



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

# Hyperledger Fabric

Julian Gordon - VP, Asia Pacific, Hyperledger

Candace Chan - Blockchain Industry Solution Manager, IBM Hong Kong

David Liu - Contributor of Hyperledger Fabric

Edmund To - Organizer of Hyperledger Meetup Hong Kong

# Housekeeping

- Thank you to our supporting organizations
- Volunteer
- Community
- Remote Participants

# MeetUp with Hyperledger Executive Director - Brian Behlendorf

- Date: Friday 22 June
- Time: Tbc, likely 6.30pm-8pm
- Venue: Tbc



- About Brian Behlendorf - Brian is a leading figure globally in blockchain and open source. He was a primary developer of the Apache Web server, the most popular web server software on the internet, and a founding member of the Apache Software Foundation.

# Agenda

- Introduction of Hyperledger and the Linux Foundation- Julian Gordon
- Blockchain use cases Part 1
- Blockchain use cases Part 2 - Candace Chan
- Adoption Model & Business Model
- About Hyperledger Fabric - David Liu
  - Introduction
  - Architecture
- Topic 1: Before to production
- Topic 2: TPS and Caliper
- How to Contribute to Fabric
- Q&A



**HYPERLEDGER**

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

# Hyperledger, open source and the future of blockchain

Julian Gordon  
VP, Asia Pacific, Hyperledger  
[jgordon@linux.com](mailto:jgordon@linux.com)

June 2018

# 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  
chains, healthcare,  
manufacturing,  
technology and more.

# Look to The Linux Foundation



Thankfully, that's where The Linux Foundation® comes in. For the last 16 years, The Linux Foundation has provided unparalleled support for open source communities through financial and intellectual resources, governance structure, IT infrastructure, services, events, and training.

Dedicated to building sustainable ecosystems around open source projects, The Linux Foundation is working with the global technology community to solve the world's hardest problems through open source and **creating the largest shared technology investment in history**.

The Linux Foundation is the umbrella organization for **more than 60 open source projects** accelerating open technology development and commercial adoption. Some of the game-changing initiatives hosted by The Linux Foundation include:



# The Hyperledger Greenhouse

## Infrastructure

Technical, Legal,  
Marketing, Organizational

Ecosystems that accelerate  
open development and  
commercial adoption

Cloud Foundry

Node.js



Hyperledger

Open Container  
Initiative

## Frameworks

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

Hyperledger  
**Fabric**

Hyperledger  
**Sawtooth**

Hyperledger  
**Iroha**

Hyperledger  
**Indy**

Hyperledger  
**Burrow**

## Tools

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

Hyperledger  
**Composer**

Hyperledger  
**Cello**

Hyperledger  
**Explorer**

Hyperledger  
**Quilt**

Hyperledger  
**Caliper**

# Hyperledger Momentum

**240+**

Members  
(40+ in China)

**10**

Hyperledger  
projects

**2**

Production 1.0  
releases

**3.6M+**

Lines of Code

**72K+**

Enrolled in 1st  
Edx.org course

Hosted the most  
popular webinar in  
history of The  
Linux Foundation  
on Hyperledger  
Fabric v1.0

**100+**

Meetups  
Worldwide

**28K+**

Meetup  
Participants

**35K+**

Twitter Followers

**1.5K+**

Avg. monthly press  
mentions in 2017

# 240+ Hyperledger members

Hyperledger Premier Members Serving on the Governing Board.



# 240+ Hyperledger members

## General Members

1WorldBlockchain  
8base  
8Common  
ABN Amro  
Aetna  
Agavon  
AlphaPoint  
Altoros Americas LLC  
Aktsiaselts Eurostep  
Digital  
AMIHAN Global  
ANNE  
ANZ  
ArcBlock  
B9lab  
BBVA  
Beijing Botuzongheng Science & Technology Co., Ltd.  
Beijing RZXT Technology Development  
Beijing Xiaomi Mobile Software  
Belink Technologies  
Bitmark  
BitSE  
Blockchain  
Blockchain Training Alliance  
BLOCKO Inc.  
Bloq  
BNP Paribas

Broadridge Financial Solutions

BTS

CA Technologies

Calastone

Capgemini

Centra Tech

Chain Connected

ChainNova

China Merchants Bank

China Minsheng Bank

CITIC

Clause, Inc

Cloudsoft Corporation Ltd.

CLS Bank International

CME Group

Cognition Foundry

Coinplug

CollectorIQ Inc.

Cuscal Limited

Data Deposit Box

Dealer Market Exchange

Deloitte Consulting LLP

Deutsche Börse Group

DLT Labs

Easy Visible Supply

Chain

Management

EBPI BV

Elementrem

Eli Lilly & Company  
Embleema

Energy Blockchain Labs

Ernst & Young

Factom Foundation

Filament

FORFIRM

ForgeRock

FZG360 Network Co.

Ltd

GameCredits

Gem

Gibraltar Stock

Exchange (GSX)

Global Blockchain

Technologies

Global Peersafe

Technology Corp.

GM Financial

Guardtime

GXChain

Hangzhou Fuzamei

Technology

Hashed Health

Huawei

Hyperchain

Technologies

Information Builders

Inspur

IntellectEU

Kaiser Permanente

Korea Exchange

Korea Security  
Depository

Koscom

KrypC Corp

LedgerDomain

Libra

Loyal Corporation

Lykke

MadHive

Majid Al Futtaim

Medicalchain

MetaX

MIRACL UK Limited

Monax Industries Limited

MonetaGo

Moscow Exchange

Murphy & McGonigle, P.C.

National Stock

Exchange

of India

New H3C Technologies

NEX Optimisation

Nexiot

Norbloc

NTT Data

Oracle

Orange Magic Cube

Patientory

Paxos

PetroBloq

PDX Technologies

Pravici

PwC

R3

RadarWin Cyber Technology

Red Hat

Revelry Labs

Robert Bosch

Samsung SDS

Sberbank

ScanTrust SA

Schroder Investment Management Limited

SecureKey

Technologies

~sedna GmbH

Sempre IT

Shanghai Ginkoo

Financial Technology Shanghai Onechain

Information

Technology

Shenzhen Forms

Syntron

Information

Shenzhen

Sinolending Ltd

Smart Block Laboratory

Smartchains

Smart Link Lab

Soramitsu

State Street

SWIFT

## Associate Members

Accord Project

Bank of England

Blockchain Research Institute

Chamber of Digital Commerce

Cloud Security Alliance

Colored Coins

IC3

IFSE

Infrachain

Mercy Corps

Ministry of Finance of the Republic of Lithuania

Nxt Foundation

OSCRE

Peking University

Produce Marketing Association

ShareIT.io

Sovrin Foundation

Smart Dubai Office

Taiwan Fintech Association

Tecnalia Research & Innovation

TNO

University College London

University of Luxembourg

University of Rome Tor Vergata/Inuit Foundation

Vilnius Gediminas Technical University

VSP Global

Zhejiang University

The Illinois Blockchain Initiative

# 大中华区超过50个会员

高级会员:



常规会员:





# Global Industry Use Cases

Distributed ledgers for different use cases can have vastly different requirements.

