Hyperledger Fabric v1.0 on a Raspberry Pi Docker Swarm

<http://www.joemotacek.com/hyperledger-fabric-v1-0-on-a-raspberry-pi-docker-swarm-part-1/>

<http://www.joemotacek.com/2018/03/>

<http://www.joemotacek.com/category/development/iot/>

<http://www.joemotacek.com/category/development/hyperledger/>

Hyperledger Fabric on Raspberry pi 3

<https://stackoverflow.com/questions/45800167/hyperledger-fabric-on-raspberry-pi-3>

Hyperledger Fabric & Apache Kafka

<http://hihellloitland.tistory.com/28>

<https://www.popit.kr/kafka-consumer-group/>

<https://www.techrepublic.com/article/apache-kafka-is-booming-but-should-you-use-it/>

<https://marutsingh.com/2016/09/12/why-kafka/>

<https://techbeacon.com/what-apache-kafka-why-it-so-popular-should-you-use-it>

<https://stackoverflow.com/questions/36755477/what-are-the-pros-and-cons-of-creating-kafka-broker-and-zookeeper-in-two-docker>

Hyperledger Fabric

Fabric이란?

* Hyperledger 내의 블록체인 프로젝트 중 하나로서 다른 블록체인 기술과 마찬가지로 장부가 있고 스마트 컨트렉트를 사용하며 참가자가 거래를 관리하는 시스템이다.
* Public Blockchain Network는 신원을 확인하기 위해 “작업 증명”을 하지만 Private Blockchain Network인 Hyperledger Fabric Network에서는 구성원이 Membership Service를 통해 등록한다.
* 여러 가지 플러그 가능 옵션을 제공하는데 원장 데이터는 여러 형식으로 저장할 수 있으며 합의 알고리즘을 전환할 수 있으며 다른 MSP가 지원됩니다.
* 채널을 생성하는 기능을 제공 하므로 참가자 그룹이 별도의 원장을 생성할 수 있습니다. 이것은 일부 참가자가 경쟁자 일 수 있고 모든 거래를 원하지 않는 네트워크에서 주로 사용되는 기능이다. ( 예시 : 일부에게만 정보를 공개하고자 할 때 공개하고자 하는 일부로만 구성된 채널을 구성하면 해당 채널의 참가자만이 해당 채널의 장부 사본을 갖는다. )

공유 원장 ( Shared Ledger )

* Hyperledger Fabric은 특정시점에서 원장 상태, 원장 데이터베이스인 World State와 현재 상태에서 모든 트랜잭션의 기록인 Transaction Log로 구성되어 있다.
* Transaction Log의 구성 요소는 World State의 업데이트 기록이다.

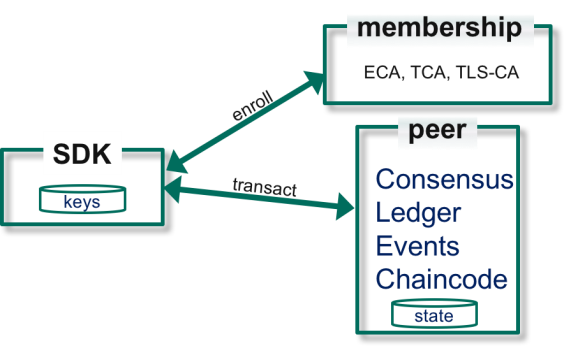
Smart Contracts

* Hyperledger Fabric에서의 Smart Contracts는 Chain Code로 작성되며 해당 응용 프로그램이 원장과 상호작용해야 할 때 블록체인 외부의 응용 프로그램에 의해 호출된다. 대부분의 경우 Chain Code는 원장의 Transaction Log가 아닌 World State[ ex) query ]에서만 상호 작용한다.
* Chain Code는 Go language로 작성된 프로그램이며 검증된 피어 프로세스에서 격리 된 보안 Docker Container에서 실행됩니다. 또한, Chain Code는 응용 프로그램에서 제출 한 트랜잭션을 통해 원장 상태를 초기화하고 관리합니다.
* Chain Code는 여러 프로그래밍 언어로 구현 될 수 있으며 현재 지원되는 체인 코드 언어는 Java 및

Fabric v0.6

1. Fabric v0.6 architecture

구성 : Membership Service, Peer

1. 역할
   1. Membership Service : 사용자에 대한 enroll 및 관련 인증서를 생성, 조회 담당 컴포넌트
   2. Peer : 블록체인 개념에서 필요한 작업들(Consensus, ChainCode, Event, Ledger 관리 등을 수행한다.
   3. 

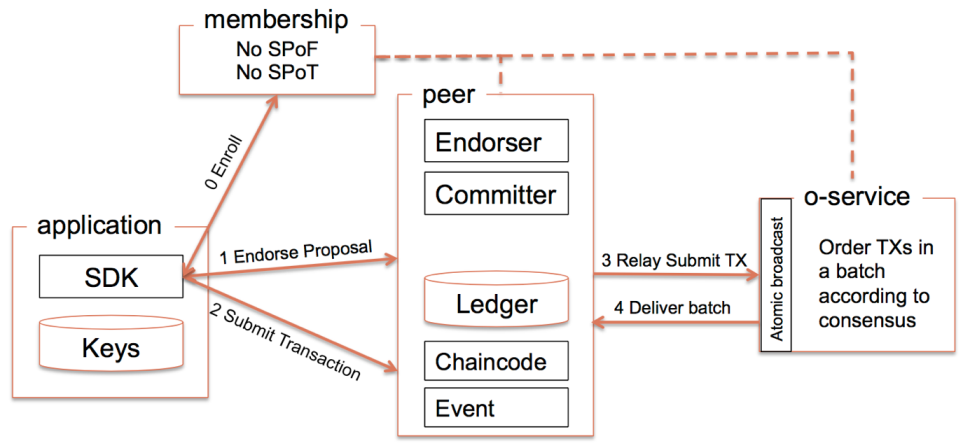
Fabric v1.0

1. Concept
   1. 트랜잭션에 서명(Endorsement)하여 비즈니스 프로세스에 반영
   2. 개인정보 보호 및 기밀유지 지원
   3. 참여자 수와 TPS 조절
   4. 비결정적(Non-deterministic) 트랜잭션 제거
   5. 플러그인 방식의 데이터 스토어 (원장관리를 위한 DB)
   6. Fabric과 체인코드의 동적 업그레이드 지원
   7. SPF(Single Point of Failure)를 제거하고 멀티플 멤버쉽 서비스를 가능하게 함
2. 변경점

2.1 Peer 역할의 분할

- Endorsing Peer ( Endorser )

- Ordering-service-node (Orderer)

- 

- v0.6에서의 REST API가 없어지고 grpc 또는 CLI를 통해서 블록체인 서버와 통신을 하며 다소 복잡해진듯이 보이는 트랜잭션의 흐름을 갖는다.

2.2 트랜잭션의 흐름

(1) 최초 Client가 블록체인에 트랜잭션 수행을 위한 제안을 한다.

(2) 제안은 Endorse Peer에서 수행됩니다.

(3) E