HITACHI
Inspire the Next

Hyperledger Fabric1.0 概要

日立製作所　研究開発グループ
システムイノベーションセンタ
　山田　仁志夫

Global Center for Social Innovation North America,
R&D Division, Hitachi America, Ltd.
　大島　訓

2017/3/16

目次

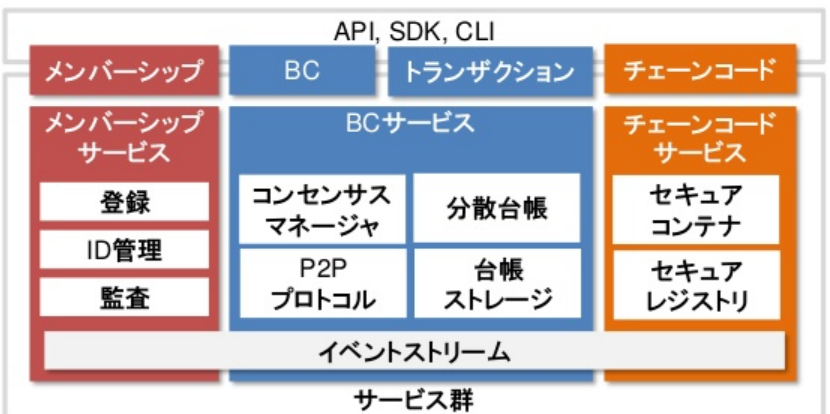
HITACHI
Inspire the Next

Fabric1.0 は、現在開発中であり、設計や実装が日々変更されています。
本資料は、最新でない情報を含む可能性があります。

0．Hyperledger Fabric0.6まで（1）

HITACHI
Inspire the Next

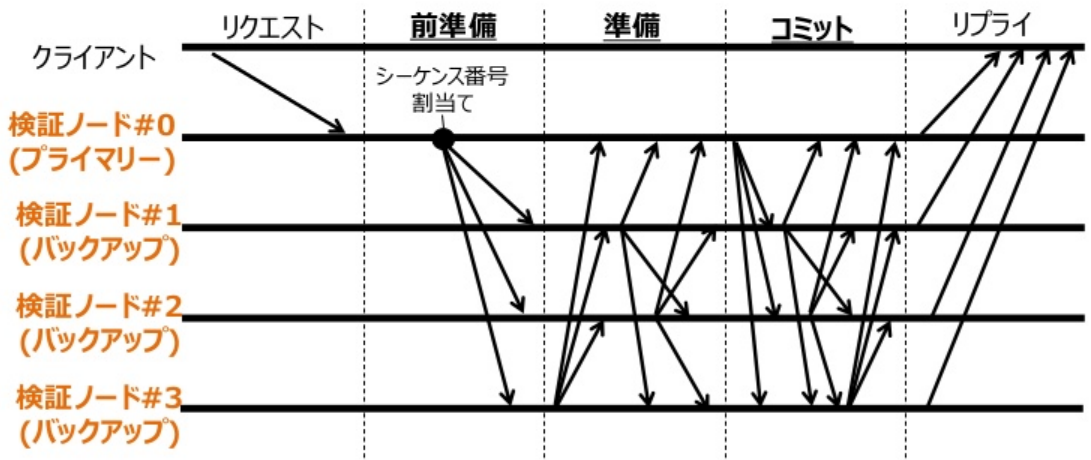
- Hyperledger プロジェクトは，2016年2月から活動開始
- ブロックチェーン基盤
  - Active　　：Fabric(IBM)
  - Incubation　：Iroha(Soramitsu)，Sawtooth Lake(Intel)、CORDA(R３予定)
- Fabric v0.6アーキテクチャ



