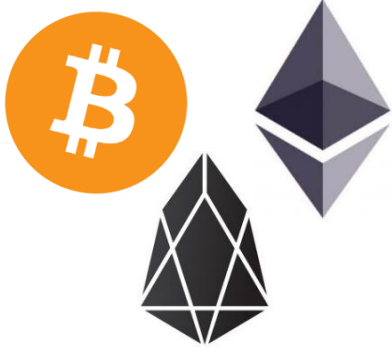


하이퍼레저 페브릭 개요

Hyperledger Fabric 소개



퍼블릭 블록체인

- 네트워크 누구나 참여 가능



프라이빗 블록체인

- 허가받은 사용자만 참여 가능



HYPERLEDGER **FABRIC**

Hyperledger Fabric 소개



Distributed Ledgers



Java-based
Ethereum client



Permissionable smart
contract machine (EVM)



Enterprise-grade DLT
with privacy support



Decentralized identity



Mobile application focus



Permissioned & permissionless
support; EVM transaction family

Libraries



Tools



Domain-Specific



Hyperledger Fabric 소개



Enterprise-grade DLT
with privacy support

- IBM이 주도하는 프로젝트
- 프라이빗 블록체인에서 가장 유명함
- 접근 제어 기능을 제공



- DID 제공 플랫폼



Permissioned & permissionless
support; EVM transaction family

- Intel이 주도하는 프로젝트
- IoT 환경에서의 블록체인

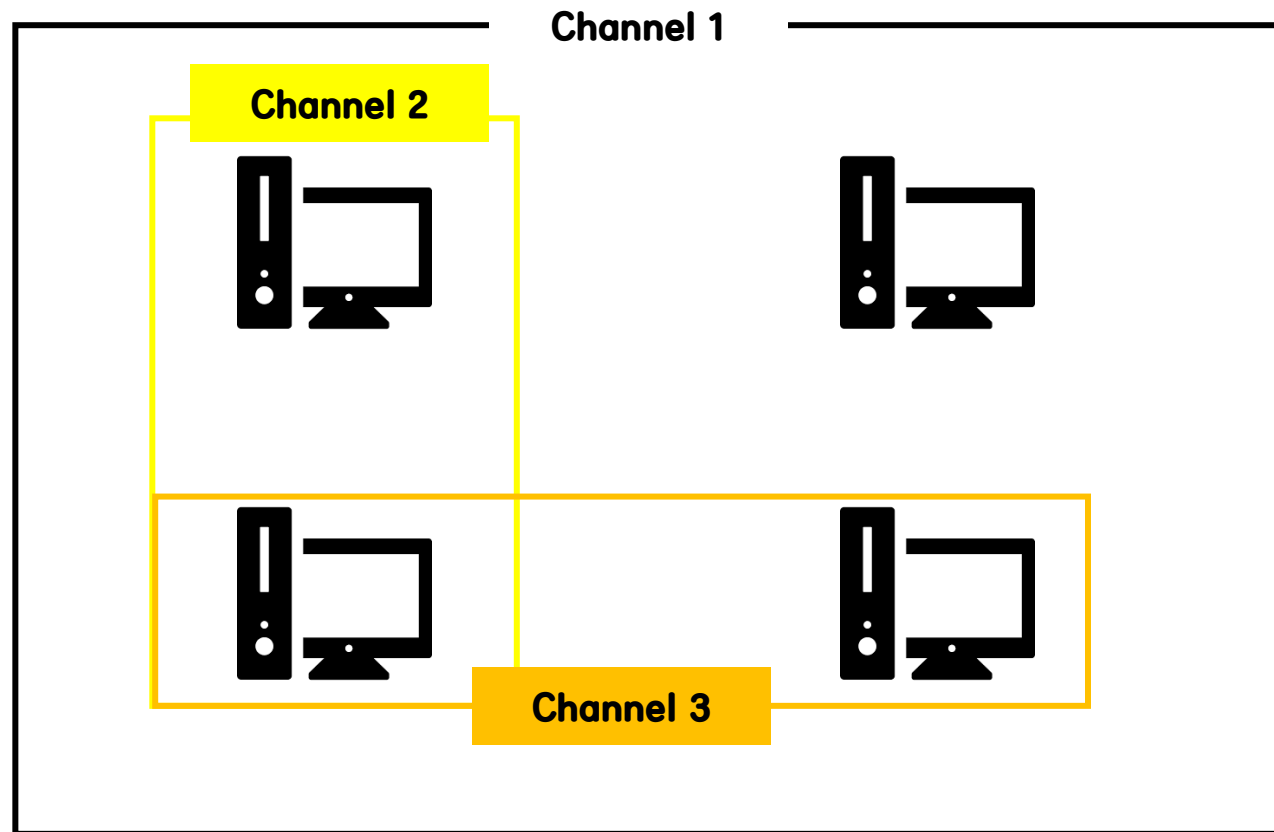


- 이더리움 프라이빗 네트워크

Hyperledger Fabric 소개

- Private Blockchain의 선두 주자
- 아무나 참여할 수 있는 Public Blockchain과 달리 허가된 사람만 참여 할 수 있는 구조
- P2P Network의 형태를 띄고 있으나, 완전한 P2P Network 구조는 아님
- Channel, 정책 등을 통해서 블록체인 참여자들 간의 프라이버시를 강화

Hyperledger Fabric 소개



Hyperledger Fabric 소개

Fabric의 데이터 처리 과정



```
graph LR; Execute --> Order; Order --> Validation;
```

Execute → Order → Validation

Execute — 트랜잭션을 실행하고 결과값을 검증하는 작업을 수행

Order — Execute 단계에서 검증이 끝난 트랜잭션을 취합하여 순서에 맞게 정렬한후 블록을 생성

Validation — 블록에 포함된 모든 트랜잭션에 대한 검증을 수행 후, 최신 블록 업데이트

Hyperledger Fabric 소개

Fabric의 데이터 처리 과정

기존 블록체인

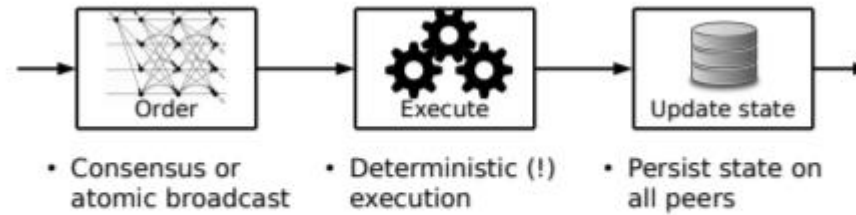


Figure 1: Order-execute architecture in replicated services.

Fabric

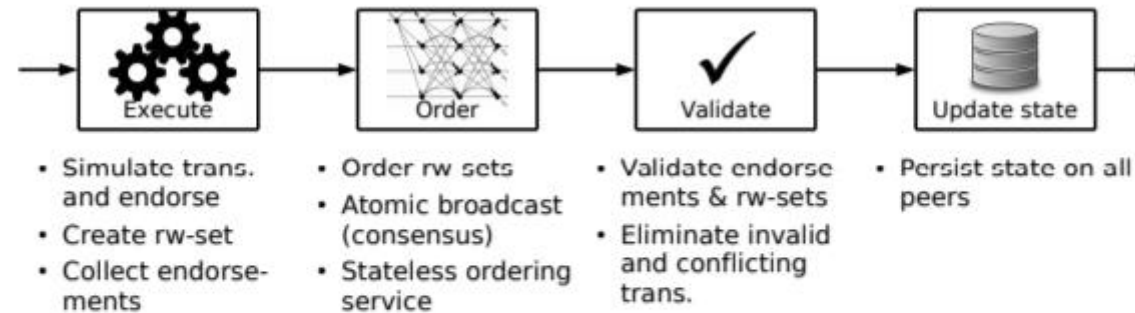
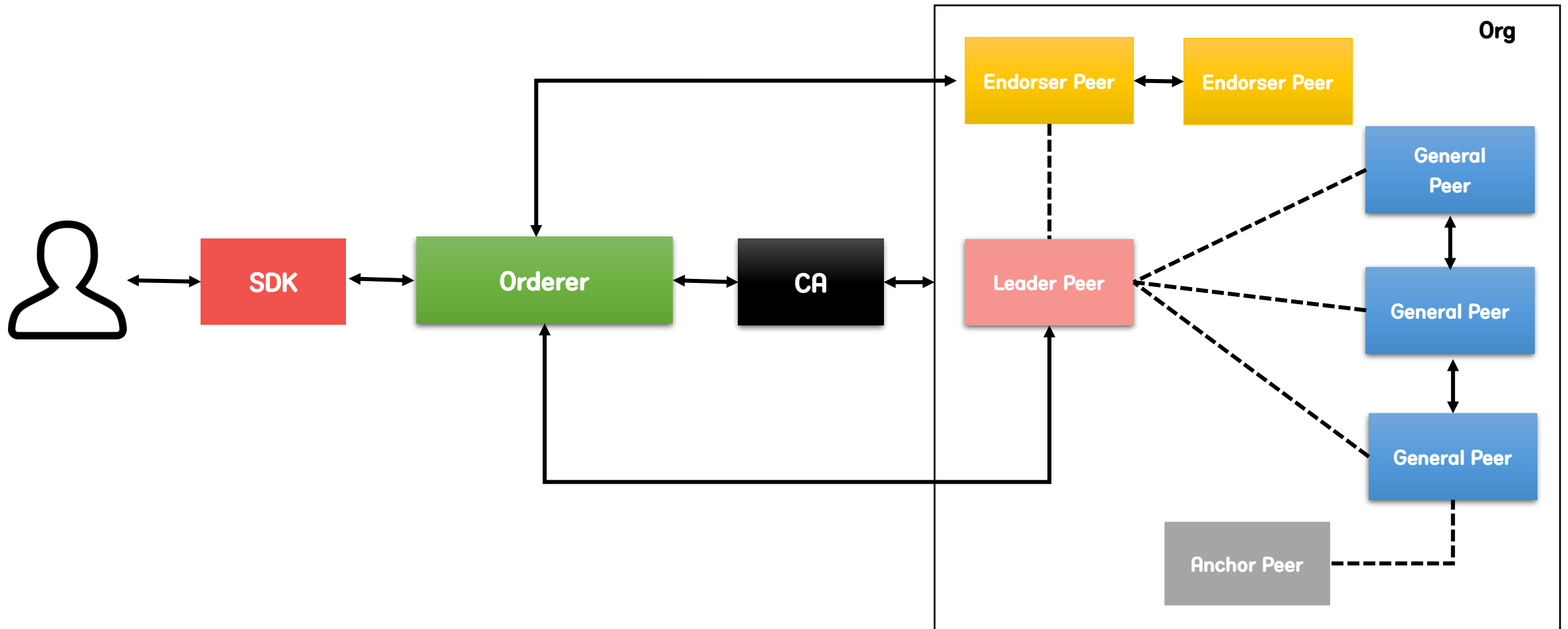


Figure 2: Execute-order-validate architecture of Fabric (*rw-set* means a readset and writeset as explained in Sec. 3.2).

Hyperledger Fabric 소개

Fabric Architecture



Hyperledger Fabric 구성요소



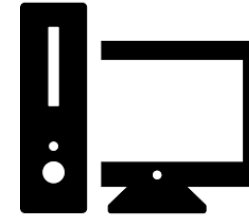
**General
Peer**

- Committing peer
- 분산원장, chaincode관리
- 모든 노드는 이 역할을 수행



Endorser Peer

- 트랜잭션을 Endorsement Policy 에 의해 검증하기 위한 Peer



**Anchor
Peer**

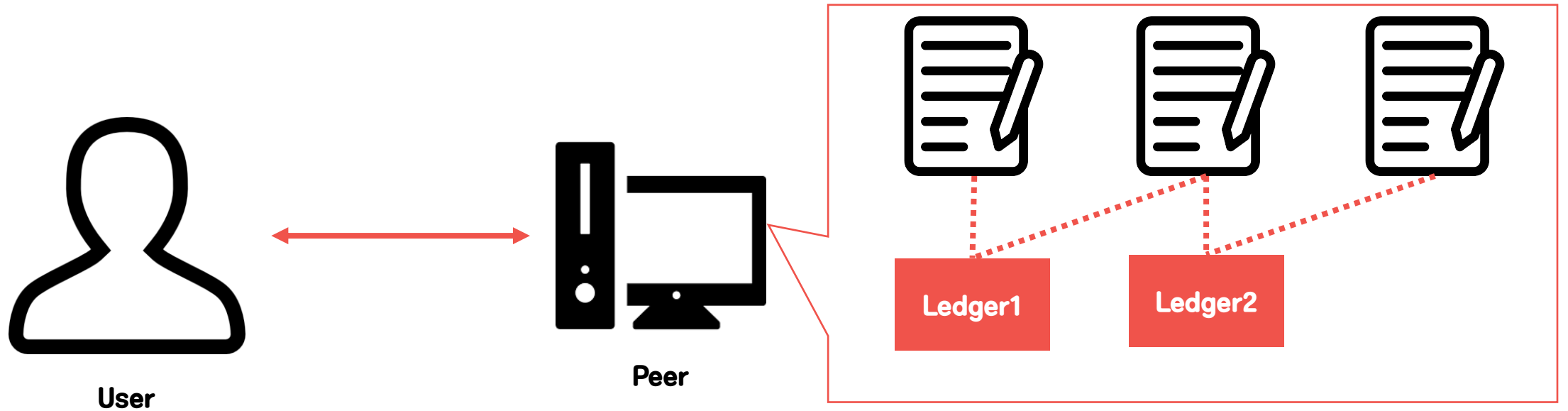
- 외부와 통신하기 위한 Peer



**Leader
Peer**

- Orderer와 통신하기 위한 Peer

Hyperledger Fabric 구성요소



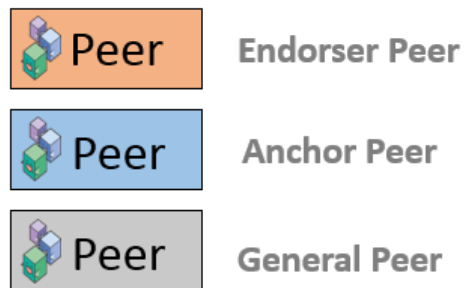
- Public blockchain 에서 peer는 참여자를 의미하지만, Fabric에서는 의미가 다름
- 분산원장과, Chaincode를 관리하는 역할
- 참여자는 peer에 설치된 Chaincode를 호출하여 정보를 읽어 올 수 있음.