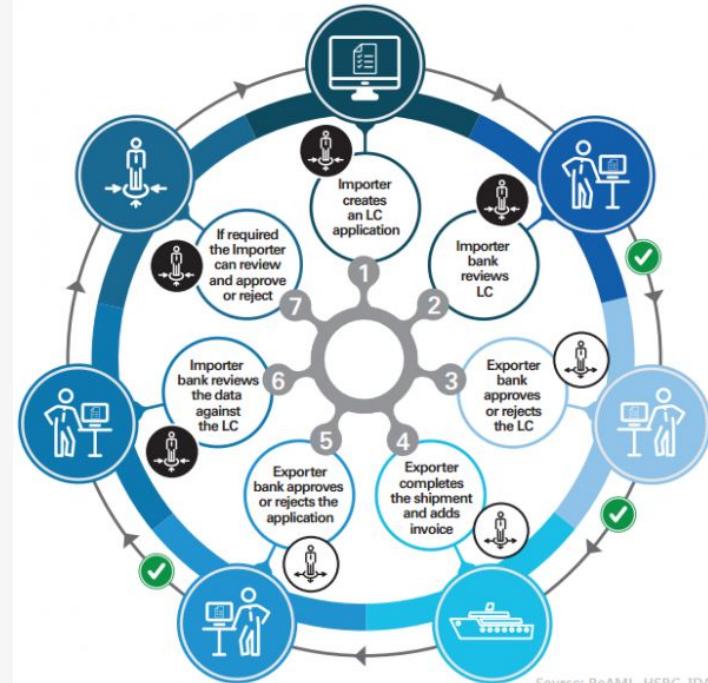
Hyperledger embraces the full spectrum of industry use cases, especially enterprise scenarios with widely varied requirements for decentralization, trust, continuity and confirmation times. Each represents a potentially unique optimization point for the technology.

Here you will find industry-specific use cases to illustrate how blockchain technologies can and are being implemented to bring more trust across industries and the world today.

# Digital Trade Chain



The Challenge	Complex trade value chain with multiple parties (buyers, sellers, custom, shipping, banks) all having their own documentation. Prone to errors.
The Collaboration	A consortium of major world banks including: Deutsche Bank, HSBC, KBC, Natixis, Rabobank, Société Générale, Santander, UniCredit and Nordea
The Technology	we.trade is a blockchain-based international trading system using Hyperledger Fabric that enables accurate trading posture information, order to settlement control, risk coverage, track and trace options



Source: BoAML, HSBC, IDA

# Supply Chain Finance

## The Challenge

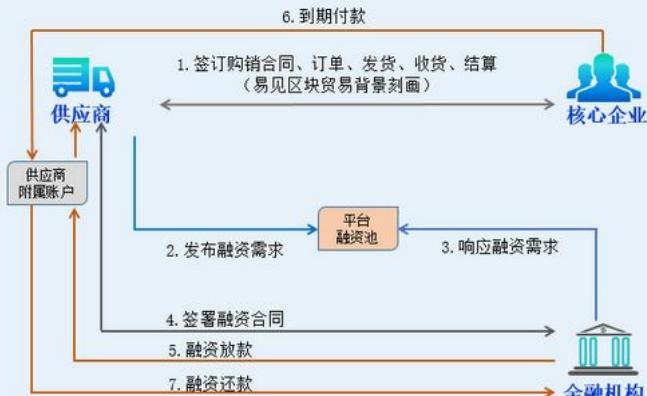
In the traditional supply chain finance business, logistics, capital flow, and information flow are not traceable, visual, and transparent, which makes it difficult for SMEs to raise financing and, and high-quality assets. On top of that companies in supply chain are also trapped by the increase in operating costs caused by inefficient turnaround and high financing costs.

## The Collaboration

There are 161 active users online on the E-visible block online, of which 155 are corporate registered users, including pharmaceutical, bulk, chemical, manufacturing, logistics, and real estate industries, and 6 financial institutions.

## Technology

E-Visible used Hyperledger Fabric as the blockchain platform along with IOT device to enable financial institutions to identify high-quality assets. This enables financial institutions to provide credit services and structured financing related services to help companies obtain better and more options in financial services offerings.



易见区块系统



# Join Us!

Ensure the strength and longevity of a core technology to your business.

Publicly proclaim your leadership in the blockchain space.

Work with other blockchain leaders to develop and promote Hyperledger blockchain for business technologies.

Visit [hyperledger.org/about/join](http://hyperledger.org/about/join)  
or email [jgordon@linux.com](mailto:jgordon@linux.com).



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## Blockchain in Action

Candace Chan - Blockchain Industry Solution  
Manager, IBM Hong Kong

Diverse clients engagement either for specific enterprises, within industries and across industries.....

IBM Blockchain

Trade Finance	Pre and Post Trade	Complex Risk Coverage	Commercial Real Estate
   	   	 	
<b>Identity/ Know your customer (KYC)</b>   	<b>Unlisted Securities / Private Equity Funds</b>    	<b>Loyalty Program Mgt.</b> 	<b>Distributed Energy &amp; Grid Mgt.</b>   
<b>Medical Health Data Exchange</b> 	<b>Anti-Fraud &amp; Port Mgt.</b>  	<b>Carbon Credit Mgt.</b> 	<b>Asset Tracking</b> 
<b>Supply Chain &amp; Logistics</b>  	<b>Food Safety</b>        	<b>Audit</b> 	<b>Digital Rights &amp; Copyright Mgt.</b>  
	     		

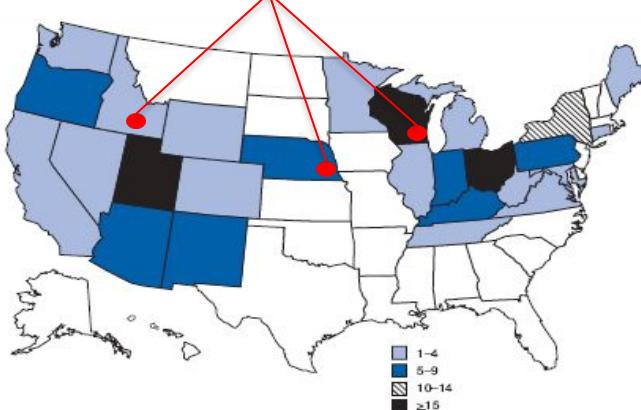
# 2016 E. Coli Outbreak in Spinach

IBM Blockchain

## Nationwide Impact

199 cases reported across **26 states**, 31 hemolytic uremic syndrome (HUS) incidents

3 deaths



## Country-wide Recall



## Impact on the Industry

6-7 years to recover...