0．Hyperledger Fabric0.6まで（2）

HITACHI
Inspire the Next

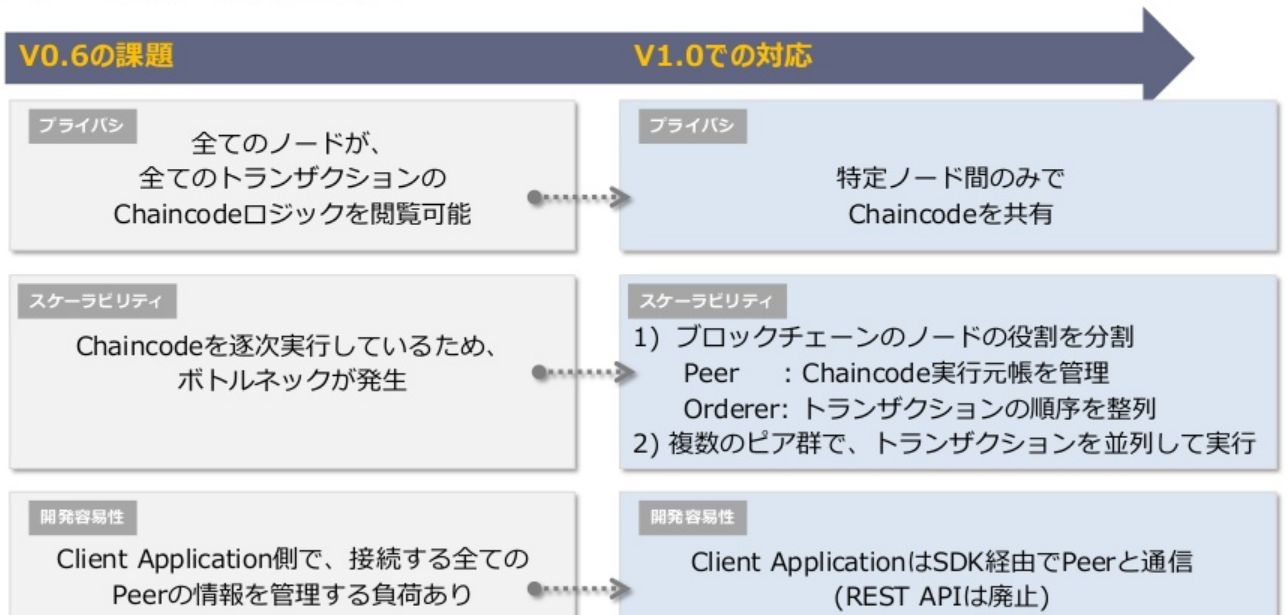
- PBFT：Castro等に提案された分散合意形成アルゴリズム
- 検証ノードの総数n台に対して、(n-1)/3台までのノードが故障/悪意を持ったノードであっても正常に合意形成可能 (最小構成4台)
- 3フェーズコミット（前準備、準備、コミット）
- 対PoW(Proof of Work)：大量の計算処理が不要、取引が確定するといった利点あり