Endorsement Policy

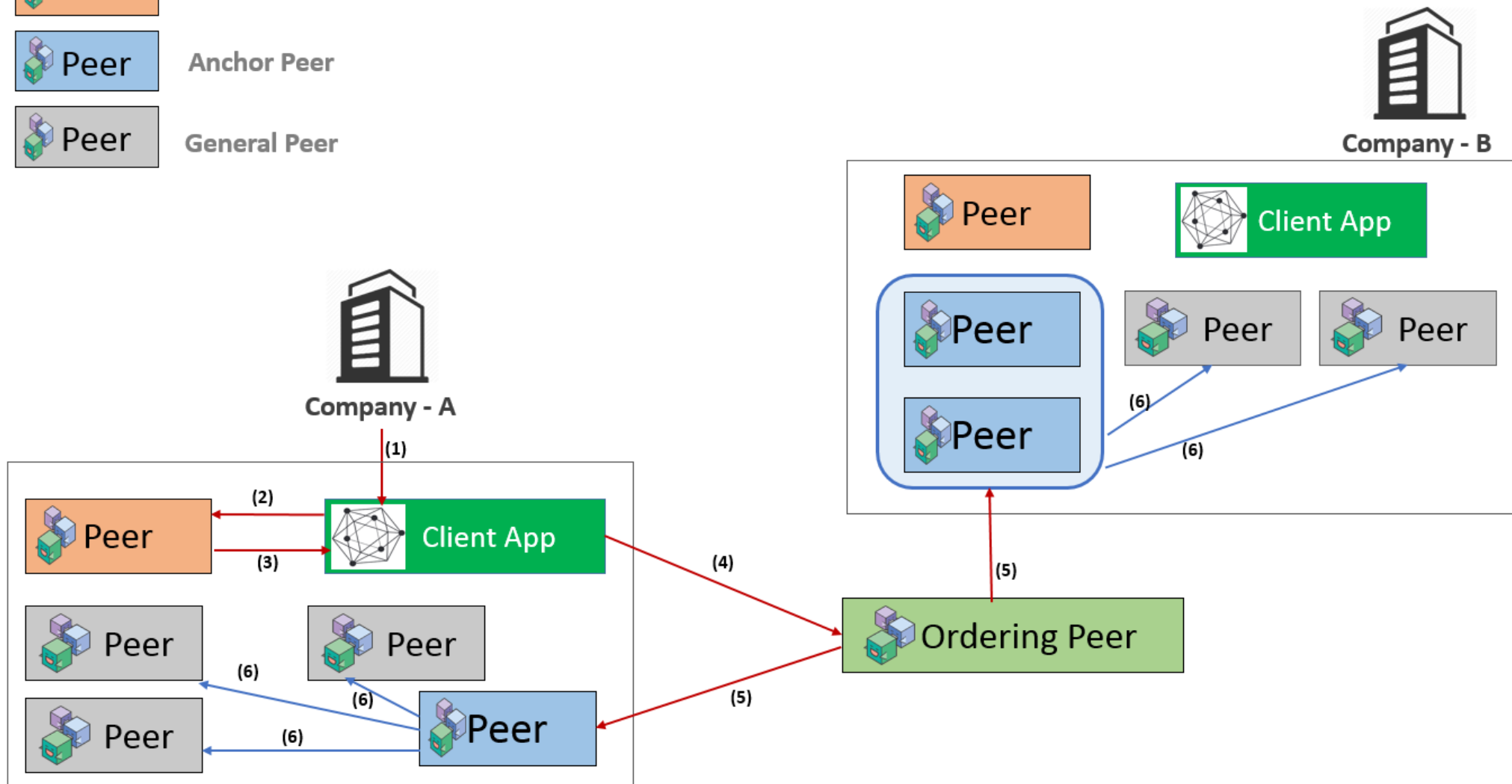
Transaction 검증 Rule을 정함

- Endorser peer = {peer 1=15, peer2 = 5, peer 3 = 30}
- 가중치의 합이 20을 넘어야 검증 완료
- {peer1 OR peer3} AND {가중치 20이상}

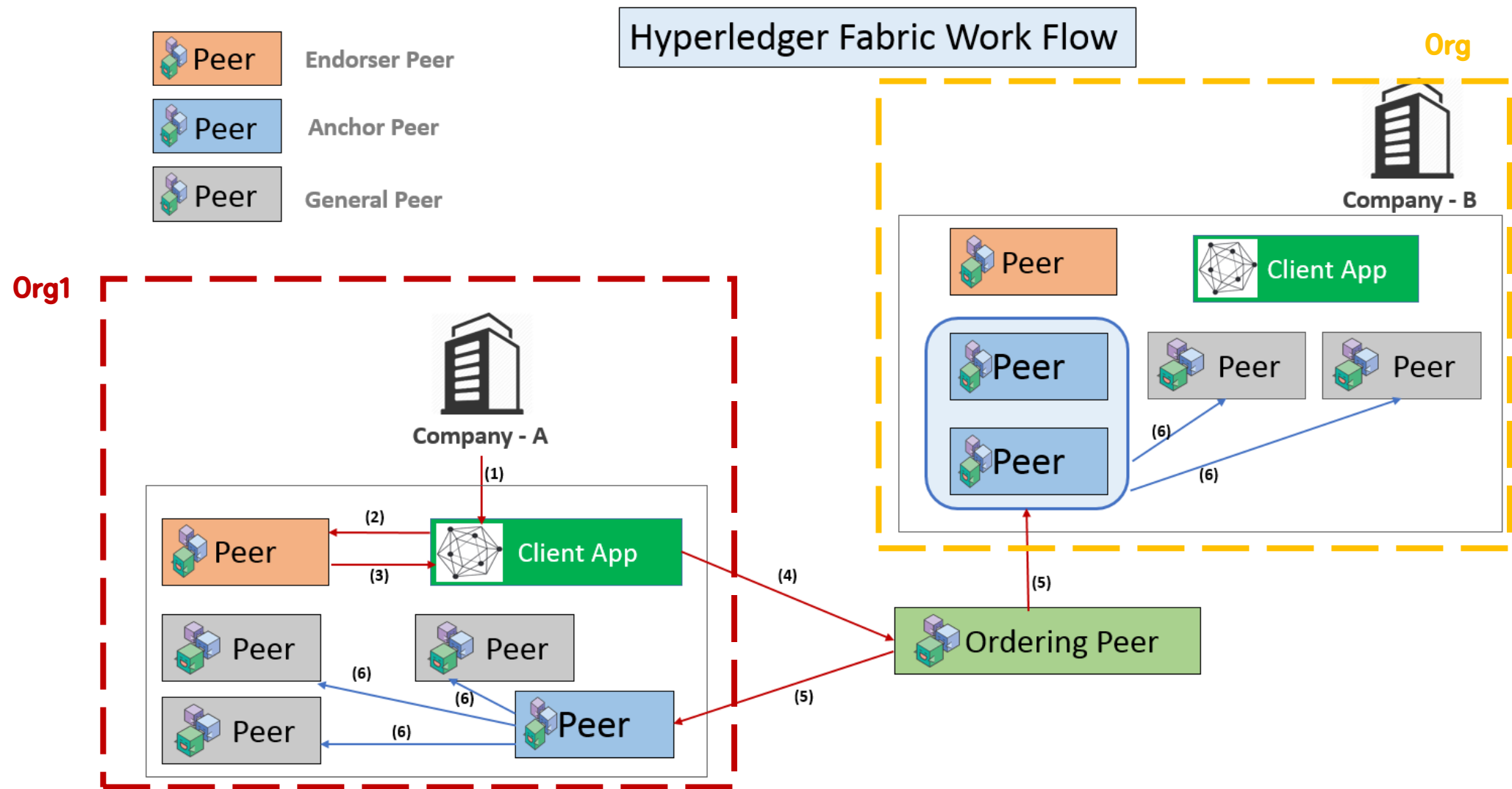
Hyperledger Fabric 구성요소



Hyperledger Fabric Work Flow

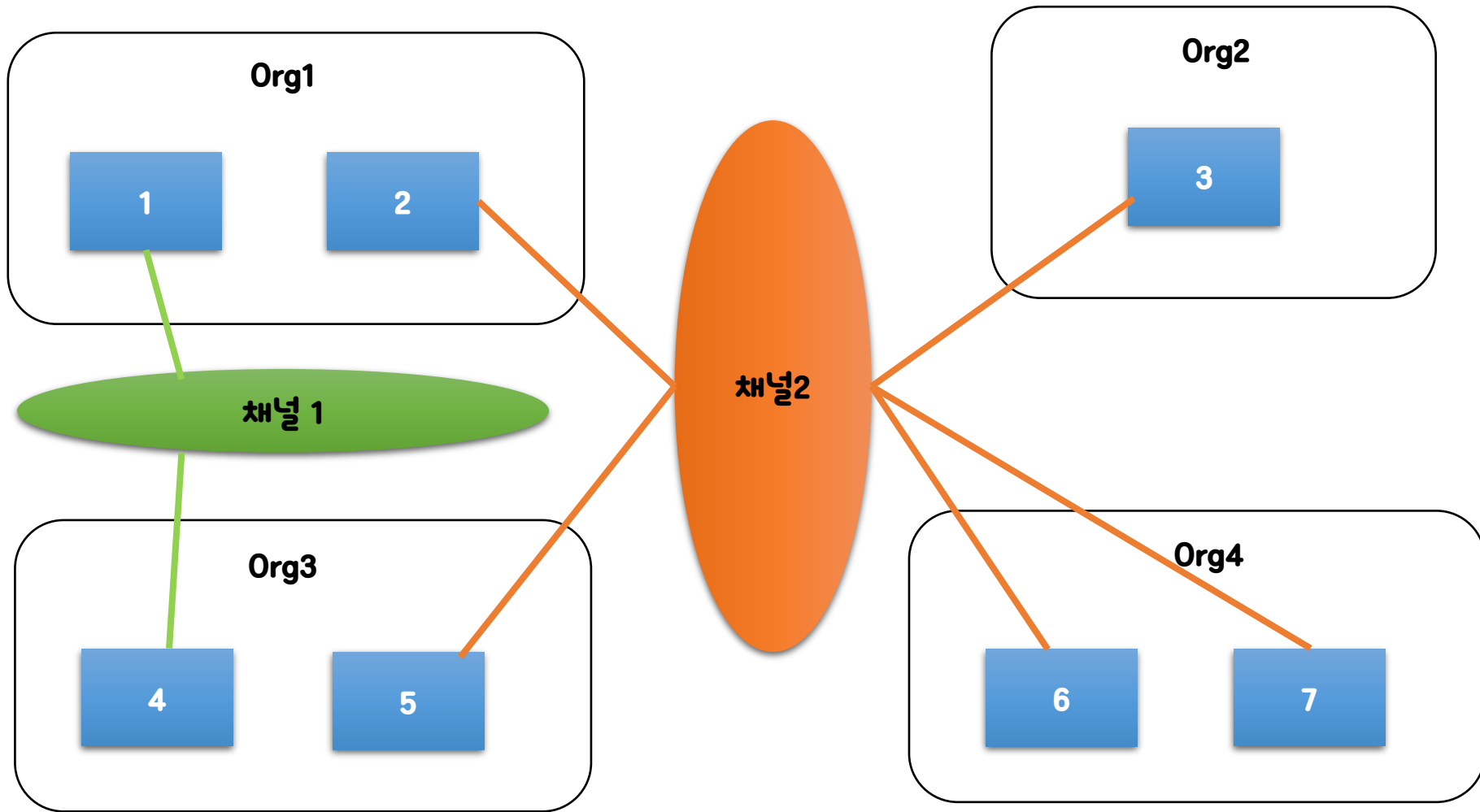


Hyperledger Fabric 구성요소



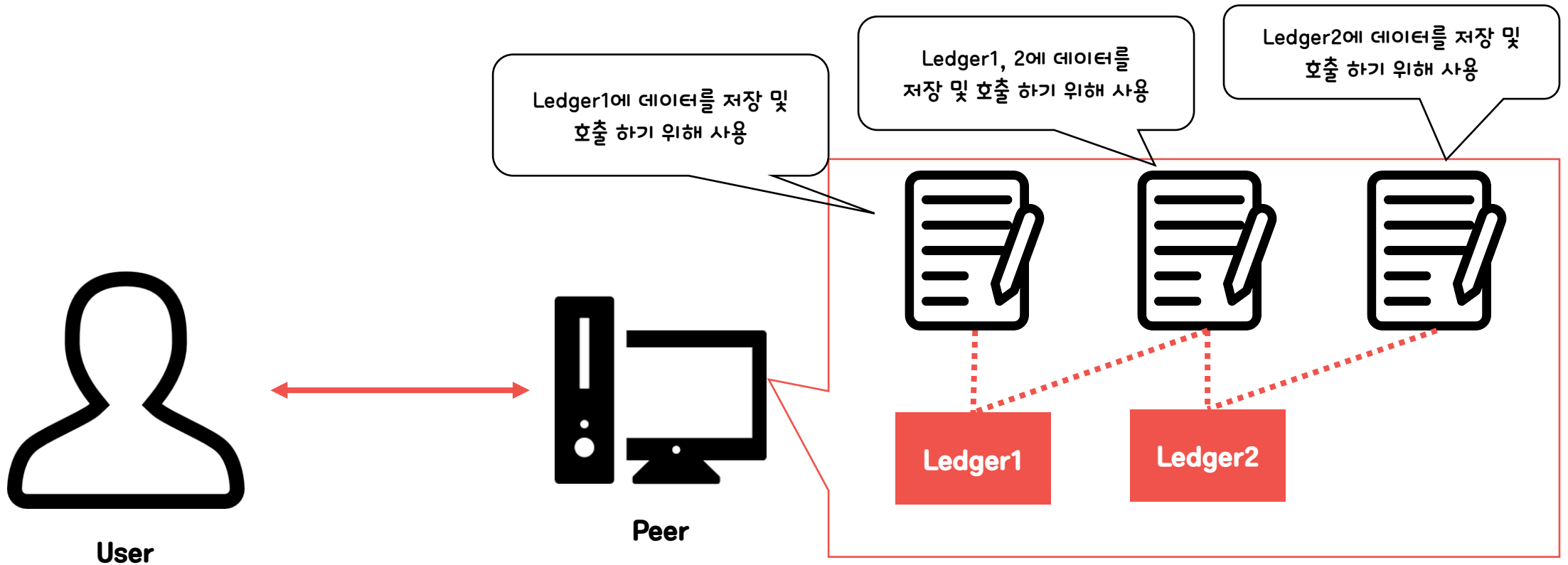
<https://medium.com/coinmonks/how-does-hyperledger-fabric-works-cdb68e6066f5>

Hyperledger Fabric 구성요소



Hyperledger Fabric 구성요소

Fabric에서 모든 데이터를 읽고 쓰기 위해서는 Chaincode 를 통해서 진행함



Hyperledger Fabric 구성요소

Fabric의 Smart Contract = Chaincode

Chaincode



Chaincode

Ethereum 의 Smart contract와 비슷한 기능

데이터를 읽고 쓰는데 사용

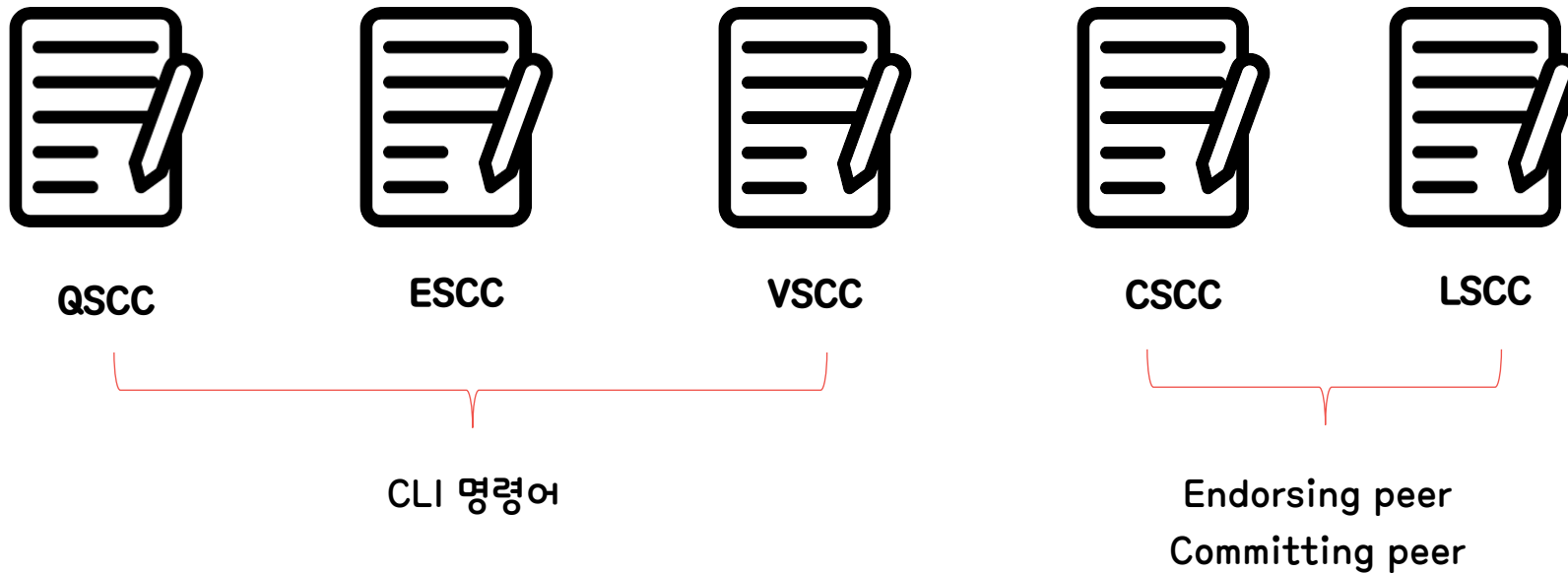
System Chaincode

시스템 레벨에서의 설정이 필요할 때 사용됨

트랜잭션 보증, 블록 검증, 채널 설정 등

Hyperledger Fabric 구성요소

System Chaincode는 Application 이나, User와 연결되는
일반 Chaincode랑 달리 fabric 시스템을 구성하기 위해 사용됨



