생성일	@2021년 3월 4일 오후 7:42
태그	

> 하이퍼레저 패브릭은 모듈러 아키텍쳐를 이용한 어플리케이션/솔루션 개발을 가능하도록 해주는 프레임 워크이다.

> 모듈러 아키텍쳐 : 서로 연결할 수 있는 개별 구성요소로 구성된 모든 시스템의 설계

하이퍼레저 패브릭은 **허가형 프라이빗 블록체인**의 형태를 가진다. 누구나 자유롭게 참여 가능한 퍼블릭 블록체인과 달리, **인증 관리 시스템**에 의해 허가된 사용자만이 블록체인 네트워크에 참여할 수 있다. 따라서 **패브릭 네트워크에 참여한 노드**들은 이미 시스템에 의해 **허가된 노드**로 볼 수 있고, 퍼블릭 블록체인에서 사용하는 **합의 알고리즘이 필요 없다.** (만약 필요하다면 합의 알고리즘을 네트워크 내에서 선택적으로 사용은 가능하다)

패브릭에서 모든 노드가 동일한 원장으로 정보를 공유할 수 있고, 비지니스 목적에 맞게 공유하고자 하는 노드간에만 **별도의 원장**을 생성하는 것이 가능하다.

하이퍼레저 패브릭은 네트워크 내에서 목적에 맞는 별도의 원장을 생성할 수 있는 **채널**을 제공함으로써 기업이 사용하기 용이하도록 고안되었다.

하이퍼레저 패브릭의 특징

1) Configurable Module 구조

블록체인 네트워크에는 많은 컴포넌트와 정책들이 있다. 그 모든 것들을 원하는 비지니스 모델에 맞춰 선택할 수 있다.

2) 일반 개발 언어 사용 가능

패브릭은 golang, nodejs, java 등을 지원한다.

3) Execute-Order-Validate 아키텍처

보통 실행하고 검증하는 다른 플랫폼들과는 달리 패브릭은 order단계가 추가된다. 이는 원장에 대한 트랜잭션이 **비결정성**을 갖는 경우를 배제하기 위함이다.

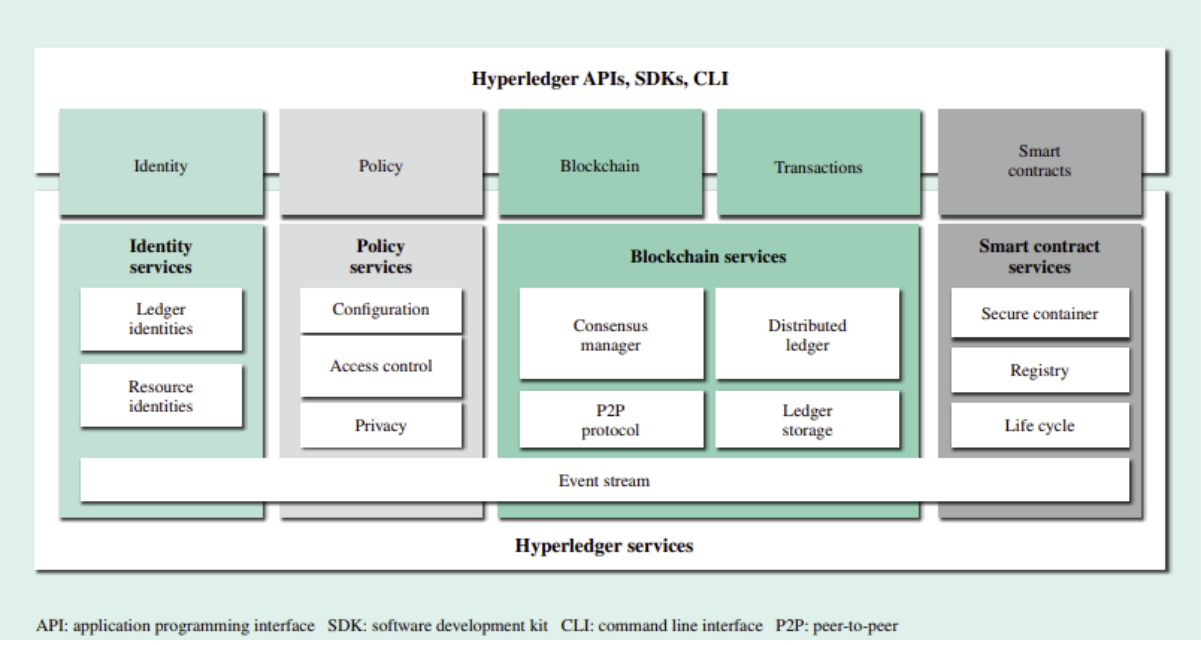
> 비결정성 : a라는 인풋에 대해 b라는 하나의 아웃풋이 결정되는 것이 아닌 b,c,...처럼 여러개의 아웃풋이 나오는 경우를 뜻한다.