...due to **1 supplier, 1 lot, 1 day**

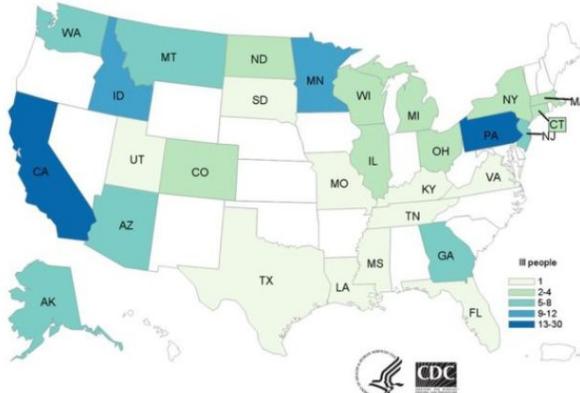
# 2018 E. Coli Outbreak in Romaine Lettuce

IBM Blockchain

## Nationwide Impact

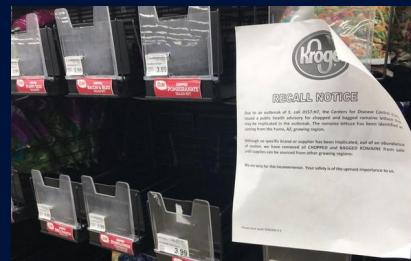
Early April, the U.S. Centers for Disease Control and Prevention raised the red flag on [an E. coli outbreak](#) that had affected 17 people across 7 states

As of May 17<sup>th</sup>, 172 cases across 32 states have fallen ill, of 46 people hospitalised, including 10 people who developed kidney failure.



## Country-wide Recall

“Do not eat or buy romaine lettuce unless you can confirm it is not from the Yuma, Arizona, growing region”



## Impact on the Industry

Another 6-7 years to recover...

# IBM Food Trust solution: started with a Walmart mango pilot

IBM Blockchain

## Pilot Test Case

How long does it take to trace a package of sliced mangoes back to the farm?



## Supply Chain



## Results

Typical manual, mixed digital and paper-based method

6 days  
18 hours  
26 minutes

IBM Food Trust digital solution

2.2 seconds

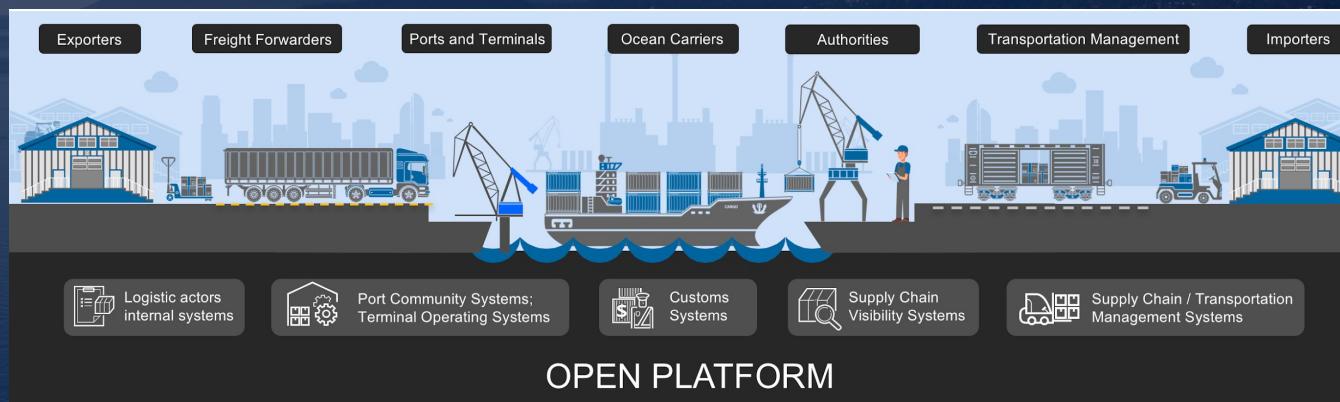
# Maersk and IBM – Global Trade Digitization Platform (GTD)

## What?

Providing Trusted, Tamper-Proof, Cross-Border Workflows for Digitized Trade Documents.

## How?

Providing Shared Visibility and Shared State for Container Shipments

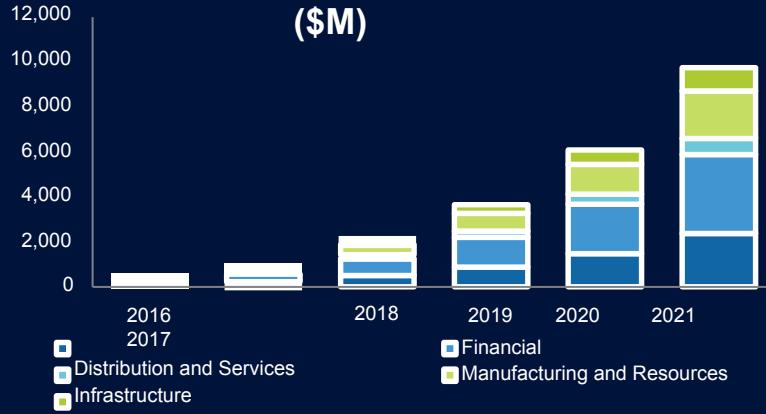


## Benefits

- Increase speed and transparency for cross border transactions through real time access to container events.
- Reduced cost and increased efficiency through paperless trade