1. Fabric1.0 設計コンセプト

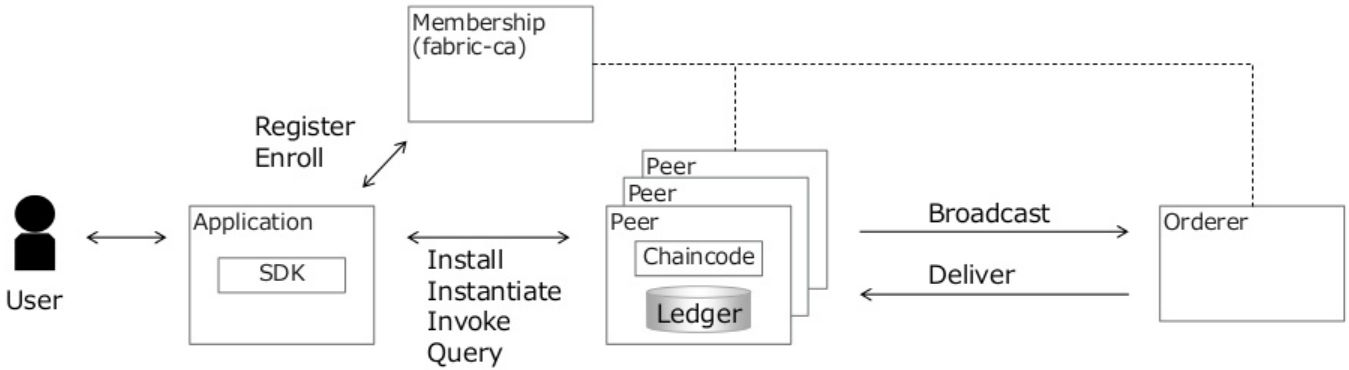
HITACHI
Inspire the Next

- Hyperledger fabric v1.0 は、エンタプライズ領域への適用に向けた以下の3つの課題を満たすべく設計・開発を推進



2. システムアーキテクチャ

HITACHI
Inspire the Next



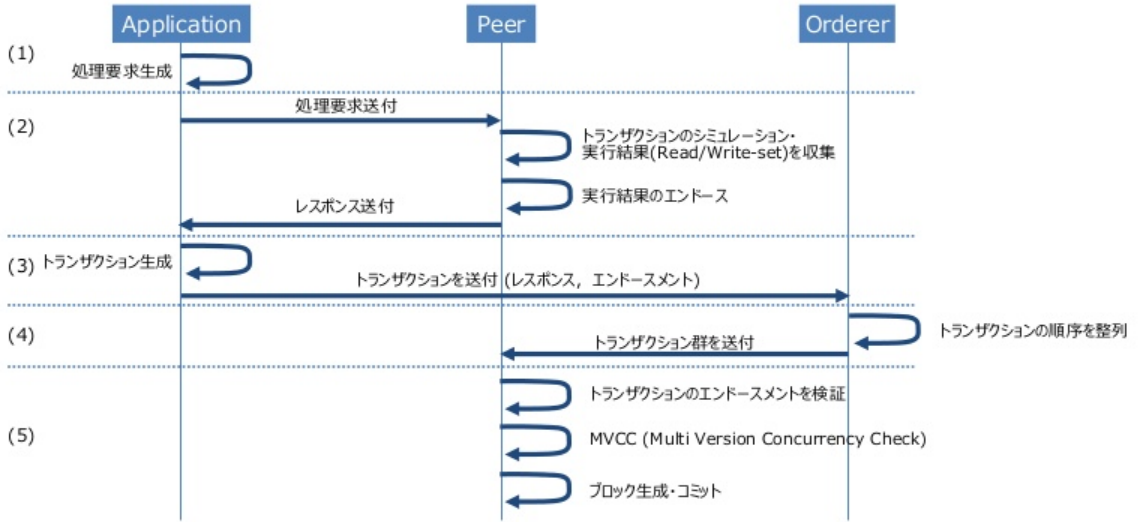
コンポーネント	役割
SDK(データアクセスライブラリ)	トランザクションを要求
Membership (fabric-ca)	Peerのエンロール(登録・承認) ユーザのエンロール(登録・承認)
Peer (Endorser)	トランザクションをシミュレート実行する トランザクションを検証、元帳を管理
Chaincode	トランザクションを実行するスマートコントラクトプログラム
Ledger	データを格納(Blockchain とState)
Orderer	トランザクションの順序を整列

[Figure Source] High level data flows (https://jira.hyperledger.org/browse/FAB-37), FabricNext-DevUpdate.pdf