다른 블록체인 플랫폼에서는 고유 언어(solidity)로 이를 컨트롤하지만, 패브릭에서는 일반 언어를 사용하되 아키텍쳐를 달리하여 이를 해결한다.

4) 데이터 기밀을 위한 채널

패브릭은 다양한 상황에서 유연하게 대처가능한 채널이라는 요소를 추가하였다.

하이퍼레저 패브릭 구성요소



Asset과 Ledger

블록체인은 '거래'를 전제로 하는 '네트워크'이다. 거래를 하면 무엇인가가 오고 가고 하는데, 이것이 바로 **Asset(자산)**이다. 자산은 key-value 쌍으로써 하이퍼레저 패브릭 내부에 존재하게 되고, 바이너라 값이나 JSON 형태로 표현된다.

거래가 진행됨에 따라 자산의 state가 생기는데, 이를 기록해둔 것이 **ledger(원장)**이다.

원장은 불변하고 순서가 있다. 채널당 하나의 원장을 가지고 있고, 각각의 피어는 그들이 속해있는 모든 채널에 원장 복사본을 관리하게 된다.

Chaincode/Smart Contract

Chaincode는 블록체인 네트워크 외부의 클라이언트 응용 프로그램에 의해 호출되는 코드로, 전체 상태(World State)의 일련의 키-값쌍에 대한 접근 및 수정을 관리한다. 쉽게 말해 원장에 새로운 내용을 업데이트 하거나 기존의 내용을 읽어 오기 위해 필요한 것이다.

> 전체 상태(World State) : 체인 트랜잭션 로그에 포함된 모든 키의 최신 값을 나타내는 데이터베이스.

체인 코드에 의해 생성된 상태는 해당 Chaincode로만 범위가 지정되며 다른 Chaincode로 직접 액세스 할 수 없다. 그러나 동일한 네트워크 내에서 적절한 권한이 주어지면 Chaincode는 다른 Chaincode를 호출하여 해당 상태를 액세스 할 수 있다.

Chaincode는 일반적으로 네트워크 구성원이 동의한 비즈니스 논리를 처리하므로 'smart contract'라고도 불린다.

Transaction

Transaction은 블록체인 네트워크에서 비즈니스 로직을 수행하기 위한 요청으로, 보통 블록체인 네트워크에서의 **이벤트**이다.

체인코드(스마트 컨트랙트)가 설치되거나 호출될때 발생한다.

- invoke transaction(호출 트랜잭션) : 원장의 읽기/쓰기
- instantiate transaction(인스턴트화 트랜잭션) : 채널에서 체인 코드를 시작하고 초기화하는 요청
    - 인스턴트화 : 다른 피어들에게 체인코드에 대해 알리는 역할을 한다.

> 인스턴트화를 하는 이유? 체인코드가 어떤 피어에 설치되면 바로 사용할 수 없다. 왜냐하면 그 피어가 호스팅하는 채널에 연결된 다른 구성 요소들은 어떤 체인코드가 설치됐는지 모르기 때문이다. 때

> 문에 체인코드를 사용하기 위해서는 구현 로직이 아닌 해당 체인코드의 인터페이스를 다른 피어에게 알려야한다.

Organization

Organization은 노드들을 **특정 목적에 따른 논리 집합**으로 채널과 유사하다. 멤버라고도 한다.

- *MSP(Membership Service Provider)**를 통해 조직을 네트워크에 가입시키며 이는 블록체인 네트워크 서비스를 제공하는 기업등에서 수행된다.

조직들이 모여 Consortium을 구성하는데, 모든 조직이 컨소시움의 일원은 아니지만 모든 조직은 블록체인 네트워크의 일원이다.