# Worldwide Blockchain Opportunity by Sector, 2016-2021

## 2016–2021 Blockchain Sector Spending (\$M)



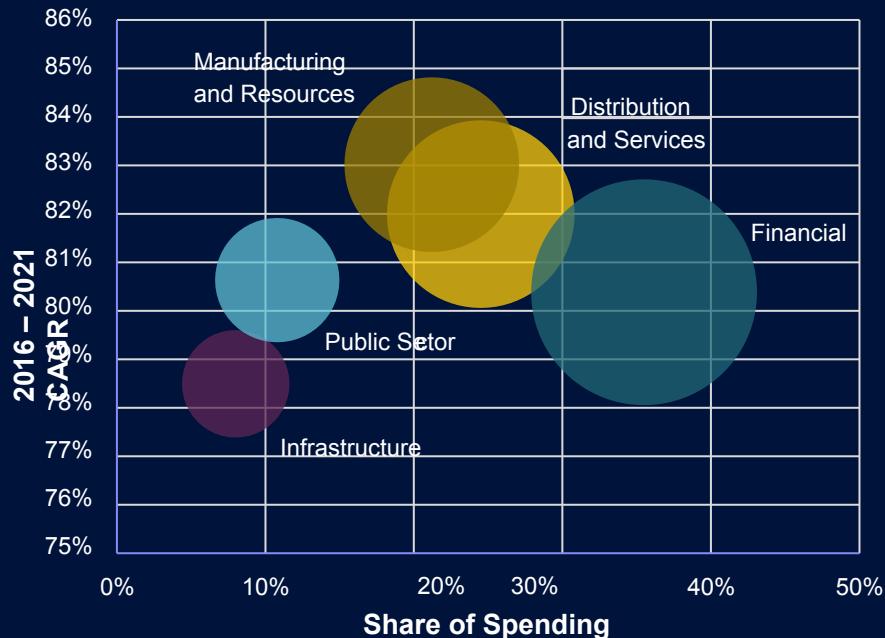
### Sector Growth Rate

- ▲ Public Sector CAGR ~81%
- ▲ Manufacturing and Resources CAGR ~83%
- ▲ Infrastructure CAGR ~79%
- ▲ Financial CAGR ~80%
- ▲ Distribution and Services CAGR ~82%

### Total Market CAGR

~ 81%

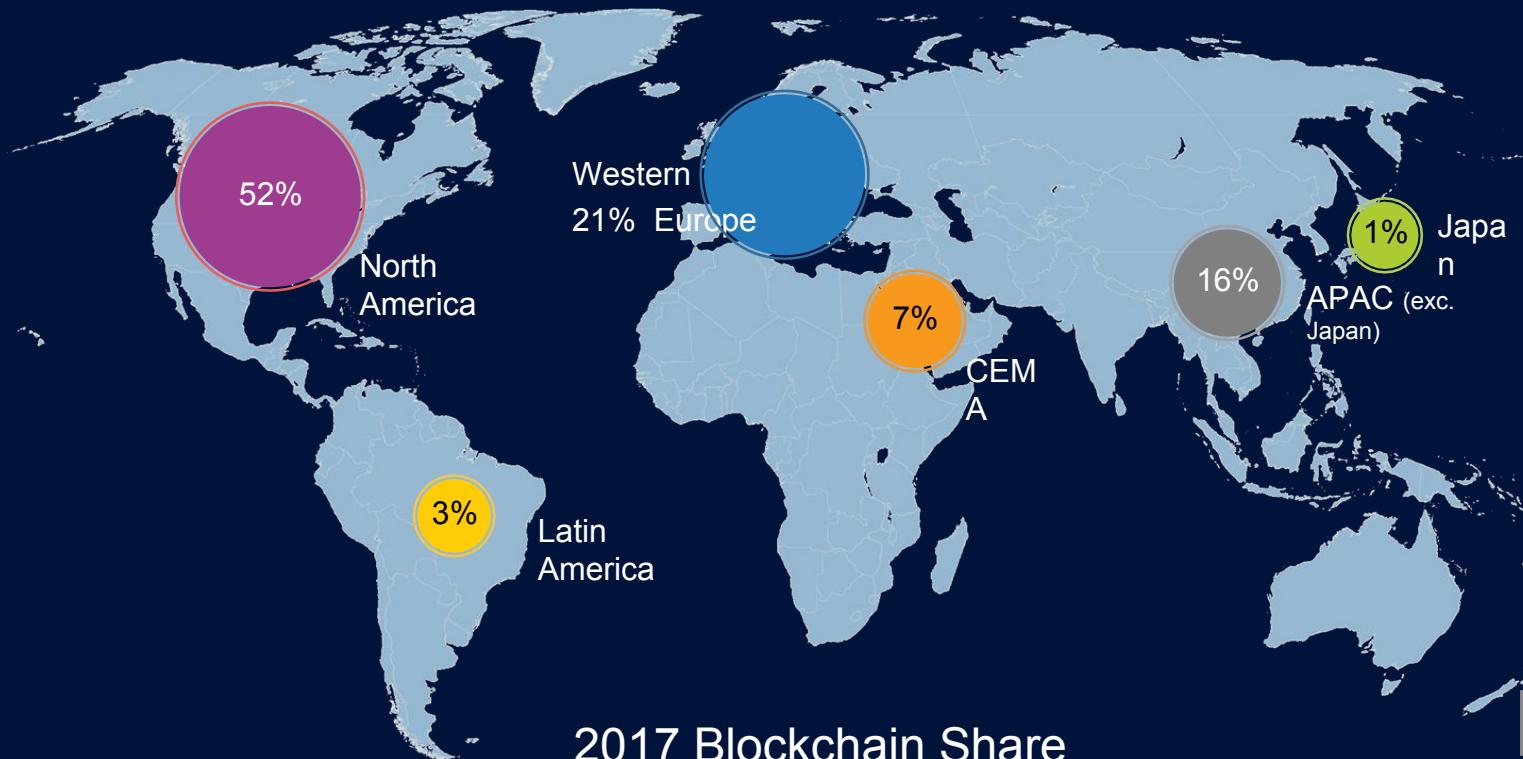
## Size of Bubble Represents Size of Opportunity in \$M in 2017



Source: IDC's Worldwide Semiannual Blockchain Spending Guide, 2017H1 (January 2018)

Note: All numbers in the document may not be exact due to rounding.

\$945 Million spent in 2017 - Blockchain Technology  
Investments Will Reach \$9.7 Billion by 2021



Source: IDC's Worldwide Semiannual Blockchain Spending Guide, 2017H1 (January 2018)  
Note: All numbers in the document may not be exact due to rounding.