Hyperledger Fabric 구성요소



QSCC

- 블록체인에 저장된 데이터를 읽어올 때 사용됨
- 블록번호, 블록 해시값, 트랜잭션 ID등을 통해 데이터를 읽어올 수 있음



ESCC

- 보증 정책을 담당하는 System Chaincode

Hyperledger Fabric 구성요소



VSCC

- 블록 검증을 할 때 사용되는 System Chaincode



CSCC

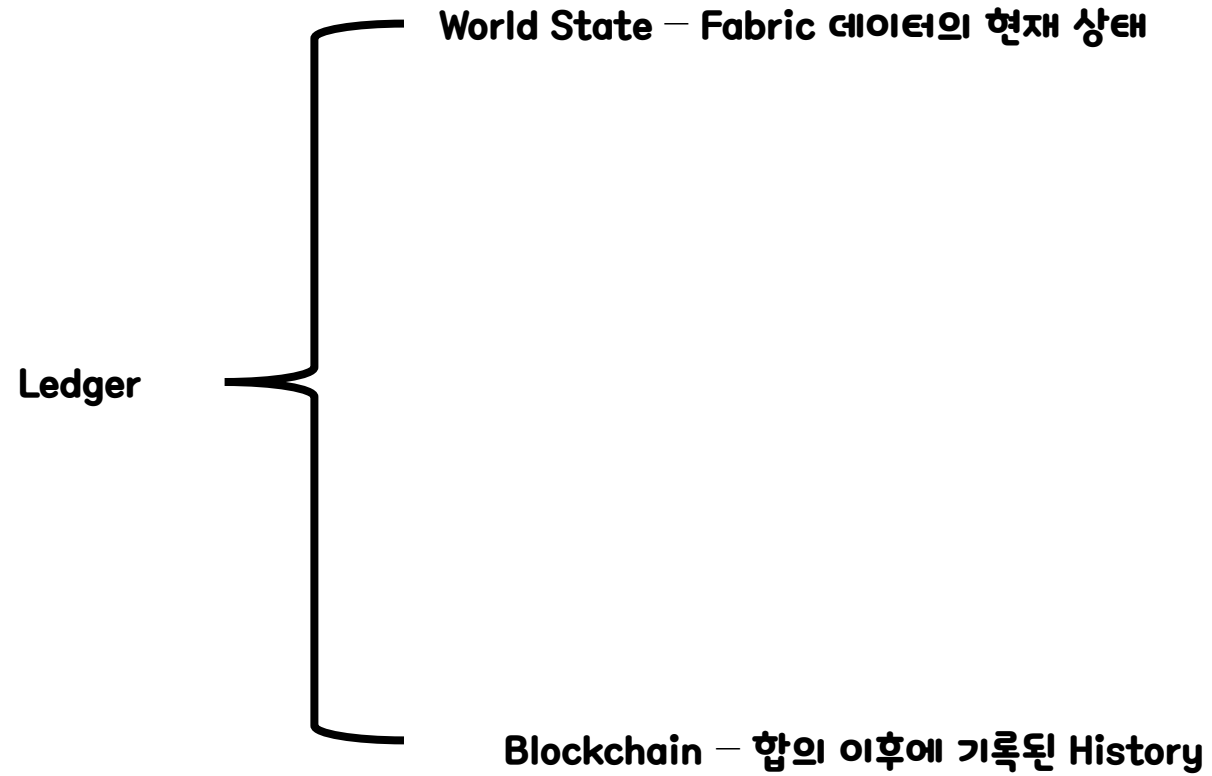
- 채널 설정시 사용되는 System Chaincode
- Peer를 채널에 참여 시키는 기능을 제공
- Peer channel create/join 명령어를 사용할 때 실행됨



LSCC

- Chaincode에 관련된 모든 과정을 수하는데 사용되는 System Chaincode
- Peer chaincode install/instantiate 명령어를 사용할 때 실행됨

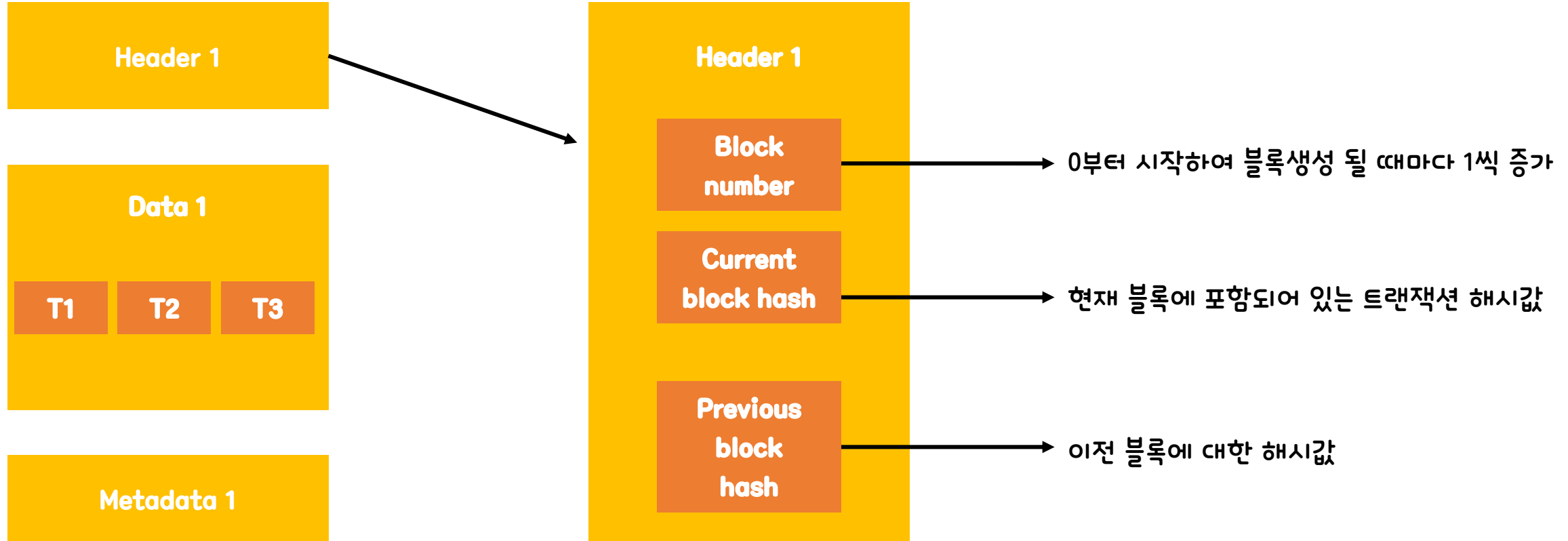
Hyperledger Fabric 구성요소



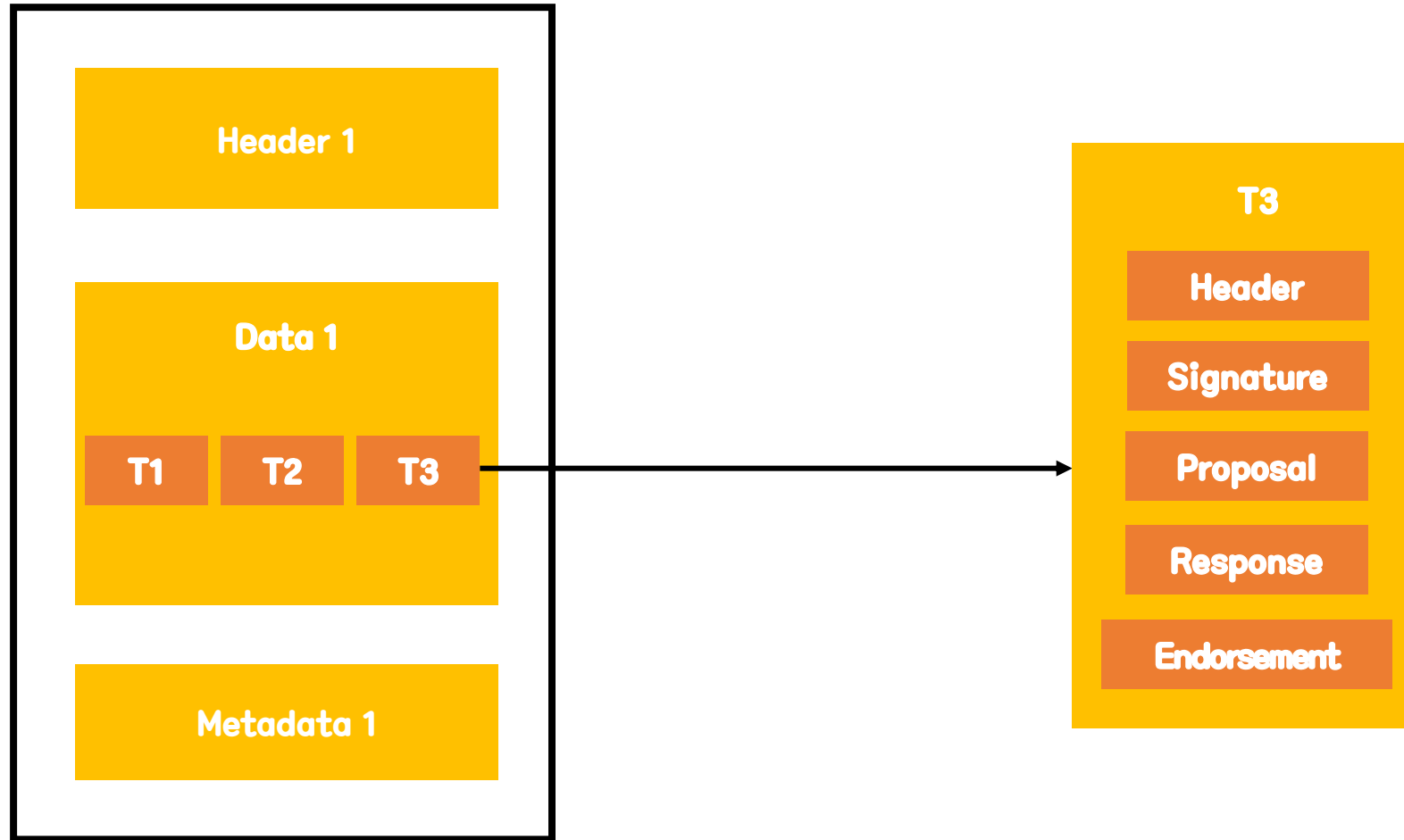
World state

- 블록체인과 분리되어 있음
- 합의가 이루어지기 전까지 체인코드를 통해 조회/변경/삭제 가능
- DB (LevelDB, CouchDB)에 저장
- 트랜잭션의 Version과 World state Version을 비교하여 데이터를 업데이트

Hyperledger Fabric 구성요소



Hyperledger Fabric 구성요소



Hyperledger Fabric 구성요소

Header — 트랜잭션의 version 정보와 트랜잭션이 실행되는 Chaincode 이름 명시

Signature — 트랜잭션 생성자의 관련 identity

Proposal — Chaincode에 들어가는 트랜잭션의 입력값이 저장됨, 해당 입력값을 이용해서 체인코드 실행

Response — 트랜잭션 처리 결과값을 Read/Write set으로 반환

Endorsement — 트랜잭션을 보증해 준 peer의 identity 가 포함됨

Metadata — 블록 생성자의 identity정보, 블록에 포함되어 있는 transaction 보증 여부



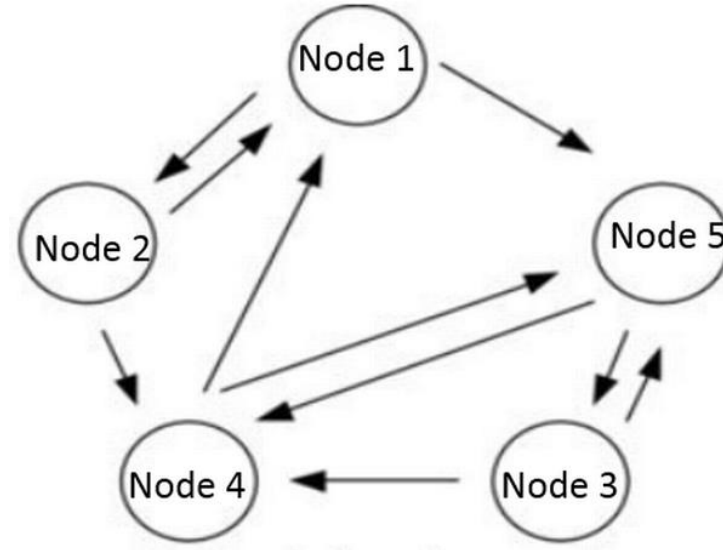
Ledger 검증

- 블록 검증시에는 해당 블록에 해당하는 Endorsement Policy를 확인
- 각 트랜잭션마다 Read/Write Set을 확인하여 결과 값이 올바른지 확인
- 블록과 Transaction Version을 확인

Ledger 업데이트

- 트랜잭션 검증 작업 후에 유효 태그를 가진 트랜잭션 내용만을 World State 에 업데이트
- 무효 태그를 가진 트랜잭션은 블록체인에 저장

Hyperledger Fabric 구성요소



Gossip Protocol

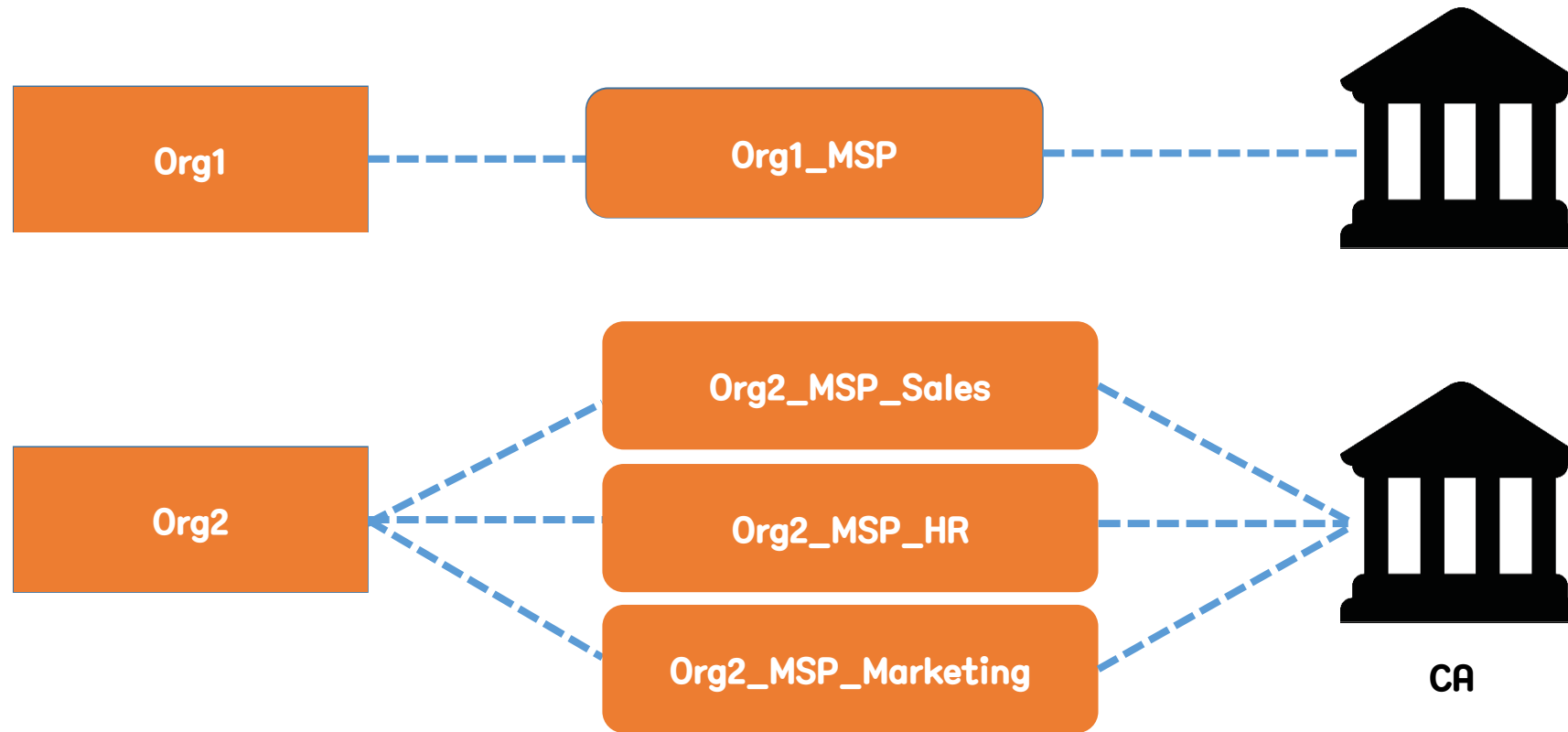
- 같은 채널에 속해 있는 노드에게 메시지를 전달할때 노드 전체에게 알리지 않아도 됨
- Peer가 접속 상태가 양호한지 확인하기 위해 사용
- Orderer가 Leader peer와 통신하면서 시작됨

Hyperledger Fabric 구성요소



- CA (Certificate Authority)
- 모든 노드의 인증서 관리 및 네트워크 접속 관리
- Private Key 발급 및 Public Key 관리