3. Transaction Flow (Endorser-Orderer Model)

HITACHI
Inspire the Next

- 複数のコンポーネントが連携し合い、トランザクションを処理

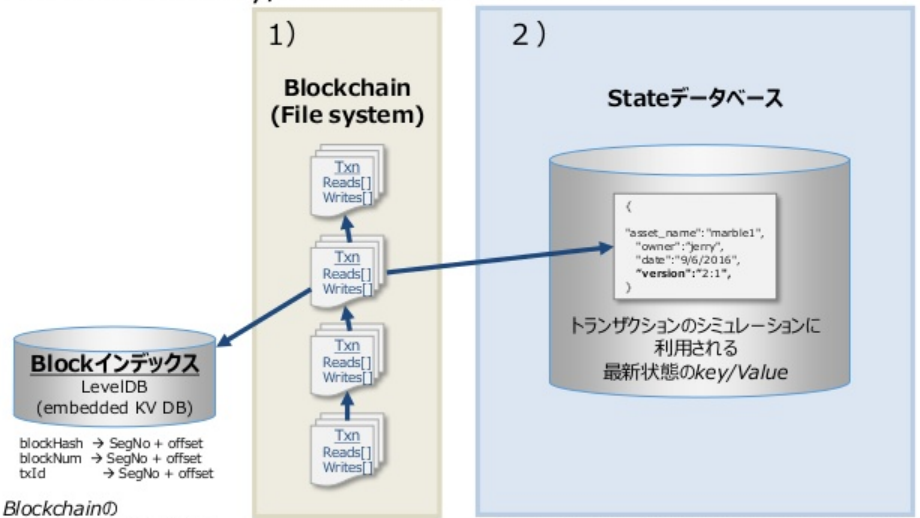


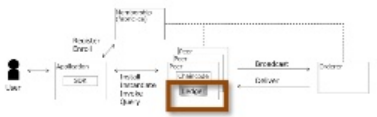
(1) *Application(SDK)* は、1つ以上の*Peer*に、トランザクションの処理要求を送付
(2) 各*Peer* は、Chaincodeをシミュレート, 実行結果に署名 (エンドース) 後、*Application* へ返信
(3) *Application(SDK)* は、複数の*Peer*から受け取ったエンドースメントを集約し、*Orderer* へ送付
(4) *Orderer* は、トランザクション群内でのトランザクションを整列し、各*Peer* にトランザクション群を配信
(5) 各*Peer* は、トランザクション群を検証し、ブロックを生成してLedgerにコミット

4.1 Ledger / Blockchain, Stateデータベース

HITACHI
Inspire the Next

- Ledger は、2種類のデータストアで構成
  1) Blockchain: トランザクションのハッシュチェーン
  2) State Database: key/Valueで格納されるデータのステート





Stateデータベースは以下の2つのオプションから選択可能
- LevelDB (デフォルトの Key/Value DB) ：キー検索クエリをサポート
- CouchDB(オプション) ：全履歴データ検索などリッチクエリをサポート

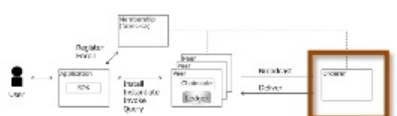
[Figure Source] Ledger (https://jira.hyperledger.org/browse/FAB-758), HyperledgerFabric_LedgerV1_20170308.pptx

4.2 Orderer

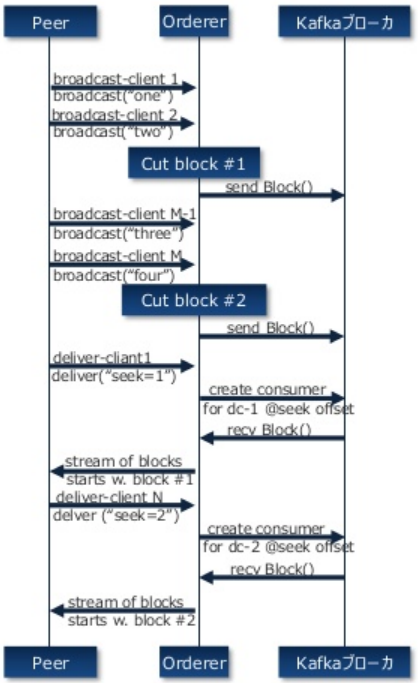
HITACHI
Inspire the Next

- Orderer は、ブロードキャスト型のオーダーリングサービスを提供
- Client Peer は、Ordererに、メッセージを通知し，全てのClient Peerは、同一の整列済みブロックをレスポンスとして受け取る
- Fabric1.0 は、3種類のオーダリングサービスをサポート予定

#	Orderer種類	概要
1	Solo	開発・テスト用のシングル・プロセスサービス
2	Kafka	整列のため、KafkaのPub-subを利用
3	SBFT (Simple BFT)	整列のため、BFTを利用 ※現状はα版レベル



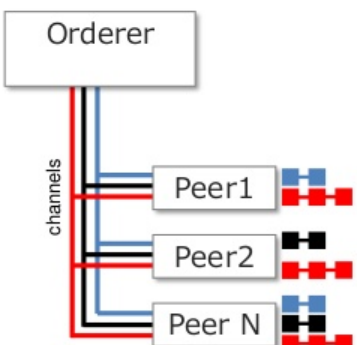
[Figure Source] Prototype ordering service based on Kafka (https://jira.hyperledger.org/browse/FAB-32)



4.3 Multi-Channel/Sub-ledger

HITACHI
Inspire the Next

- Multi-Channel と Sub-ledger により、データを分離し、機密性を確保
- Channelは、データ共有の範囲を定義するために生成
- 1つのChannelは、1つのledgerを保持
- Channelに紐付いたpeerのみChaincodeを実行し、ledgerにアクセス可能
- Ordererは、全てのチャネルの全てのトランザクションを受け取る