# Networks are growing to form industry platform....



- A: North America
- B: Europe
- C: Asia Pacific
- D: Cross Border

© 2017 IBM Corporation



**B:**



**C:**



**D: Cross region**





# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

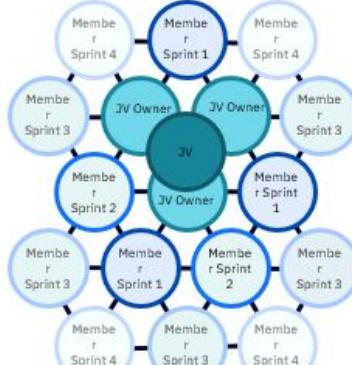
## Adoption Model & Business Model

# 4 Types of Business Model



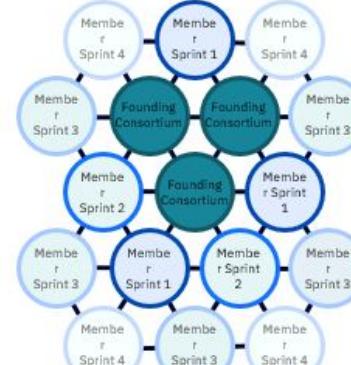
## Founder-Led Network

*Single company driving the initial project then others join the network*



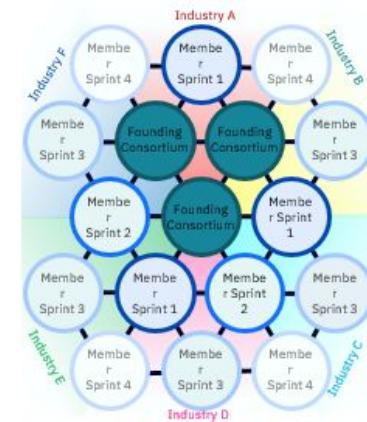
## Joint Venture Network

*2 or more create a JV to govern the initial network*



## Consortium Network

*A consortium as the initial network and governance in a specific industry or sector*



## Business Ecosystem

*A Consortium Network working across multiple industries or sectors*

Future members join each initial network type in similar ways to grow the network

# Characteristics of Each Model

## OPTION 1

### Founder Led / JV

Business Differentiation  
Value to me  
and my network

- Founder is in control & invite partners to join
- Normally a private network which drives innovation
- Collaborate with non competitors to enhance today's differentiating processes B2B-2C
- Founder has to consider the economics incentive for participants



## OPTION 2

### Consortium

Market Utility  
Solve an industry problem  
Collaborate not compete

- Collaborate with competitors to consume (buy) or build market utilities to optimise shared B2B processes
- Aim to optimize:
  - B2B cost
  - Risk
  - Capital
- Members can influence the solution and share risks at the same time
- Costs are pooled across members probably via a "joining fee"



## OPTION 3

### New Market Model

Continuously combine existing solutions with differentiation

- Collaborate with partnering industries to innovate on new value propositions, digital platforms and marketplaces
- Challenges:
  - Inter-operability
  - Governance between the 2 networks





# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

# About Hyperledger Fabric

David Liu - Contributor of Hyperledger Fabric

Introduction

# About me

- How do I know fabric
- Which part do I contribute  
fabric-sdk-node
- Daily work
  - Dynamic cluster of fabric

# What is fabric



- Federated Blockchain
- “a platform”
  - not designed as a plugin, a service but as a foundation, infrastructure
- “for distributed ledger”
  - not crypto-currency or token based
- “solutions”
  - flexible to implement enterprise application

# Features

- Pluggable
  - ledger state database: leveldb, couchdb
  - Consensus: kafka
- Security beside TLS
  - Channel isolation
  - Fabric User is privkey-cert based identity
  - 2-phase commit in chaincode invoke process
  - Configurable policy in channel: Readers, Writers, Admins
  - Configurable Endorsement policy in chaincode instantiate

TLS: Transport Layer Security, cryptographic protocols over computer network

“.....Anyway,  
what can Fabric help on our business?”

....Then, what can

NoSQL DB, Redis

IBM Websphere, Apache server

HTTPS

help your business?



# HYPERLEDGER

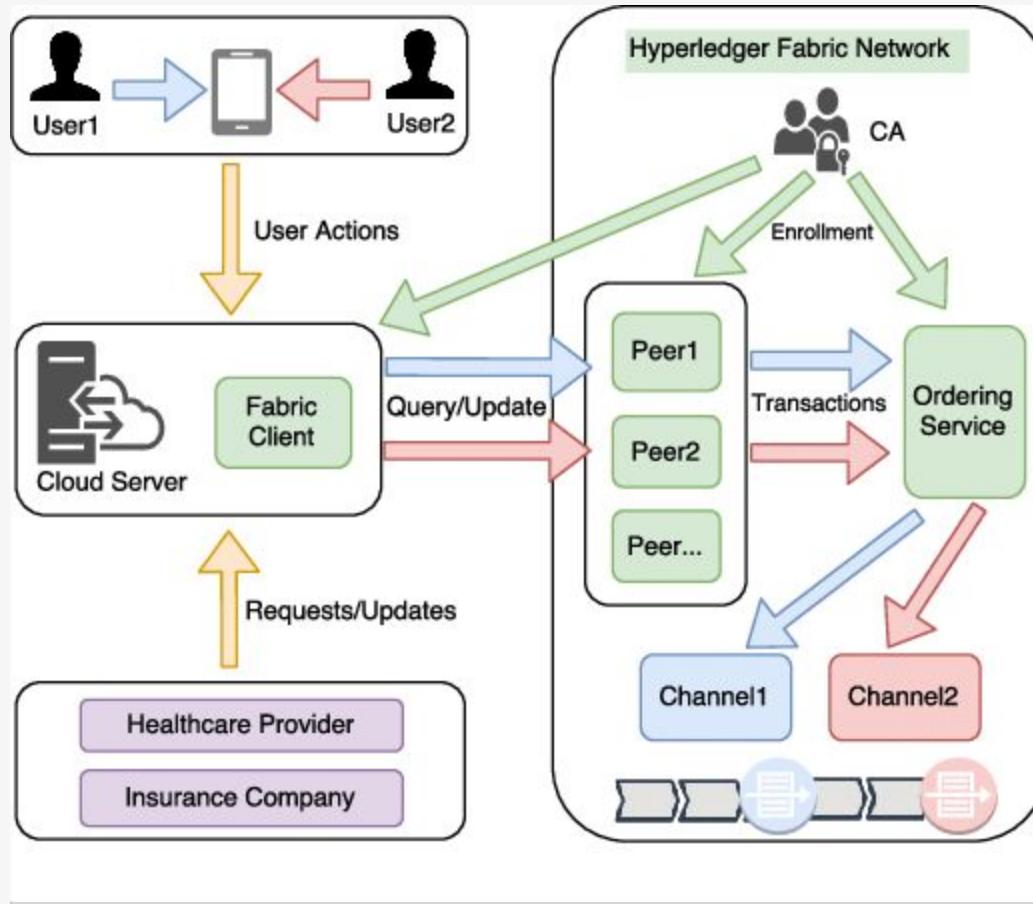
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## About Hyperledger Fabric

Architecture

# Architecture

- Concepts
  - reference: *Hyperledger Fabric v1.0 Deep Dive*, Binh Nguyen, IBM
- [Source repositories & docker image](#)



# Orderer

A node, a docker container

“The genesis of a network”

- Act as miner in bitcoin, Accept transactions and generate, deliver blocks

Clients of ordering service are peers

Orderer manages a pluggable trust engine: kafka, solo

# Peer

## A node

- maintaining state of ledger (storage)
- managing chaincodes
- provide event hub, broadcast event to listeners (block event, transaction event)

A Peer can be an endorser or commiter. An endorser is always a committer

- An endorser executes (chaincode literal) and endorses transactions → 1 phase
- A committer verifies endorsements and validates transaction results → 2 phase

# Chaincode, ledger

## Chaincode: Smart contract of Fabric

- A program, handles business logic agreed to by members
- runs in a secured Docker container isolated from the endorsing peer process.
- API to ledger
  - initializes and manages ledger state through transactions submitted by invocation.