Hyperledger Fabric 구성요소

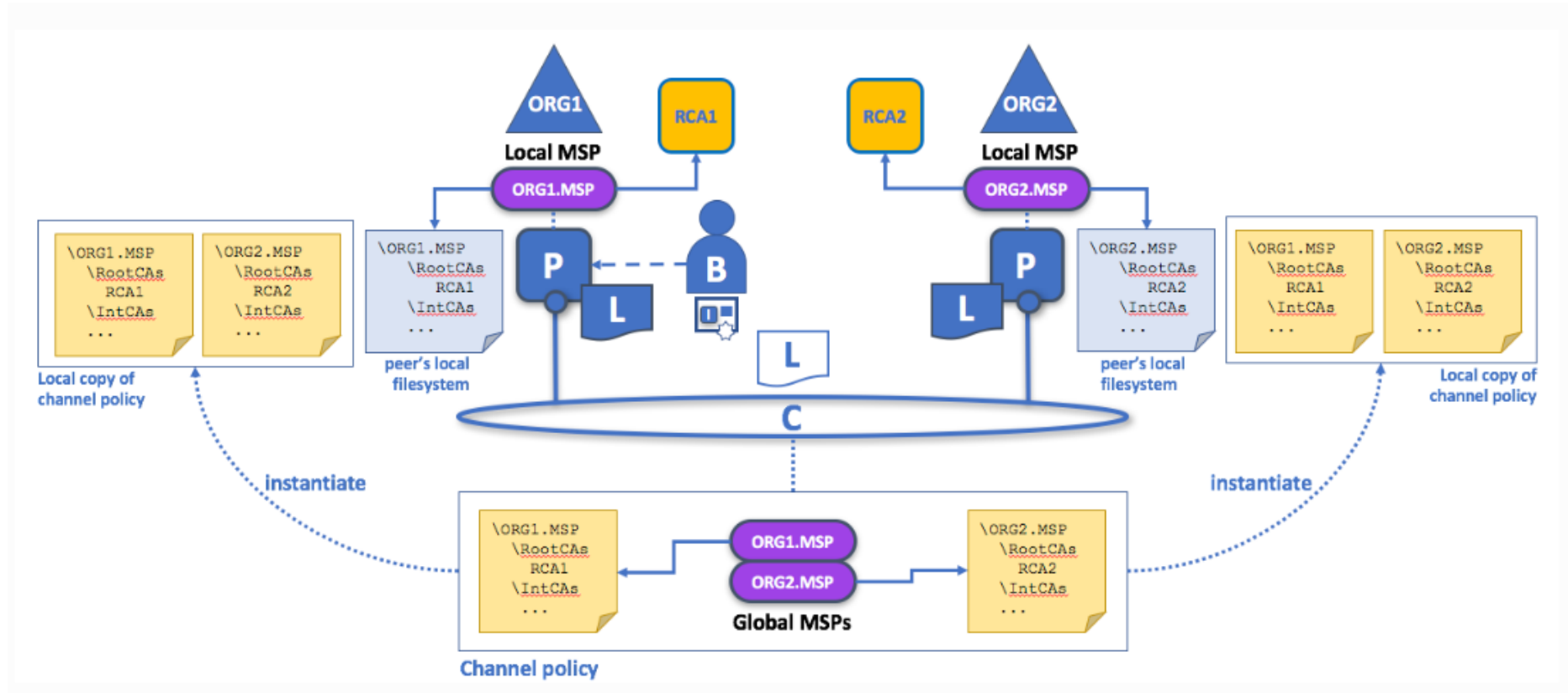


- MSP – Membership Service Provider
- Identity 기술을 바탕으로 만든 Fabric의 멤버십 관리 기술

MSP는 두 가지 종류로 구분 할 수 있음

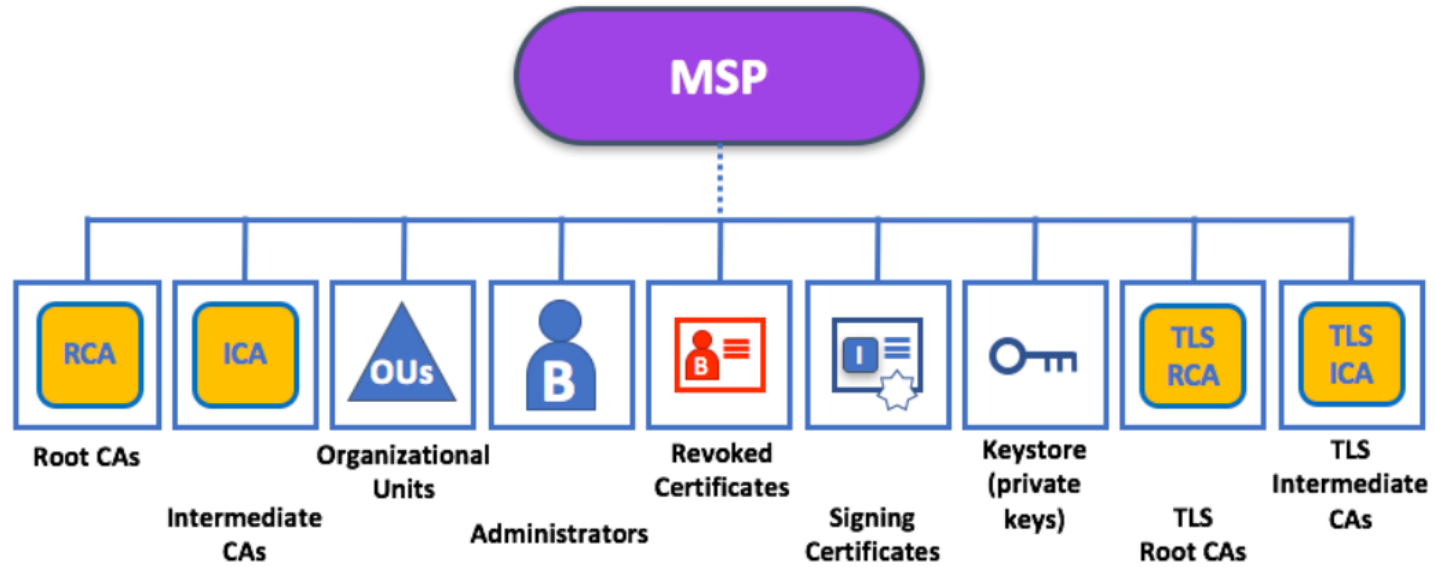
- Local MSP는 Fabric에서 노드의 역할을 부여할 때 사용하는 MSP
- Local MSP를 통해서 어떤 노드가 Peer, Orderer, Client를 정의 가능
- Fabric에 접속한 모든 노드는 하나 이상의 Local MSP가 정의되어 있어야함

Hyperledger Fabric 구성요소



- Channel MSP는 채널 구성원들에 대한 정의와 권한을 부여할 때 사용
- 채널 구성원들의 Local MSP를 이용해서 하나의 Channel MSP를 생성

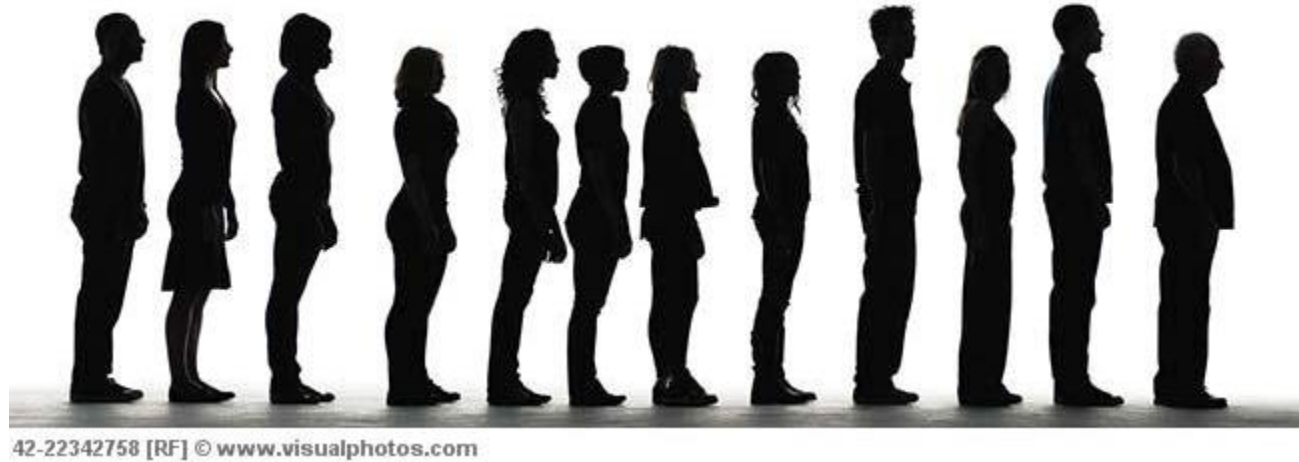
Hyperledger Fabric 구성요소



- Root CA: Fabric의 CA의 디지털 인증서
- Intermediate CA: CA가 여러 개로 구성이 될 때 세분화 하여 MSP 관리가능
- Organizational Units: ICA를 사용하지 않고, 하나의 RCA를 사용하여 세분화 할 때 사용
- Administrators: Organizations 운영자의 인증서
- Revoked Certificate: 폐기된 인증서
- Signing Certificate: Private Key로 암호화한 인증서
- TLS Root CA: 보안 강화를 위해 TLS 기능을 사용할 때 RCA로 부터 받은 TLS 인증서
- TLS Intermediate CA: CA로부터 발급받은 TLS 인증서

Hyperledger Fabric 구성요소



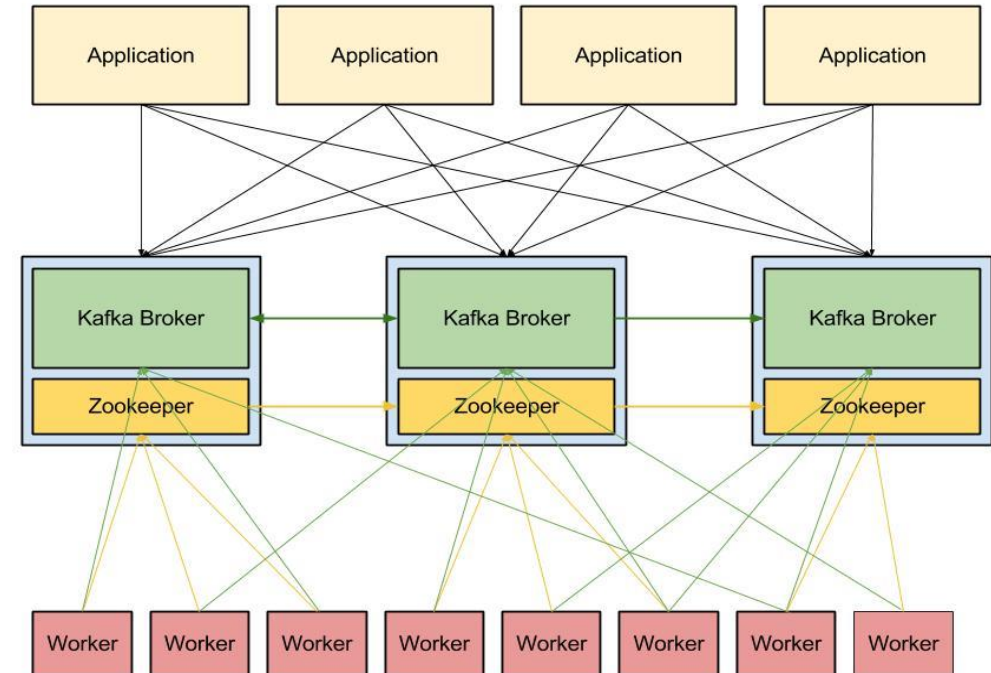
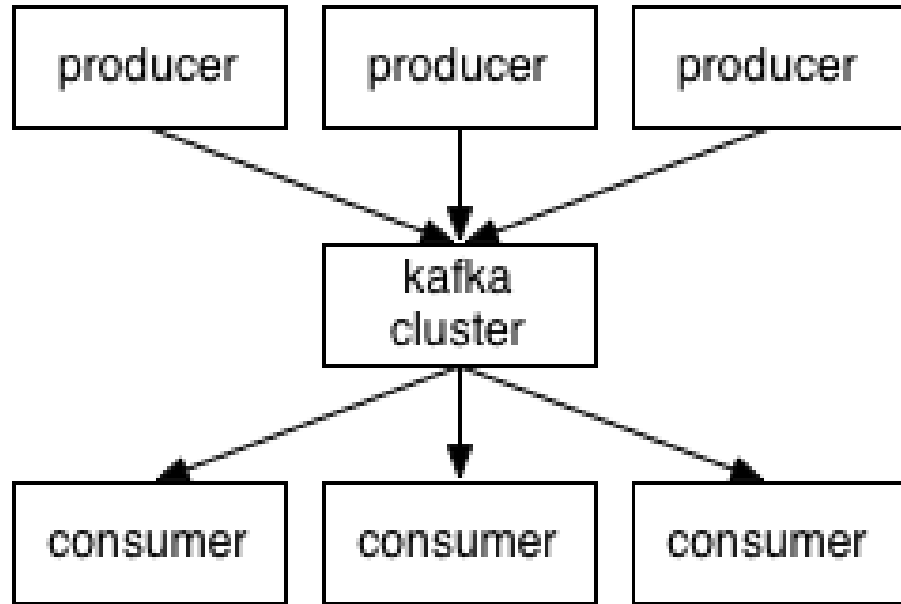


Orderer

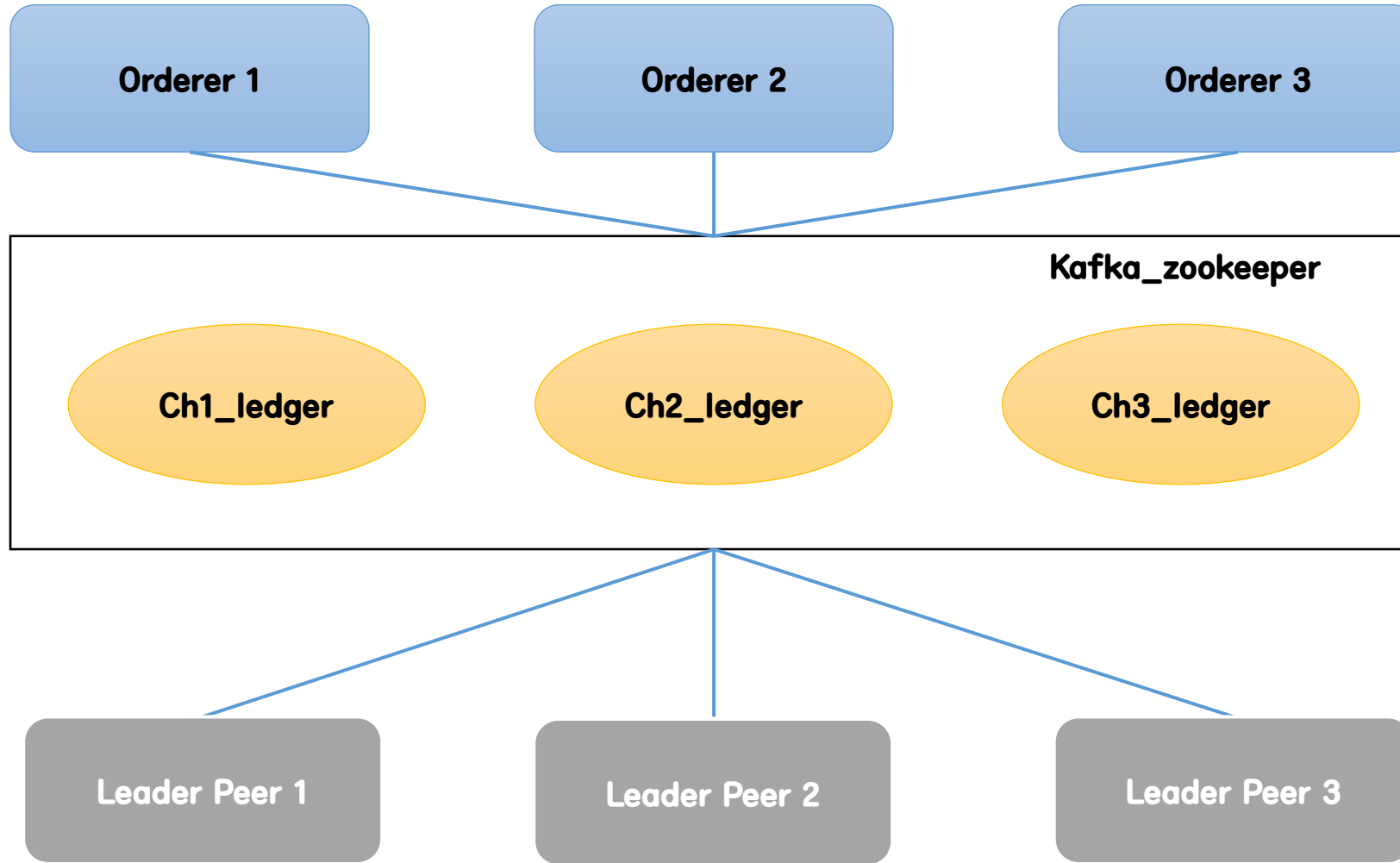
- Fabric 에서 합의를 진행 하는 역할
- 순서대로 정렬하여 최신 블록을 생성하는 역할
- Endorsing peer에게 받은 데이터를 순차적으로 정렬
- 데이터를 정렬 한 뒤 각 Organization의 Leader peer에게 전달