#	Channel	Channelに紐付くpeer	Ledger
1	青	Peer1, Peer N	青
2	黒	Peer2, Peer N	黒
3	赤	Peer1, Peer2, Peer N	赤

[Figure Source] Multichannel (https://docs.google.com/document/d/1eRNxxQ0P8yp4Wh__Vi6ddaN_vhN2RQHP-IruHNUwyhc/edit?usp=sharing)

4.4 Chaincode (スマートコントラクト)

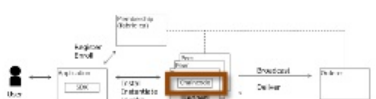
HITACHI
Inspire the Next

- Chaincode は、トランザクションを実行するスマートコントラクトプログラム
- Chaincode は、以下3種類のトランザクションをサポート
  - Install : ChaincodeをPeerに配布
  - Instantiate : ChaincodeをChannelに紐付けて初期化
  - Invoke : Chaincodeのデータ更新・参照機能を実行

- 以下のChaincode API は、ユーザ独自のChaincodeを実装する際に利用
  - GetArgs : トランザクションのメッセージから引数を取得
  - PutState/GetState/DelState : Ledgerの読出/書込/削除
  - InvokeChaincode : 他のChaincodeの呼び出し
  - (詳細未定) : アクセス制御

- Chaincode開発言語
  - Go
  - Java

- Chaincodeユーティリティ機能
  - Life-cycle management : Chaincodeのアップデート
  - Naming : Chaincodeの名前を設定



4.5 Data Privacy

HITACHI
Inspire the Next

1. Permissioned network
   - Fabric-ca が認証局(Certificate Authority; CA)として機能
     - Peerの登録・承認
     - Userの登録・承認
     - Cert(証明書)の発行・更新・破棄
   - MSP (Membership Service Provider) は、認証局が発行した証明書を使ってトランザクションメッセージの署名・検証を行う

2. アクセス制御
   - (a)プラットフォームと(b)アプリケーションの2レイヤでアクセス制御
     a) Channel Access Control : Channel上のChaincode/Ledgerへのアクセスを制御
     b) Invocation Access Control (*) : Chaincodeの関数へのアクセスを制御
   - 権限のあるユーザ/Peerのみが上記リソースにアクセス可能

3. データ暗号化 (*)
   - トランザクションのペイロードを暗号化
   - Ledgerに記録するトランザクションデータを暗号化

(*) Not supported yet

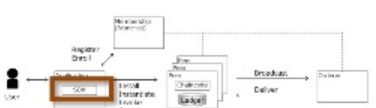
4.6 SDK

HITACHI
Inspire the Next

- SDK は、開発者によるクライアントアプリケーションの実装をサポート

- 以下のライブラリが提供されている
  - Membership Registration/Enrollment
  - Chain Construction
  - Request Transaction

- SDK は、以下の開発言語をサポート
  - Node.js
  - Python
  - Java
  - Go (新規提案あり)



5．アプリケーション開発

HITACHI
Inspire the Next

- アプリケーション実行にあたり以下が必要
  - GUI　　(Web)
  - Client　　(Node.js、Python、Java)
  - Chaincode　　(Go、Java)