Cross-chaincode invoke is allowed within the same channel, under appropriate permission setting

# Channel ----A data partitioning mechanism

“Chat room”

- message shared (ledger sync)
- Invitation only, only peer under allowed organization (MSP) can joined
- Channel update
  - configuration managed on-chain
  - via special transaction included in “config block”

problem: [FAB-5953] implement leave channel ---- Yacov Manevich

# Overview of life-cycle

1. Channel create
2. Peer join channel
3. Chaincode installed to peer
4. Chaincode instantiate
  - o chaincode can be instantiated on multiple channels, each instance is isolated within its channel

Then

Chaincode invoke/query

Others queries



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

# About Hyperledger Fabric

Architecture

Source repositories & docker image

# Source code in gerrit

- Gerrit provides web based **code review** and repository management for the **Git version control system**.
- As plugin to Git
- Remind: Pull request, issues in github mirror will be ignored

The screenshot shows a Gerrit interface for the Hyperledger Project. The top navigation bar includes links for Account signup / management, JIRA, Jenkins, Nexus, Wiki, Mailing lists, and Sign-off R. The main header displays "HYPERLEDGER PROJECT". Below the header, there are tabs for All, My, Projects, People, Documentation, Changes, Drafts, Draft Comments, Edits, Watched Changes, Starred Changes, and Groups. A search bar and a "Changes ▾" dropdown are also present.

The central area shows a specific code review titled "Change 20099 - Needs Code-Review Label". The commit message is: "FAIR-10257 NodeSDK - update packages". The commit details include the author (wenjian3 <wenjianq@gmail.com>), committer (wenjian3 <wenjianq@gmail.com>), date (May 22, 2018 8:48 AM), and time (May 23, 2018 1:33 AM). The commit hash is 9d62ed9c1c4e61d0773194ea46bc20892409bed. The patch set ID is 1007dcdf2b037ccae36542e56038bf3d3d6b57767b. The file path listed is ".gitignore".

On the right side of the commit details, there is a sidebar with "Owner: Wenjian Qiao", "Assignee: ", "Reviewers: Brett Harrison, Hyperledger Jobbuilder, david liu", "Project: fabric-sdk-node", "Branch: master", "Topic: ", "Strategy: Merge if Necessary", "Updated: 7 minutes ago", and buttons for "Cherry Pick" and "Follow-Up".

Below the commit details, a "Code-Review -1" status is shown for "david liu" with a note "+1 Hyperledger.Jobbuilder".

The bottom section displays a table of files with their paths, comments, and sizes. The table has columns for File Path, Comments, and Size. The data is as follows:

File Path	Comments	Size
.gitignore	3	10677
fabric-ca-client/lib/FabricCAClientImpl.js	6	
fabric-ca-client/package.json	10	
fabric-client/package.json	16	
A package-lock.json	28	
package.json	28	
test/integration/channel-event-hub.js	14	
test/integration/client.js	14	
test/integration/cloudant-fabricca-tests.js	14	
test/integration/configurator.js	14	
test/integration/couchdb-fabricca-tests.js	14	
test/integration/couchdb-util.js	12	
test/integration/create-configtx-channel.js	14	
test/integration/e2e/create-channel.js	2	
test/integration/e2e/e2eUtil.js	14	
test/integration/e2e/install-chaincode-fail.js	14	
test/integration/e2e/install-chaincode.js	14	
test/integration/e2e/instantiate-chaincode.js	14	
test/integration/e2e/Invoke-transaction.js	14	

# Source(gerrit) -- core

- fabric: core written in golang
  - source of docker image ‘fabric-peer’, ‘fabric-orderer’, ‘fabric-ccenv’
  - default golang chaincode
- fabric-baseimage: internal only
  - “This is only intended for release managers curating the base images on docker-hub.”
  - holding DockerFile of *thirdParty* images
    - fabric-kafka, fabric-zookeeper,fabric-couchdb
  - use another set of versioning: 0.4.8 newest
    - same apply to *thirdParty* images

Branch: master ▾			fabric-baseimage / images /
	sanchez1	[FAB-8021] sync with official couchdb image	...
			..
	couchdb		[FAB-8021]
	kafka		[FAB-774]
	packer-overview.graffle		FAB-45
	zookeeper		[FAB-764]
	packer-overview.png		[FAB-103]

# Source(gerrit) -- pluggable & implements

- fabric-ca: written in golang
  - implement of MSP(Member Service Provider)
- fabric-chaincode-\* : java, node, evm
- fabric-amcl: internal only
  - This repository helps to hold and handle the amcl package(s) we need in migrating 3rd party golang dependency management tools of hyperledger/fabric, from [govendor](#) to [dep](#).

# Source(gerrit) -- Helpers

- fabric-chaintool
  - command line client tool
  - most known as docker image of 'cli peer/container'
- fabric-sdk-\*: go, java, node, py
- fabric-sdk-rest:
  - latest support: 1.0.2
- IBM-Blockchain/fabric-spring-sdk(github)
  - New from Feb 2018
  - 'Spring' for Java Spring MVC

# Projects

- blockchain-explorer
- cello family
  - analytics, k8s-operator
- fabric-samples:
  - balance-transfer
  - basic-network
- IBM-Blockchain/marbles(github)
- Composer(github)

# Where should we start??

Projects	core ver. in use	consensus	crypto material	dependency	UI	remark
blockchain-explorer	(latest)	solo	cryptogen	byfn	Y	
cello	1.0.x	kafka	cryptogen CA	system environment	Y	using NFS
balance-transfe r	(latest)	solo	cryptogen CA	none	N	e2e web service
marbles	1.1.0- preview	solo	cryptogen CA	fabcar(basic- network)	Y	
Composer	-	-	-	(a lot outside fabric)	Y	awesome online version