Hyperledger Fabric 구성요소

Ordering 하기 위해서 링크드인에서 개발한 Kafka 시스템을 사용
오픈소스로 공개되어서 Apache 재단에서 관리

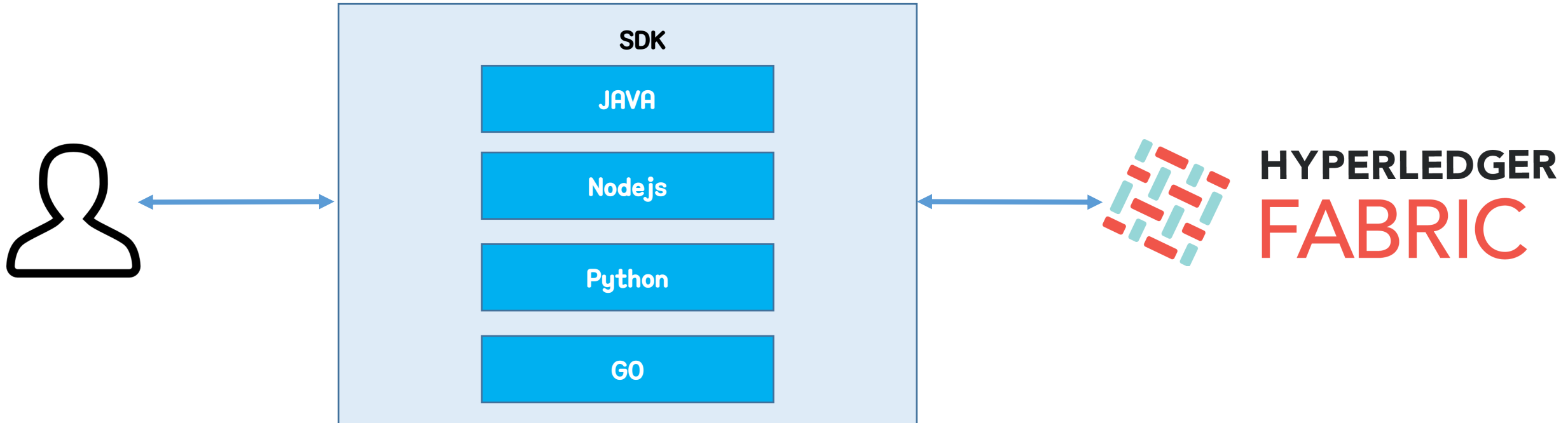


Hyperledger Fabric 구성요소



Hyperledger Fabric 구성요소

User 와 Fabric Network를 연결 시켜 주기 위한 요소



Hyperledger Fabric 구성요소

User 와 Fabric Network를 연결 시켜 주기 위한 요소

The screenshot shows the GitHub repository for `hyperledger/fabric-sdk-java`. It includes the repository name, star/fork counts, and a list of recent commits. The commit list shows files like `.github`, `.settings`, `docs`, `release_notes`, `scripts`, `src`, `.classpath`, `.gitattributes`, and `.gitignore` with their respective commit messages and timestamps.

File	Commit Message	Time
<code>.github</code>	Configure Stale ProBot	8 months ago
<code>.settings</code>	initial commit	3 years ago
<code>docs</code>	FABJ-432 Update README.MD for v2.0	
<code>release_notes</code>	FAB-5514 v1.0.1 release	
<code>scripts</code>	FAB-4574 add missing license header	
<code>src</code>	FABJ-469: Remove getChaincodeIDPath from BlockInfo	
<code>.classpath</code>	FAB-4574 add missing license header	
<code>.gitattributes</code>	initial commit	
<code>.gitignore</code>	[FAB-4363] Increase fabric-sdk-java code coverage	

Hyperledger Fabric SDK for node.js Modules ▾ Classes ▾ Interfaces ▾ Tutorials ▾ Global ▾

The Hyperledger Fabric SDK for Node.js provides a powerful API to interact with a Hyperledger Fabric blockchain. The SDK is designed to be used in the Node.js JavaScript runtime.

Overview

Hyperledger Fabric is the operating system of an enterprise-strength permissioned blockchain network. For a high-level overview of the fabric, visit <http://hyperledger-fabric.readthedocs.io/en/latest/>.

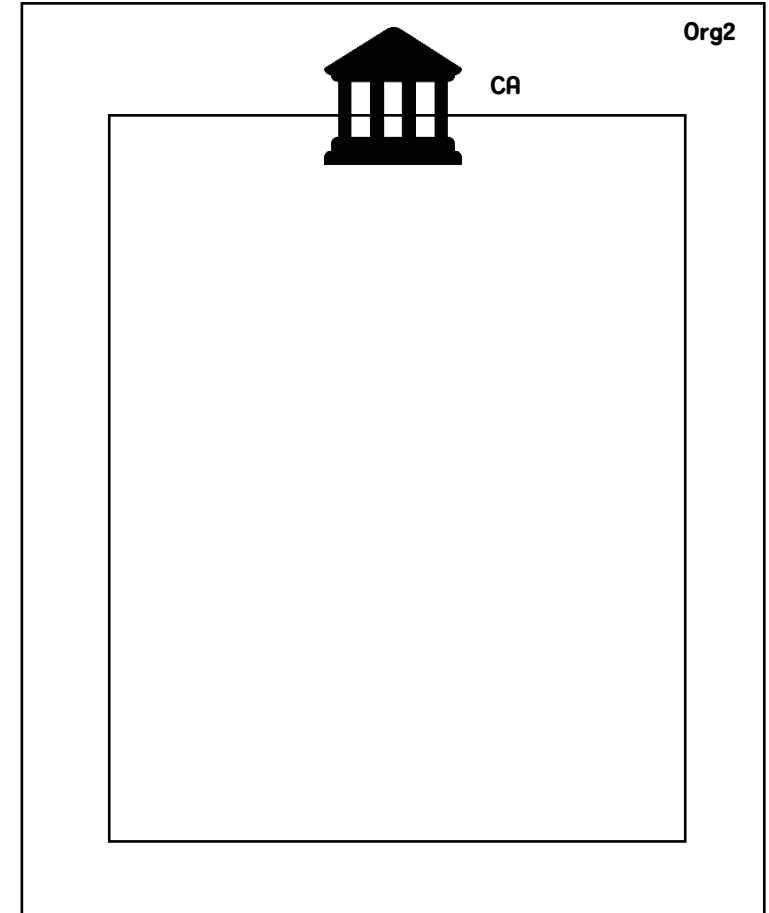
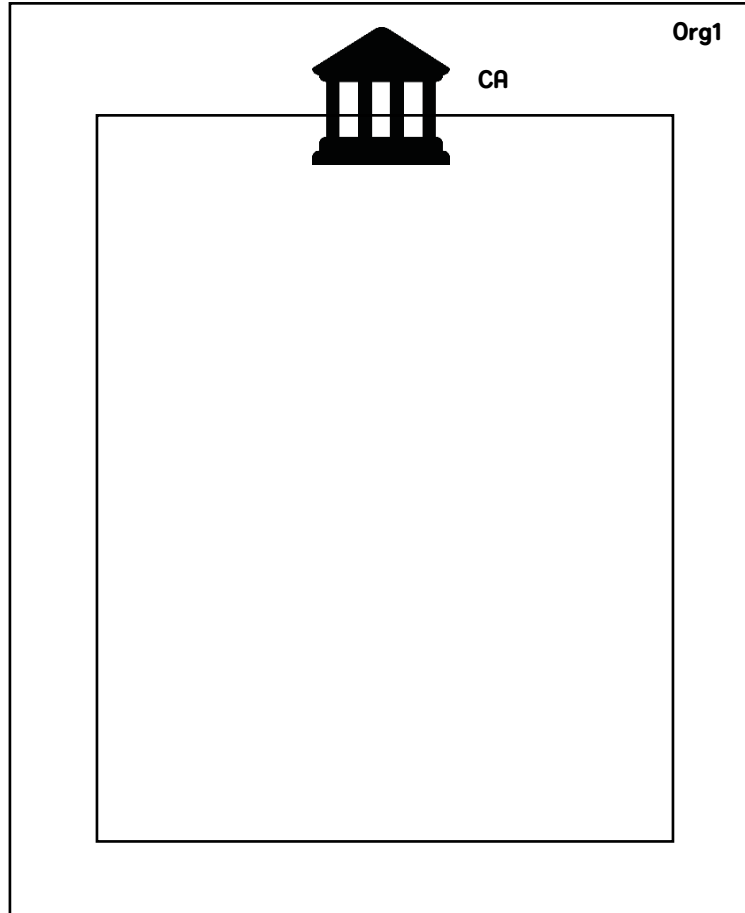
Applications can be developed to interact with the blockchain network on behalf of the users. APIs are available to:

The screenshot shows the GitHub repository for `hyperledger/fabric-sdk-go`. It includes the repository name, star/fork counts, and a list of recent commits. The commit list shows files like `.github`, `internal/github.com/hyperledger`, `pkg`, `scripts`, `test`, `third_party/github.com/hyperledger/fabric`, `.gitignore`, `.gitreview`, and `CHANGELOG.md` with their respective commit messages and timestamps.

File	Commit Message	Time
<code>.github</code>	Configure Stale ProBot	8 months ago
<code>internal/github.com/hyperledger</code>	[FABG-902] Update to latest fabric third_party pins	12 days ago
<code>pkg</code>	[FABG-902] Update to latest fabric third_party pins	12 days ago
<code>scripts</code>	[FABG-900] Remove third_party fabric module	11 days ago
<code>test</code>	[FABG-900] Remove third_party fabric module	11 days ago
<code>third_party/github.com/hyperledger/fabric</code>	[FABG-900] Remove third_party fabric module	11 days ago
<code>.gitignore</code>	[FABG-688] Cleanup gitignore	last year
<code>.gitreview</code>	[FAB-5751] Add .gitreview	2 years ago
<code>CHANGELOG.md</code>	[FABG-833] Release v1.0.0-alpha5	5 months ago

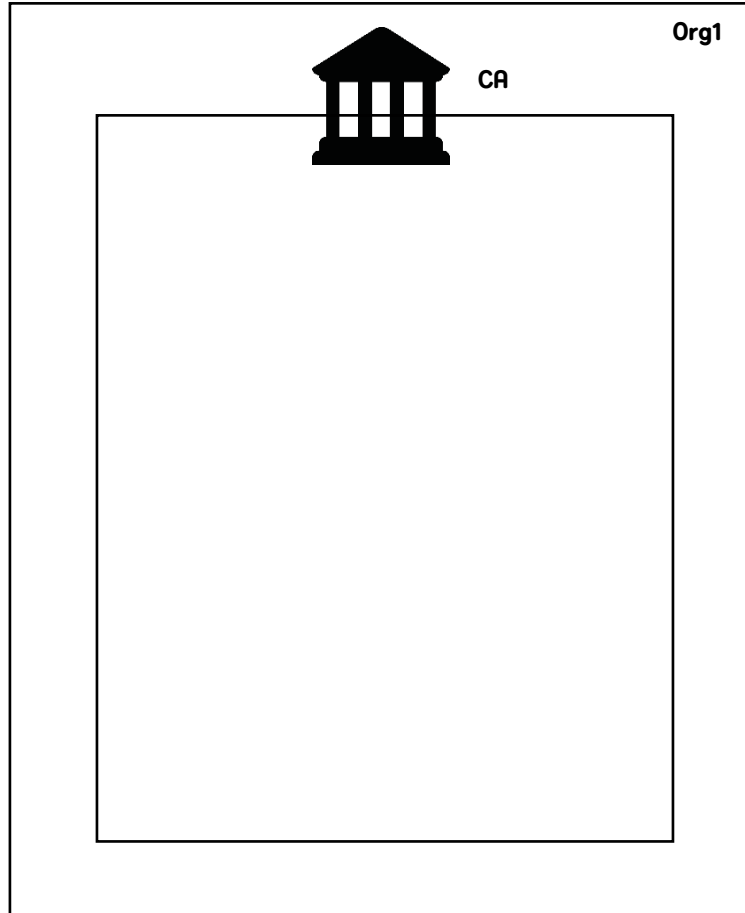
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (1)

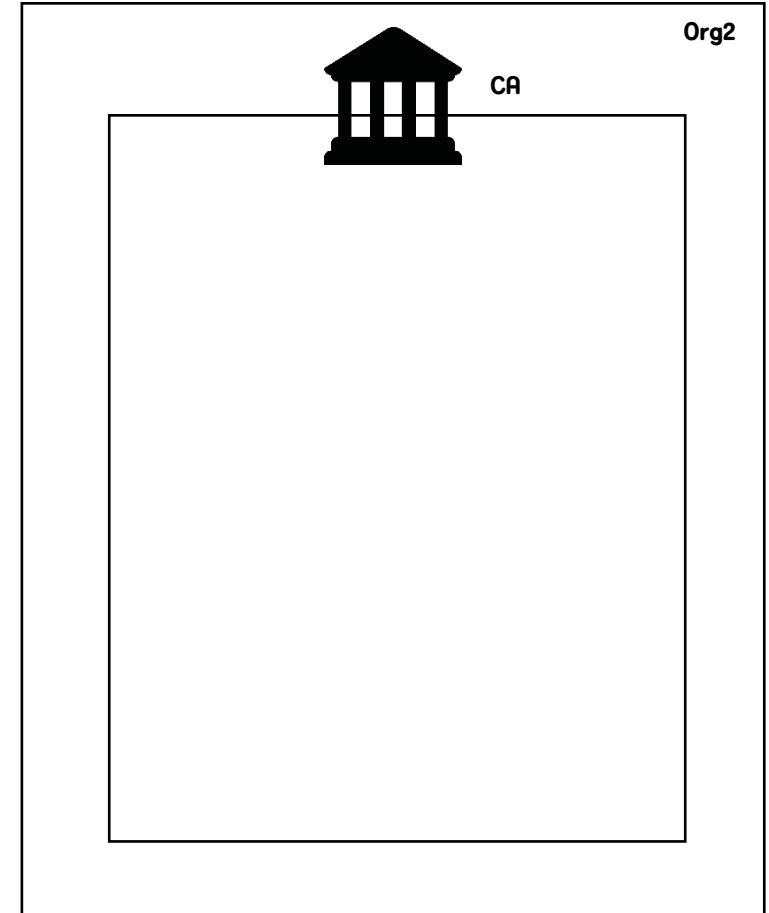


Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (2)