Consortium

Consortium은 공동의 목표를 가지고 **트랜잭션 내역을 공유하며 협력하는 조직의 집합**이다. 네트워크당 여러개 존재 가능하지만 보통 한개 존재한다.

컨소시움의 조직들은 각각 피어를 가지고 채널을 형성하거나 채널에 참여한다. 컨소시엄을 구성하면 그 조직들은 트랜잭션 내역을 공유할 수 있게 된다.

Channel

Channel은 컨소시엄 내 그룹간 커뮤니케이션 메커니즘으로, **서로 다른 노드들을 묶는 역할**을 한다. 채널을 통해 거래와 원장을 분리 가능하기 때문에 **독립적인 원장을 가지는것이 가능**하다.

하나의 채널에는 하나의 원장을 가진다. 이를 통해 private transaction의 수행이 가능하다.

다른 채널의 체인코드를 호출(invoke)하는 것은 **읽기**만 가능하다.

채널 별 원장은 채널의 피어간에 공유되며 거래 당사자는 해당 채널과 상호 작용하기 위해 채널에 올바르게 인증되어야한다.

Peer

Peer는 **원장과 체인코드를 관리**하며 패브릭 네트워크를 구성하는 노드이다.

패브릭 네트워크 참여자들은 peer에 설치되어 있는 체인코드 실행 요청을 통해 peer에 저장된 원장에 데이터를 읽거나 쓸 수 있다.

peer는 수행하는 역할에 따라 다음과 같이 크게 4가지로 구분된다.

- Endorsing peer : 체인코드 시뮬레이션을 통해 **트랜잭션이 적절한지 판단하는 역할**을 한다. 위에서 언급한 3단계 중 execute 단계에 속한다.

- Committing peer : 모든 peer가 수행하는 역할로, **최신 블록에 대한 검증**을 한다. 위의 3단계 중 validate 단계에 속한다.
- Anchor peer : **다른 조직과의 통신**을 위해 다른 조직의 peer와 통신하는 역할을 한다.
- Leader peer : **orderer와 연결되어 최신 블록을 전달받아 조직 내 다른 peer들에게 전송**하는 역할을 한다.

OSN(Ordering Service Node)/Orderer

Orderer는 Endorsing peer들이 시뮬레이션을 통해 적절하다고 판단한 트랜잭션들을 모아서 정렬한 후 **실제 블록을 생성하는 노드**이다.

orderer는 방대한 양의 **트랜잭션을 검증**하고 이들을 모아서 **블록을 생성하는 작업**을 한다. 트랜잭션이 많아지면 오버헤드가 걸리는데, 이를 위해 **MQ Solution(message queue)**을 사용한다.

발생한 트랙잭션들을 MQ에 적재해서 **orderer가 부하를 견디도록 도와주는 역할**을 한다. 보통 **solo**나 **kafka**방식을 사용한다.

- solo 방식 : 보통 테스트용으로 orderer하나가 정렬 및 블록 생성의 모든 과정을 담당하는 방식
- kafka 방식 : 분산 메시징 시스템인 kafka cluser를 통해 orderer가 트랜잭션을 정렬하고 블록을 생성하는 방식

orderer는 블록을 생성한 후 자신에게 연결되어 있는 leader peer들에게 블록을 전달하고, leader peer들이 다시 자신이 속한 채널의 peer들에게 블록을 전달하면 peer들은 블록을 검증한 후 자신의 원장에 추가한다.

Client

Client는 블록체인에 접근하기위해 필요한 노드이다.

트랜잭션을 생성하여 피어 노드의 엔도저(endorser, 보증)에 보내고, 거래 보증응답이 오면 거래 제안을 생성하여 orderer에게 보낸다.

MSP, CA

허가형 블록체인인 패브릭은 사용자의 권한 및 인증을 위해 **MSP(Membership Service Provider)라는 인증 관리 시스템**을 사용한다. MSP에는 네트워크 내 노드의 역할과 권한 등이 정의되어 있다.

이러한 **MSP를 발급하고 관리하는 역할을 하는 기관**은 **CA(Certificate Authority)**라고 한다. 사용자를 인증해 주는 것은 중요한 역할이므로 CA는 보통 신뢰 있는 기관이 담당하는데, 하이퍼레저 패브릭에서는 **fabric-CA 노드**가 그 역할을 수행한다.