# Learning fabric

- docker: time to be devOps
- golang
  - for golang-chaincode develop
  - for fabric source code inspect
- 1 month focus at least
  - Hundreds of configurable options
    - docker env, config file, command option
  - ‘silent’ TLS, domain or handshake error.
  - Coupled design
  - time consumed in building chaincode containers
    - **Testing Using dev mode**
    - <http://hyperledger-fabric.readthedocs.io/en/release-1.1/chaincode4ade.html#testing-using-dev-mode>

# PS: node-sdk is preferred

- Most tags to match fabric core version
- node-sdk-based projects:
  - Caliper: fabric benchmark test
  - [fabric-sample/balance-transfer](#):
    - e2e sample web app
  - [IBM-Blockchain/marbles](#)
  - fabric-sdk-rest

v1.1.0
v1.1.0-preview
v1.1.0-alpha
v1.0.5
v1.0.4
v1.0.3
v1.0.2
v1.0.1
v1.0.0
v1.0.0-rc1
v1.0.0-beta

The screenshot shows two main components of the United Marbles application:

- AUDIT** (Left): A modal window showing details of a transaction (TX 2).
  - Marble ID: m01496411887729kYGcJ
  - Owner: alice
  - Company: United Marbles
  - Owner Id: o0149641891746752yys
- United Marbles** (Right): The main interface.
  - Setup** and **Settings** buttons.
  - Search** bar.
  - Marble Count**: 3/3
  - Marble Distribution**:
    - Amy: 1 green marble, 2 red marbles.
    - Alice: 1 green marble, 2 red marbles.
    - Ava: 3 red marbles.
  - Actions**: Drag and drop marble to remove.

# Node-sdk framework stack

- build-tool: Gulp
- code-style: Eslint (ES6), typejs (optional)
- Test: npm tape
- build target: npm fabric-client & npm fabric-ca-client
  - Copy code base from fabric-client to fabric-ca-client (gulp watch)
- Config loader: npm nconf
- logger: winstonjs



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## Before to production

Topic 1

# Before to production

- docker-compose -> docker api
- docker orchestration: k8s or swarm
- cryptogen -> fabric-ca
  - <https://github.com/hyperledger/fabric-samples/tree/master/fabric-ca>
- File sharing of ‘artifacts’
- Channel update and signature service



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## TPS and Caliper

Topic 2

# Hyperledger/Caliper

- a blockchain benchmark tool
- report performance indicators, such as TPS (Transactions Per Second), transaction latency, resource utilisation...
- Not limited to Fabric, also applicable to Sawtooth, Iroha

# HTML report

Caliper Report											
round 2 - open											
performance metrics											
Name	Succ	Fail	Send Rate	Max Latency	Min Latency	Avg Latency	75%ile Latency	Throughput			
open	6000	0	710 tps	25.27 s	4.46 s	17.17 s	19.54 s	212 tps			
resource consumption											
TYPE	NAME		Memory(max)	Memory(avg)	CPU(max)	CPU(avg)	Traffic In	Traffic Out			
Process	node local-client.js(avg)		181.1MB	173.3MB	45.21%	20.92%	-	-			
Docker	dev-peer1.org2.example.co...le-v0		18.0MB	16.3MB	9.19%	0.75%	1.9MB	750.5KB			
Docker	dev-peer1.org1.example.co...le-v0		43.4MB	35.7MB	15.75%	3.83%	7.3MB	2.8MB			
Docker	dev-peer0.org2.example.co...le-v0		40.1MB	32.6MB	15.79%	3.96%	7.3MB	2.9MB			
Docker	dev-peer0.org1.example.co...le-v0		9.9MB	9.9MB	7.17%	1.09%	1.7MB	708.4KB			
Docker	peer1.org1.example.com		516.8MB	453.4MB	64.55%	47.87%	30.4MB	76.6MB			
Docker	peer1.org2.example.com		480.1MB	418.0MB	65.91%	29.35%	24.2MB	20.3MB			
Docker	peer0.org1.example.com		475.9MB	409.9MB	56.51%	28.05%	24.0MB	23.6MB			
Docker	peer0.org2.example.com		502.6MB	449.5MB	70.62%	45.94%	30.3MB	13.2MB			
Docker	ca_peerOrg1		6.2MB	6.2MB	0.06%	0.00%	729B	0B			
Docker	orderer.example.com		242.1MB	222.3MB	31.98%	11.01%	22.9MB	89.2MB			
Docker	couchdb		93.1MB	93.1MB	2.47%	0.94%	659B	0B			
Docker	ca_peerOrg2		6.4MB	6.4MB	0.44%	0.03%	729B	0B			
round 3 - query											
performance metrics											

# Report 1

Memory: 16G, CPU: i5-6500 3.20GHz \*4, no SSD

5 node process

**Invoke** : up to 200 TPS

- txNumber:6000 /each round
- Rate: 800 pushes

**Query**: up to 600 TPS

- txNumber:6000 /each round
- Rate: 1000 pushes

# Report 2

		Ubuntu 16.04 VM on Win10 host				Ubuntu 16.04	
i7-5960X @3.0GHz		2 cores	4 cores	8 cores	8 cores with 16 threads		
Invoke		VM image on HDD				VM image on SSD	Installed on SSD
Invoke		127 tps	249 tps	416 tps	441 tps	431 tps	433 tps
Invoke		156 tps	313 tps	600 tps	556 tps	544 tps	652 tps



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## How to contribute

Source contribution

# Steps

[Advertisement] My Chinese blog post <https://davidkhala.github.io/2017/08/30/fabric-contribute/>

- install Gerrit, prepare Linux Foundation ID, create patch and collaboration

Want help?

1. Rocket chat: the best channel to get instant reply.
2. Hyperledger Email list
3. JIRA issue

# Active People

Christopher Ferris <[chris.ferris@gmail.com](mailto:chris.ferris@gmail.com)> - Chair of the Hyperledger TSC,Distinguished Engineer

David Enyeart <[enyeart@us.ibm.com](mailto:enyeart@us.ibm.com)> - another 1.1 release managers, maintainer

Gari Singh <[garis@us.ibm.com](mailto:garis@us.ibm.com)> - Blockchain CTO

*sdk-node active main maintainers*

Bret Harrison <[beharrison@nc.rr.com](mailto:beharrison@nc.rr.com)> ;Chaoyi zhao<[zhaochy\\_2015@hotmail.com](mailto:zhaochy_2015@hotmail.com)>

sdk-python

Baohua Yang <[yangbaohua@gmail.com](mailto:yangbaohua@gmail.com)> <https://github.com/yeasy> author of <区块链原理、设计与应用>

# Future

[FAB-9081] Client Identity Chaincode Library not provided by Docker image

[FAB-7083] Use dep for dependency management (govendor, fabric-amcl)

## tCert

- Currently, enrollment credential (cert,private key pair, 'eCert') is used to sign transactions...There was a plan to use TCerts but an effort is ongoing to use Idemix zero knowledge proofs...