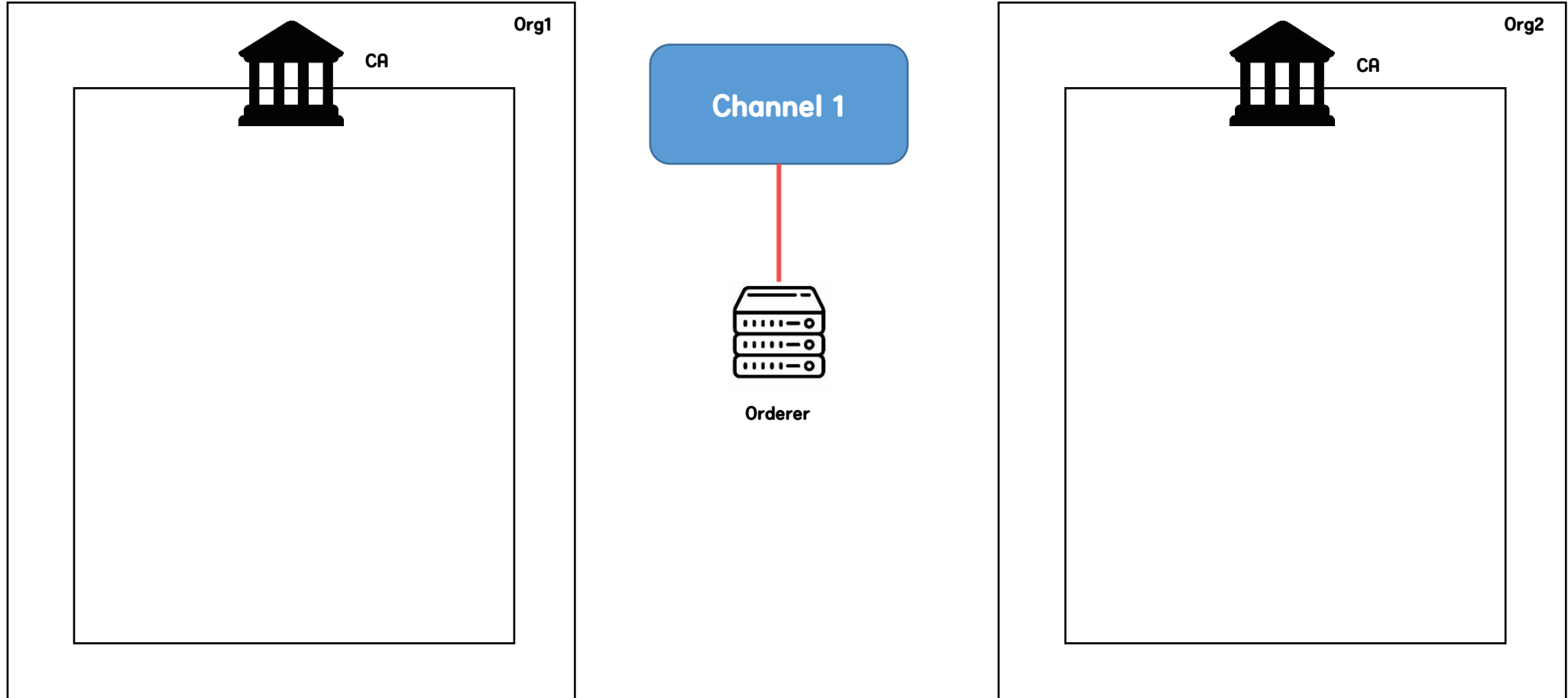


Orderer



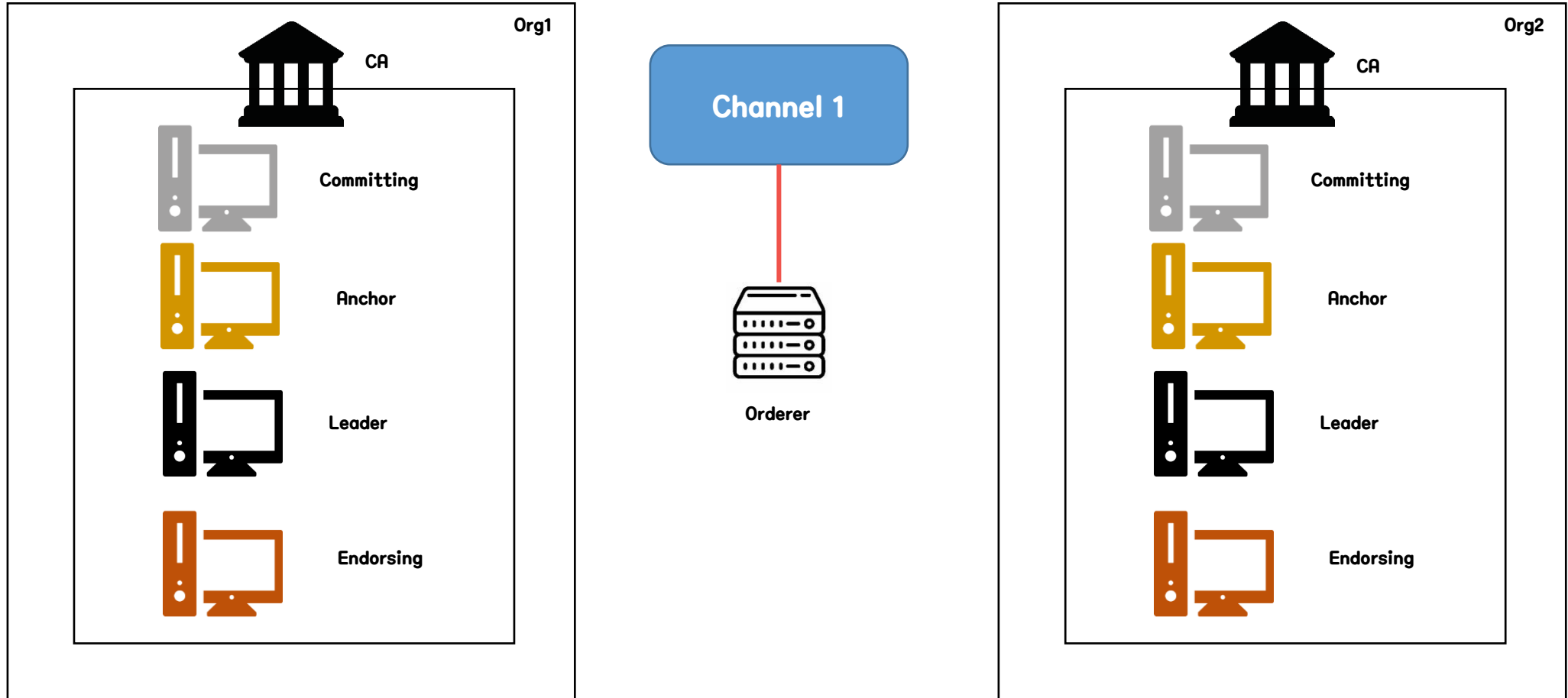
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (3)



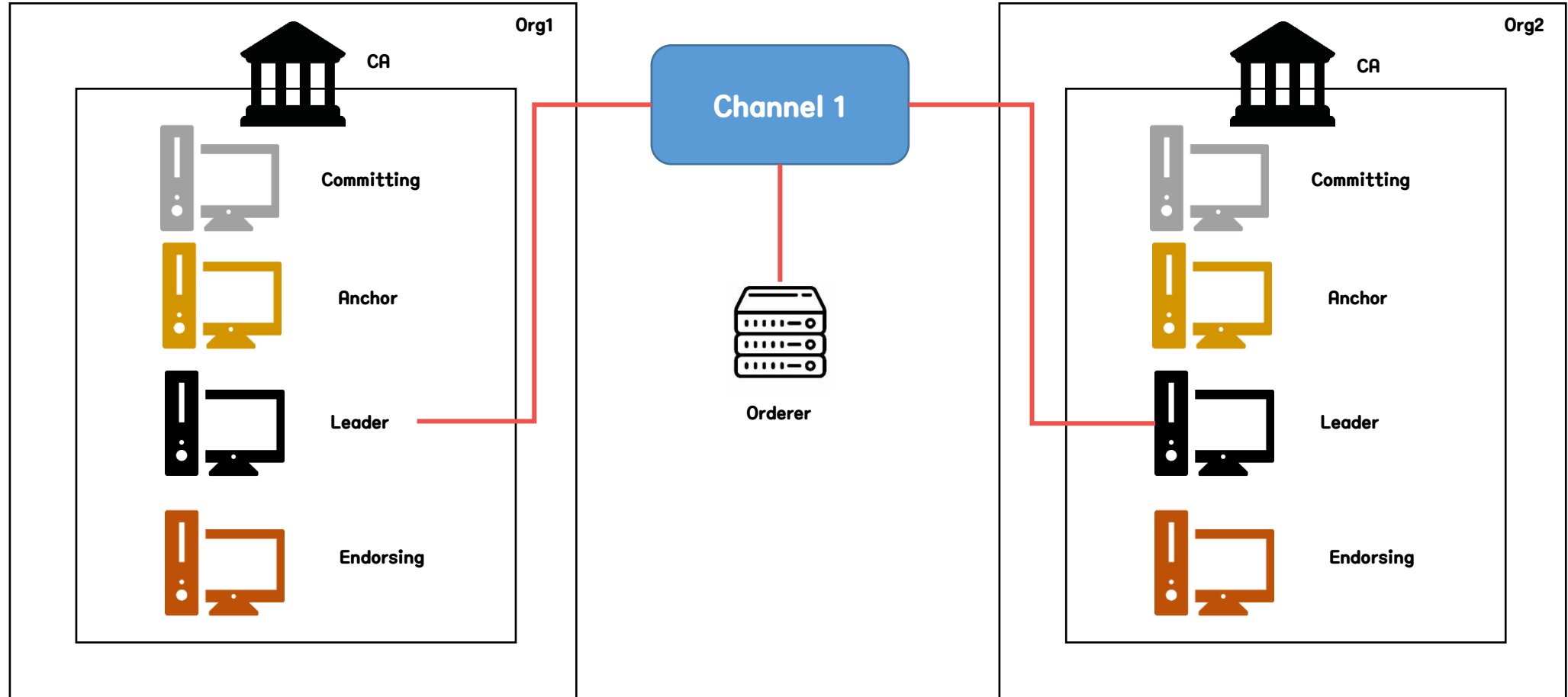
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (4)



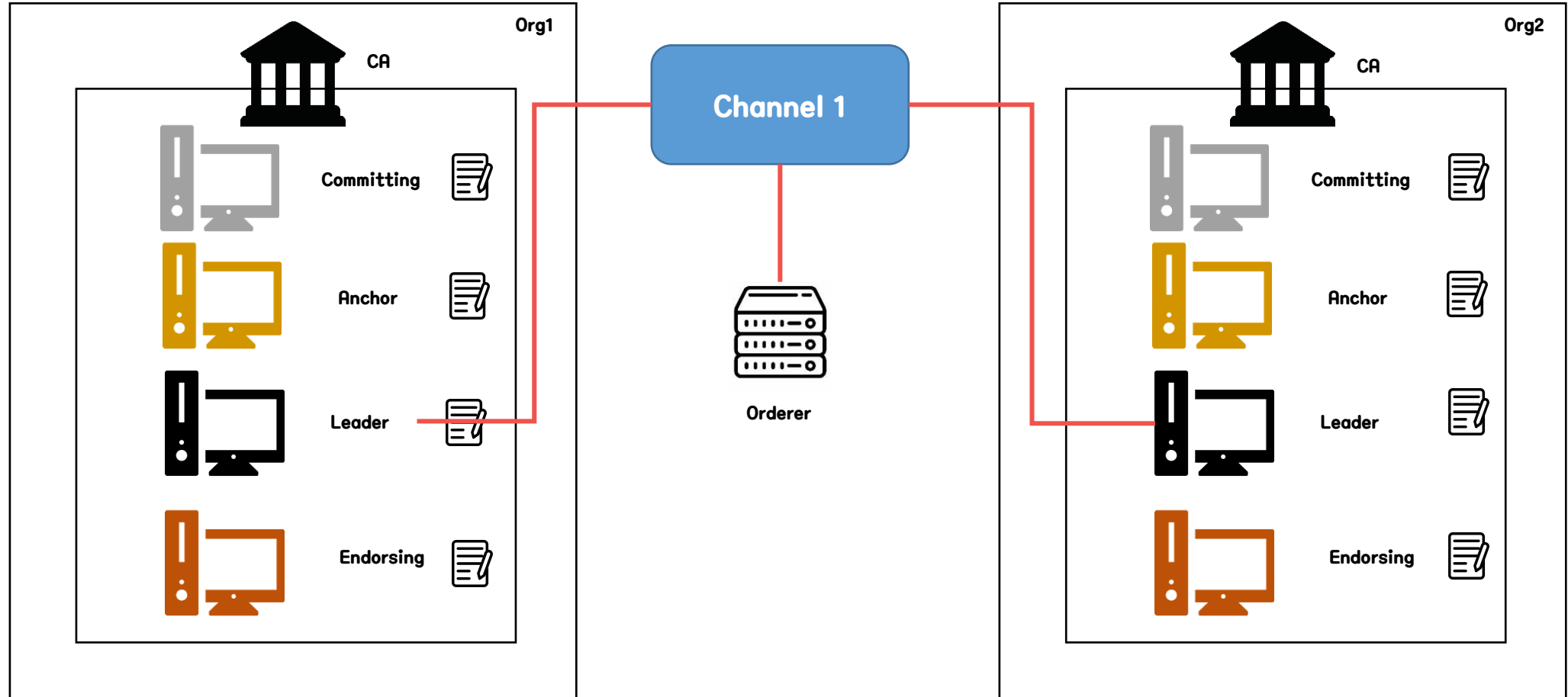
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (4)



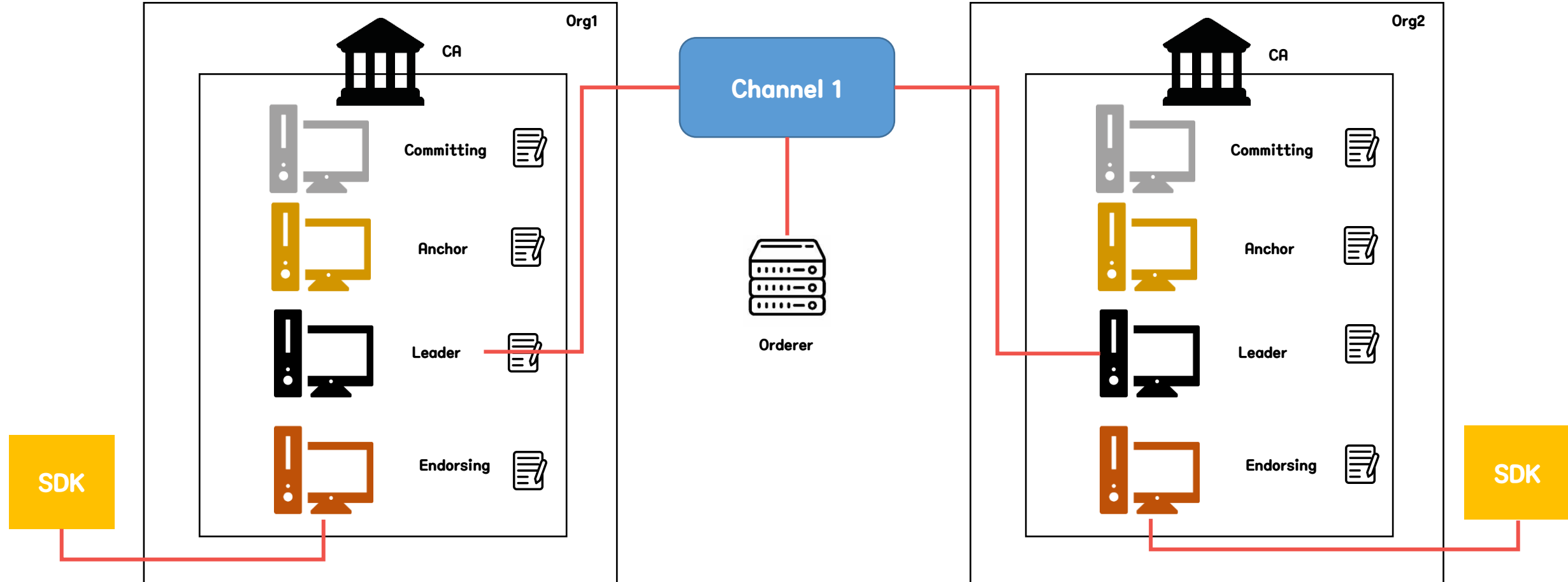
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (5)



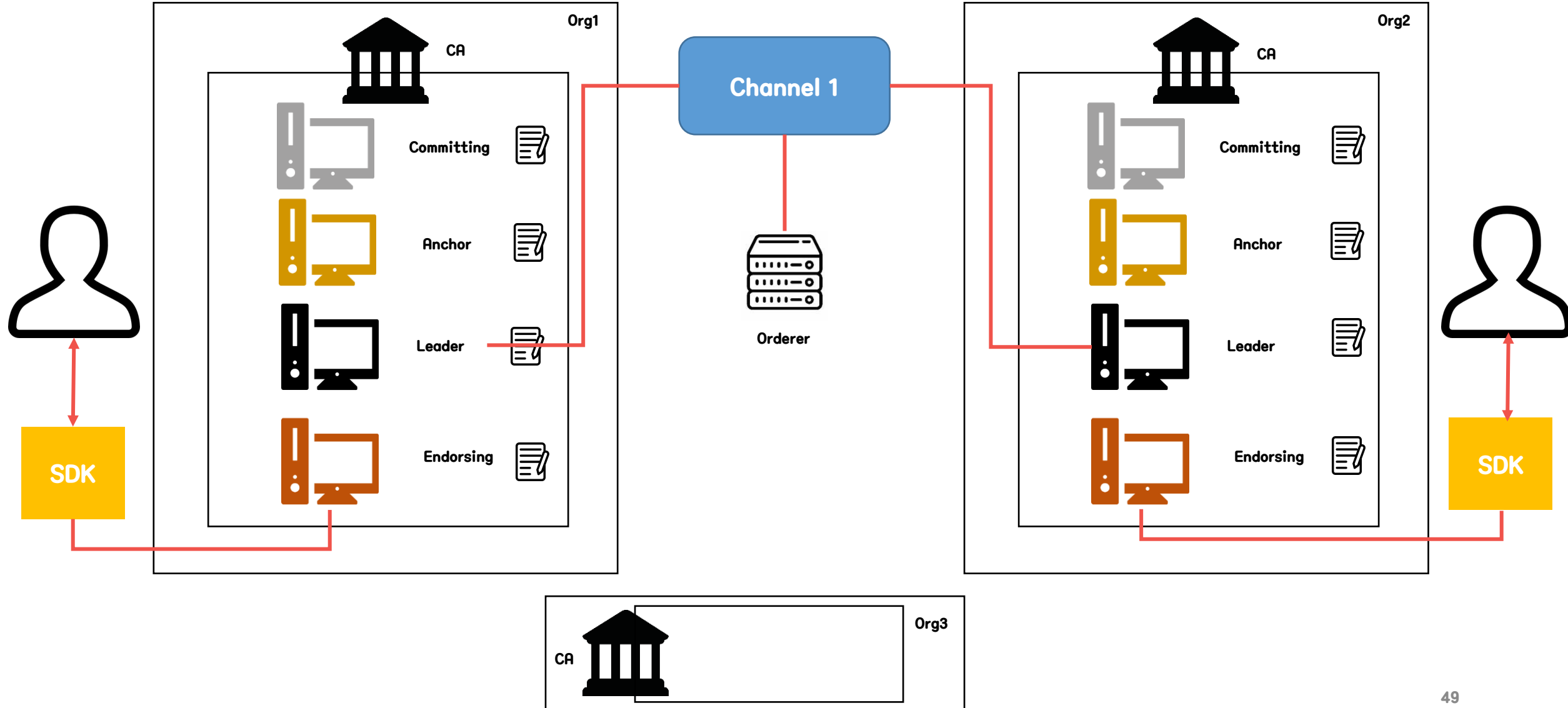
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (6)