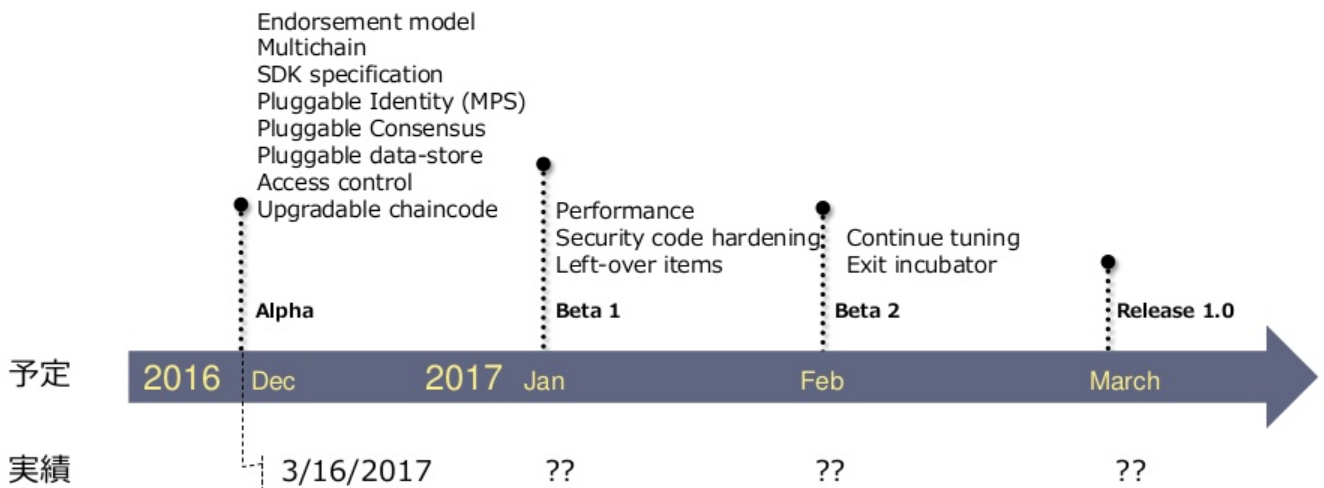
- 開発環境の構築方法
  - (Option 1) Docker（Mac/Linux/Windows用）：DockerHubからイメージ取得
  - (Option 2) Vagrant 開発環境　　：ローカルな開発環境を構築

- アプリケーション開発支援ツール
  - アプリケーションのテストツールを未提供(現時点)
  - サンプルアプリケーションを提供
  - Fabric Composer

6. ロードマップ

HITACHI
Inspire the Next

アジャイルで開発を推進しており、3末リリースに向け開発中



[Figure Source] High level data flows (https://jira.hyperledger.org/browse/FAB-37), Hyperledger Fabric DevUpdate12052016.pdf

7. まとめ

HITACHI
Inspire the Next

❖ v0.6からv1.0に向けた主な機能アップデートは以下

項目	v0.6	v1.0
分散合意形成 アルゴリズム	PBFT	エンドースメント＋ オーダリングに分離
スケーラビリティ	全ノード間での合意形成 ノードの動的追加が不可	スループット向上 ノードの動的追加が可能
認証局	メンバシップサービスが 全証明書を管理 単一障害点	認証局を分散配置 単一障害点を排除

出典

HITACHI
Inspire the Next

1) Overall architecture (https://github.com/hyperledger/fabric/blob/master/proposals/r1/Next-Consensus-Architecture-Proposal.md)
2) High level data flows (https://jira.hyperledger.org/browse/FAB-37)
3) Ledger (https://jira.hyperledger.org/browse/FAB-758)
4) SDK specification (https://docs.google.com/document/d/1R5RtIBMW9fZpli37E5Li5_Q9ve3BnQ4q3gWmGZj6Sv4/edit?usp=sharing)
5) Membership services (COP) (https://docs.google.com/document/d/1TRYHcaT8yMn8MZlDtreqzkDcXx0Wl50AV2JpAcvAM5w/edit?usp=sharing)
6) Multichannel (https://docs.google.com/document/d/1eRNxxQ0P8yp4Wh__Vi6ddaN_vhN2RQHP-IruHNUwyhc/edit?usp=sharing)
7) Gossip based data Dissemination (https://docs.google.com/document/d/157AvKxVRqgeaCTSpN86ICa5x-XihZ67bOrNMc5xLvEU/edit?usp=sharing)
8) Hyperledger Ordering Service (https://github.com/hyperledger/fabric/tree/master/orderer)
9) MSP & ACL (https://docs.google.com/document/d/1Qg7ZEccOIsrShSHSNI4kBHOFvLYRhQ3903srJ6c_AZE/edit#heading=h.2rmho7iqstbu)
10) Ordering with Kafka (https://docs.google.com/document/d/1vNMaM7XhOlu9tB_10dKnlrhy5d7b1u8lSY8a-kVjCO4/edit?usp=sharing)
11) Hyperledger JIRA (https://jira.hyperledger.org/secure/Dashboard.jspa)

HITACHI

Inspire the Next