Document about build docker image natively

[FAB-1151] SideDB: private ledger data configured in chaincode instantiate

## Service Discovery

# Problems

- create channel event callback / event hub on orderer
- channel name in configtxgen tolerant uppercase char
- fabric-ca-server with node-sdk:
  - set --csr.cn in CLI/docker ENV, required clean up ca-key.pem and ca-cert.pem manually
  - TLSCA cannot be access from node-sdk with self-generated tls-cert.pem, but ca-cert.pem
- stateful node-sdk Client Object (state of User, signing identity)
- orderer address and organization is globally configured in ‘testchainid’
  - Jason yellick



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## Q & A

# Ready to Discuss Membership Further?

If you are interested in exploring corporate membership, please complete the inquiry form on the Hyperledger Join page.

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



WHO WILL YOU  
**TRUST**  
WITH YOUR TRUST NETWORK?



**HYPERLEDGER**  
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS



# HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

## Appendix

# docker-compose

Definition: a tool for defining and running multi-container Docker applications.

- You use a YAML file to configure your application's services.

YAML is a DSL, like a combination of json and XML

- Widely used in Fabric as config file format (cryptogen, configtxgen, and even connection profile)

**sample:** <https://github.com/hyperledger/fabric-samples/blob/release-1.1/balance-transfer/artifacts/docker-compose.yaml>

# docker api

Restful

Choose your preferred language

build http request to it

get rid of bash programming

<https://docs.docker.com/engine/api/v1.35/>

ps: This <https://docs.docker.com/engine/api/v1.24/> has higher rank in Google search but not the latest  
latest is 1.37 but too new to fabric

# Cryptogen -> fabric-ca

## Cryptogen

- command line tool
- based on pre-configured yaml
- Simple user section: user0, user1

```
# -----
# "Users"
# -----
# Count: The number of user accounts _in addition_ to Admin
# -----
Users:
  Count: 1
```

# Cryptogen -> fabric-ca

## fabric-ca

- docker image provided
- Rich user identity certificates management
  - password, revoke list
- Intermediate CA, ca-chain

# fabric-ca: how to use

CA in most demo ---- mixture usage with cryptogen

- <https://github.com/hyperledger/fabric-samples/blob/release/balance-transfer/artifacts/docker-compose.yaml>

ca.org2.example.com:

environment:

- FABRIC\_CA\_HOME=/etc/hyperledger/fabric-ca-server
- FABRIC\_CA\_SERVER\_CA\_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org2.example.com-cert.pem
- 

FABRIC\_CA\_SERVER\_CA\_KEYFILE=/etc/hyperledger/fabric-ca-server-config/a7d47efa46a6ba07730c850fed2c1375df27360d7227f48cdc2f80e50  
5678005\_sk

command: sh -c 'fabric-ca-server start -b admin:adminpw -d'

volumes:

- ./channel/crypto-config/peerOrganizations/org2.example.com/ca:/etc/hyperledger/fabric-ca-server-config

# fabric-ca mixture mode

<https://jimthematrix.github.io/tutorial-app-dev-env-setup.html>

Also related to identities, you should make a decision on whether Fabric-CA should be part of your solution. This is a server with REST APIs that supports dynamic identity management with registration, enrollment (getting certificates), revocation and re-enrollment. So it is very useful in providing user identities on the fly. But note that user identities provisioned this way are only of the **MEMBER** role in the fabric, which means it won't be able to perform certain operations reserved for the **ADMIN** role:

- create/update channel
- install/instantiate chaincode
- query installed/instantiated chaincodes

For these privileged operations, the client must use an ADMIN user to submit the request

---- Jim Zhang, main maintainer of node-sdk (inactive now)

Create User for invoke/query chaincode only

# How to get ADMIN user

- Cryptogen: default user ‘Admin’
  - default user beside the user count
- No different from MEMBER user in nature, but the place it is recognized.
  - *admins* Certificates (base64 format) is configured on channel
  - in peer, act as the default user in ‘peer ...’ command
    - peer docker env: CORE\_PEER\_MSPCONFIGPATH should include ‘admincerts’

# My solution: Host machines

Leader company:

- machine *master*
- > 1 organizations
- > 1 orderer
- 1 CA, 1 peer for each peer organization, 1 CA, 1 peer for each orderer organization

Member company:

- machine *slave*
- 1 CA, 1 peer, optional orderer for each organization

# My solution: Steps

1. build a docker swarm in *master*, Invite *slave* to join swarm
2. (*master* and *slave*) create CA service
3. (*master* and *slave*) generate crypto material (MSP config) for orderer, peers, users via CA
4. (*master* and *slave*) start orderer, peer services ([file share](#))
5. (*master*) create channel and let peer in *master* join channel
6. (*slave*) delegate *master* to update channel
  - a. (*master*) [collecting signature across machines](#) and propose channel update

# File sharing

- Worst: use copy/paste
- Not bad: use Network File System(NFS)
- My suggestion
  - server design
  - docker volume plugin adapt to distributed file system (Gluster)

Prerequisites to start peer/orderer service

- MSP config for orderer , peer ---- generated by fabric-ca service
- genesis block file for orderer ---- https GET from server

# Channel update

But the new peers, orderers service are still isolated from existing channel

To add new peer/orderer organization to channel

⇒ “channel update”

# Channel update: requirements

- (*master*) ADMIN user identity from one of joined organizations
- (*master*) *update\_proto* binary file for update channel
  - decode/encode via configtxlator
- (*master and slave*) N signatures of update proto binary
  - Signature is signed by ADMIN of each organization
  - $N > \frac{1}{2}$  count of joined organizations
  - (*master*) collecting signatures from both *master* and *slaves*:
- (*slave*) Update content of new organization
  - *admins, root\_certs* certificates,
  - MSP id, MSP name

# My solution ---- Signature services

*master*

- /newOrg
  - Request:
    - *admins, root\_certs* certificates
    - MPSID, MSPName
  - Action:
    - generate *update\_proto*
    - collect signatures by calling API  
/getSignatures for *master* and all  
*slaves* iteratively
    - Channel update
- /getSignatures
  - similar to same API in *slave*

*slave:*

- /getSignatures
  - Request
    - *update\_proto*
  - Response
    - signatures

All in:

<https://github.com/davidkhala/delphi-fabric>