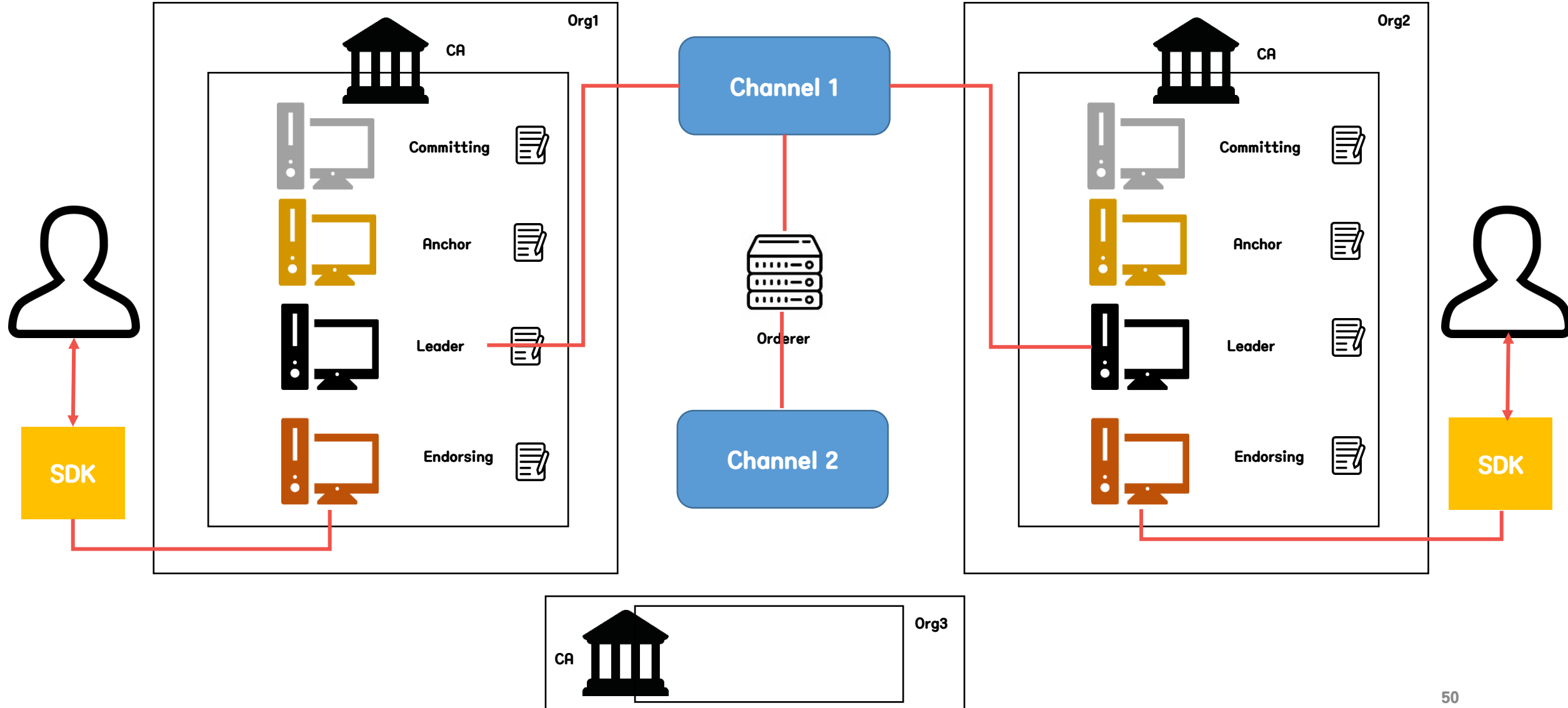
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (7)



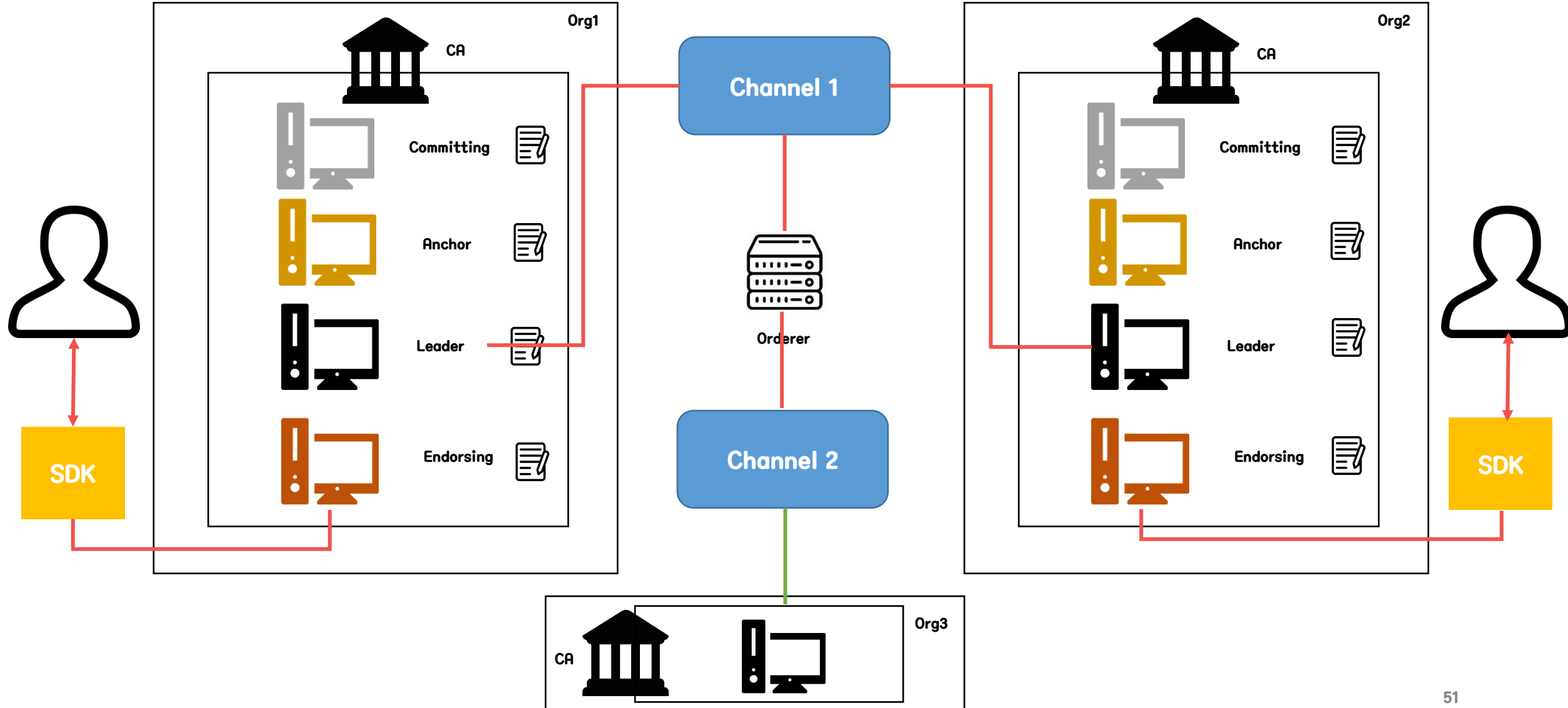
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (8)



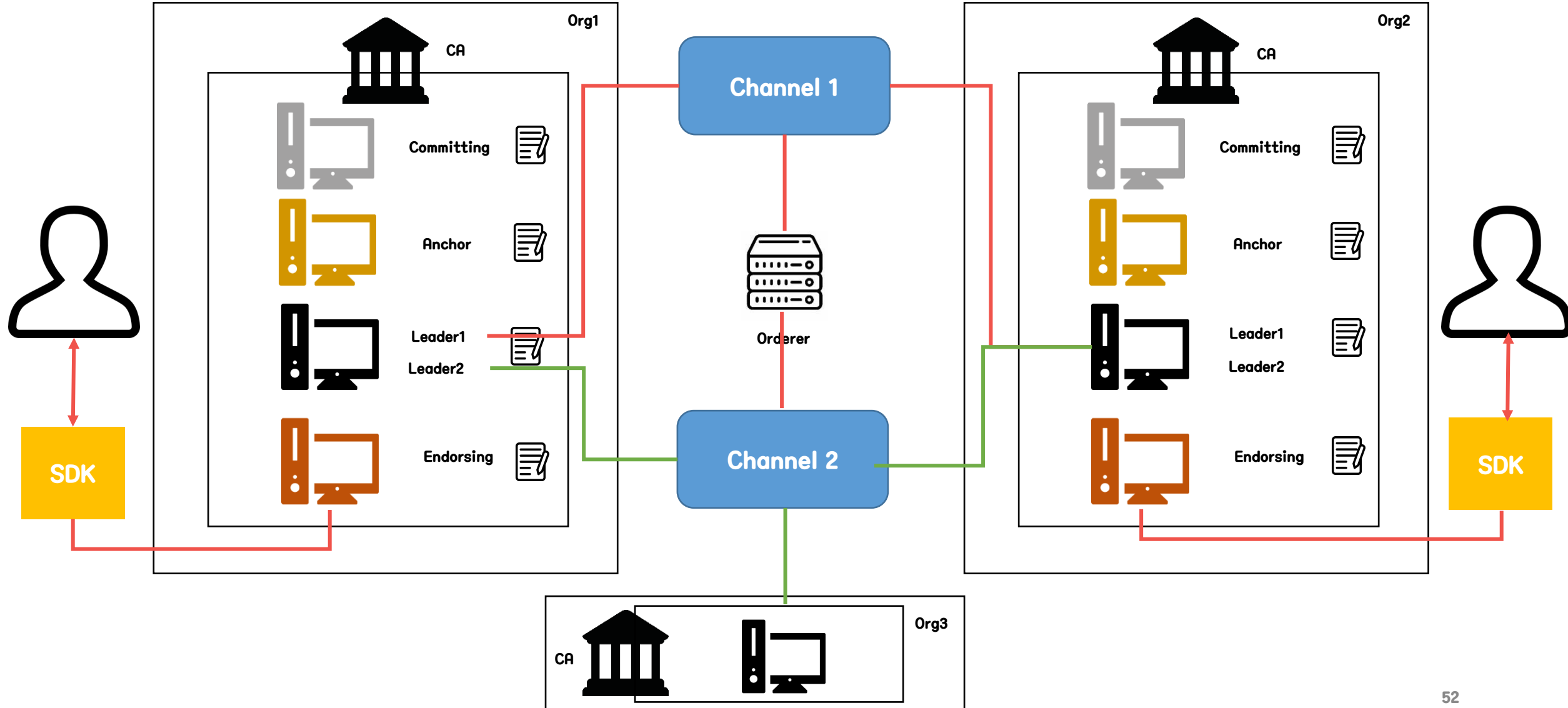
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (9)



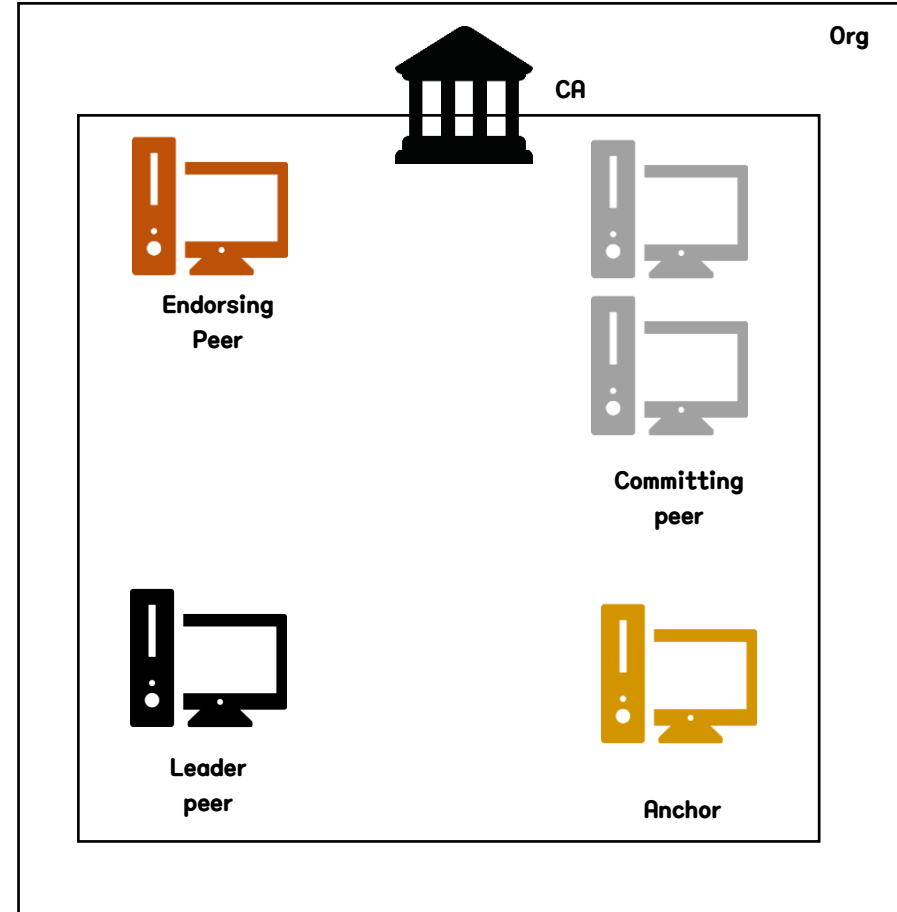
Hyperledger Fabric 네트워크 구축 과정

네트워크 구축 (10)



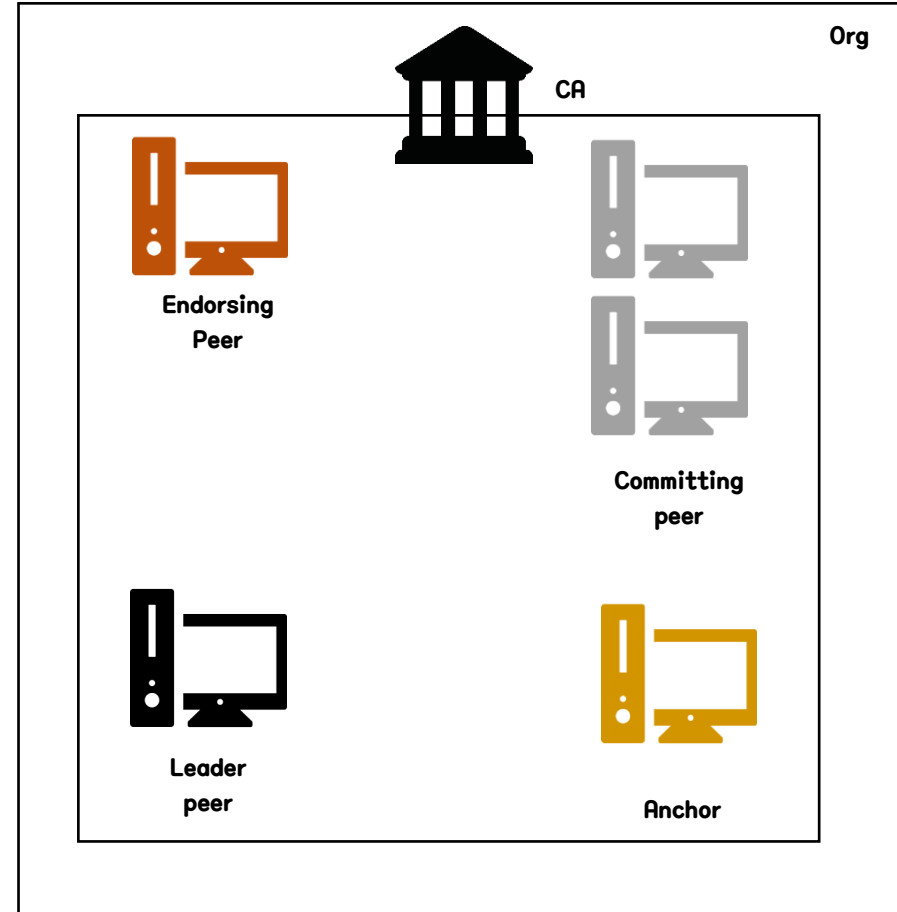
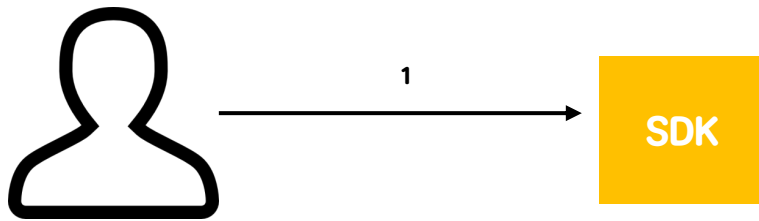
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (1)



