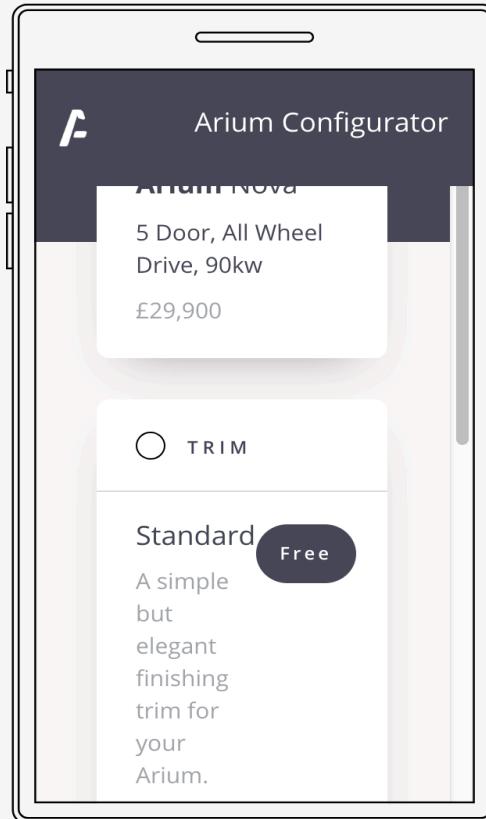
Channels



Search Channels

ID	TIME CREATED	BLOCK HEIGHT	PEERS	A
defaultchannel	07/18/18 06:08 PDT	1	org1-peer1	

Run Blockchain apps in the IBM Cloud



Thank You!

alf@us.ibm.com

<https://hyperledger.org>

<https://ibm.com/blockchain>