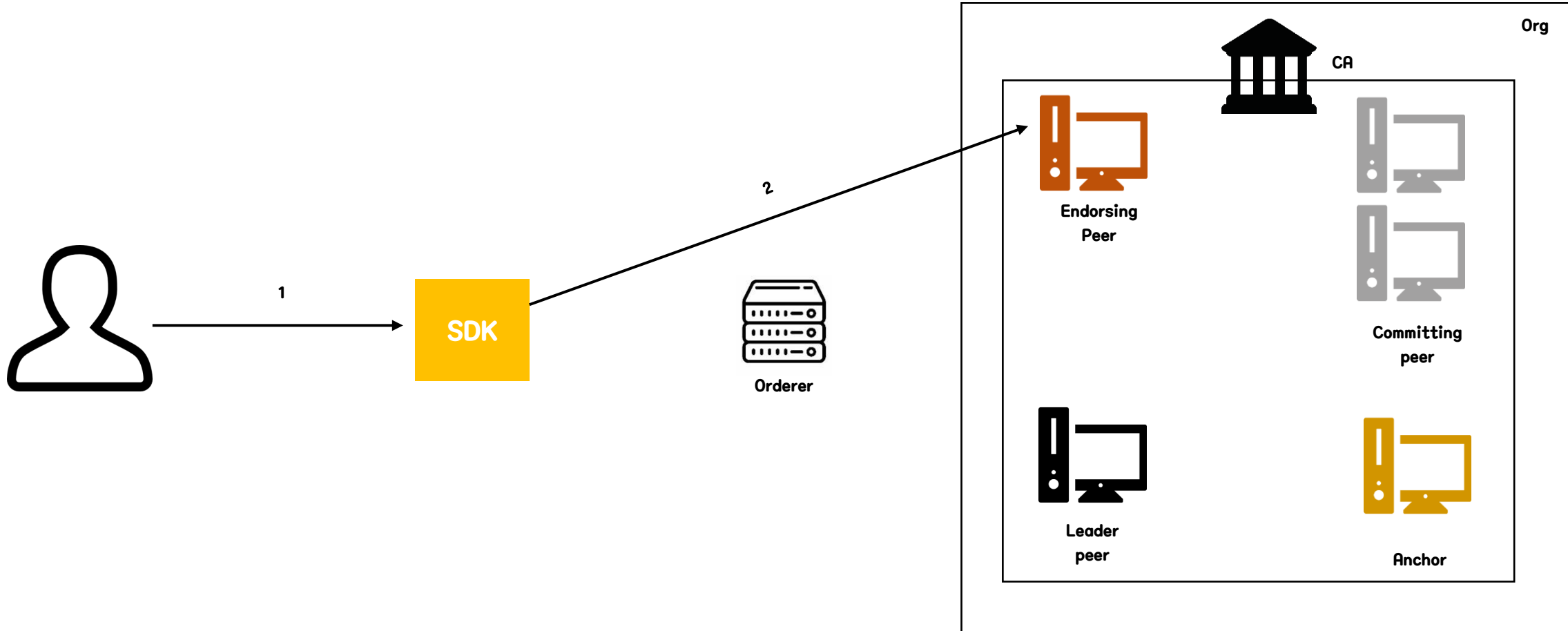
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (2)



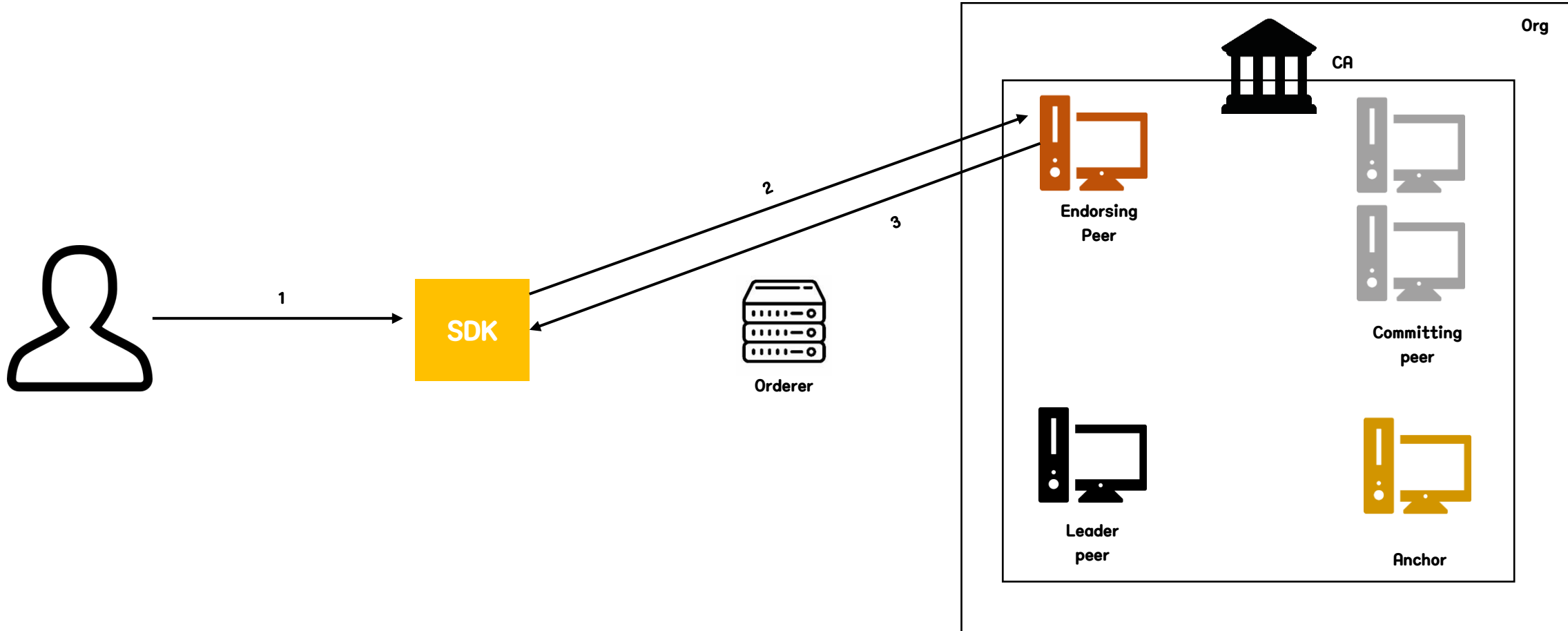
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (3)



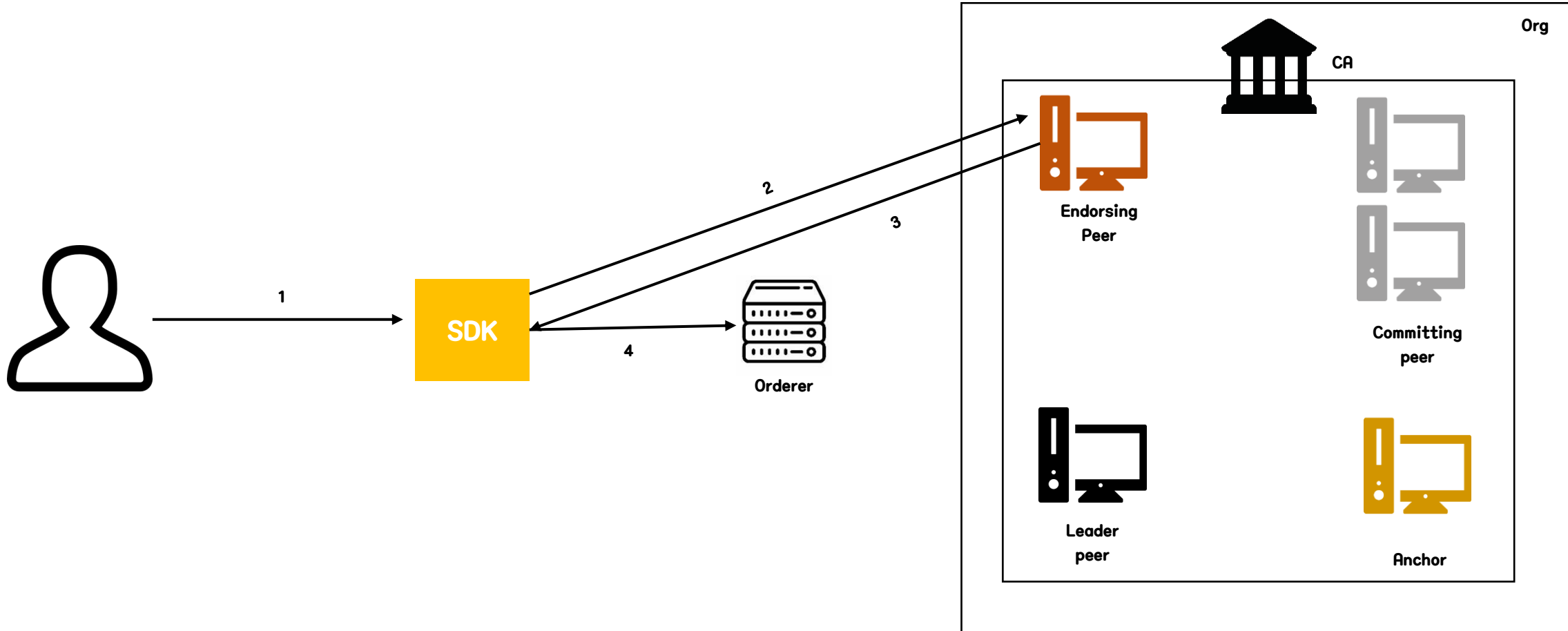
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (4)



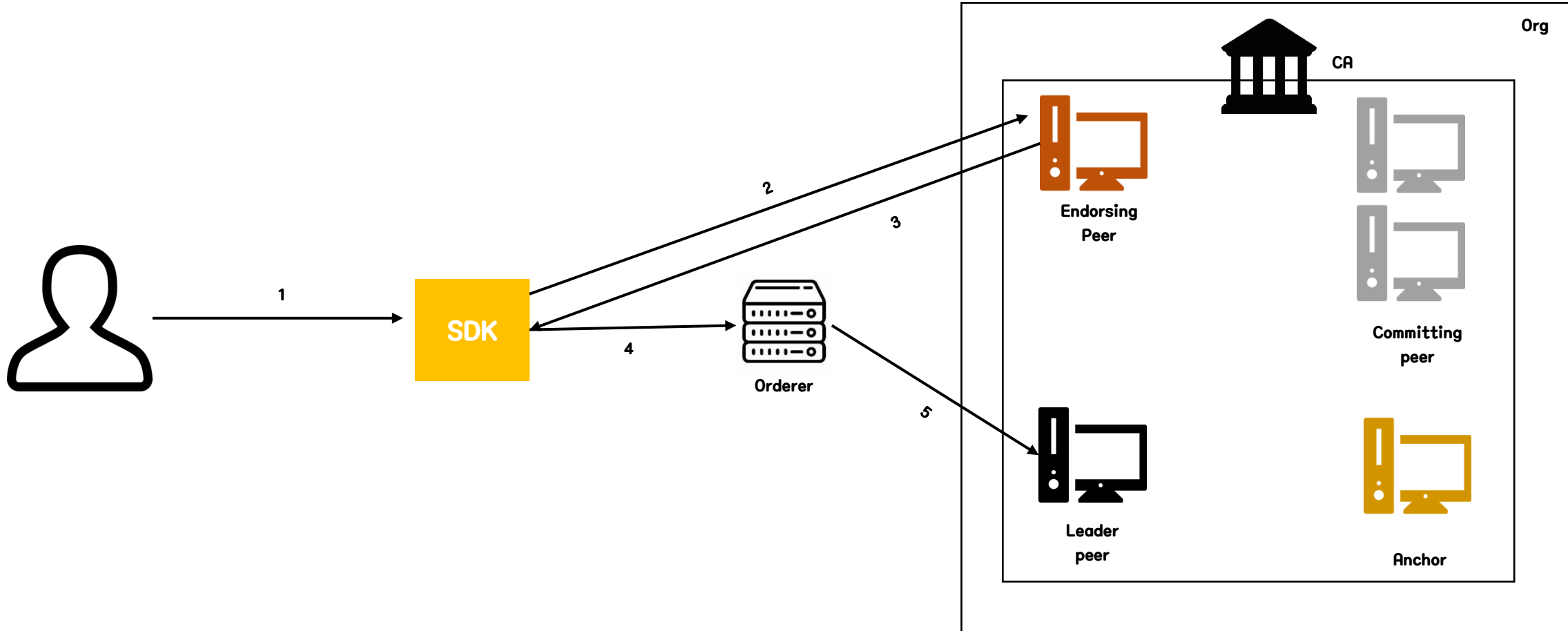
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (5)



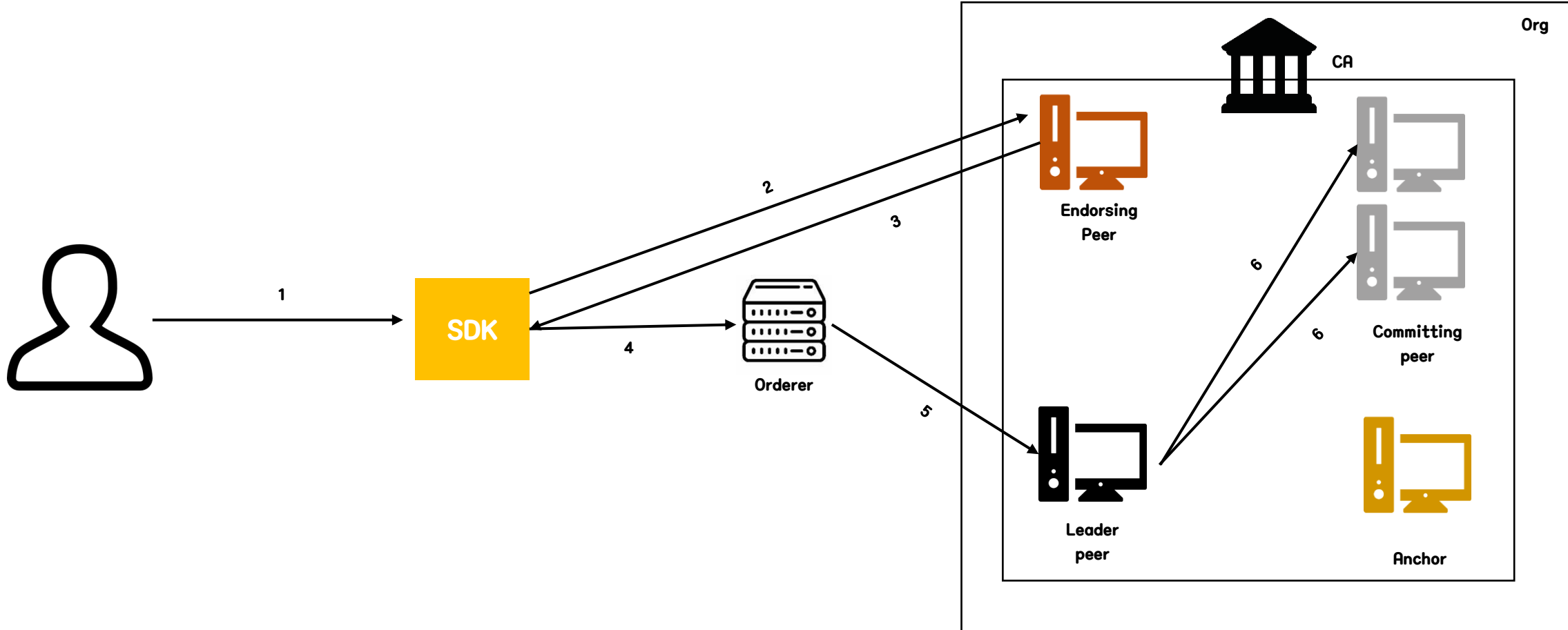
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (6)



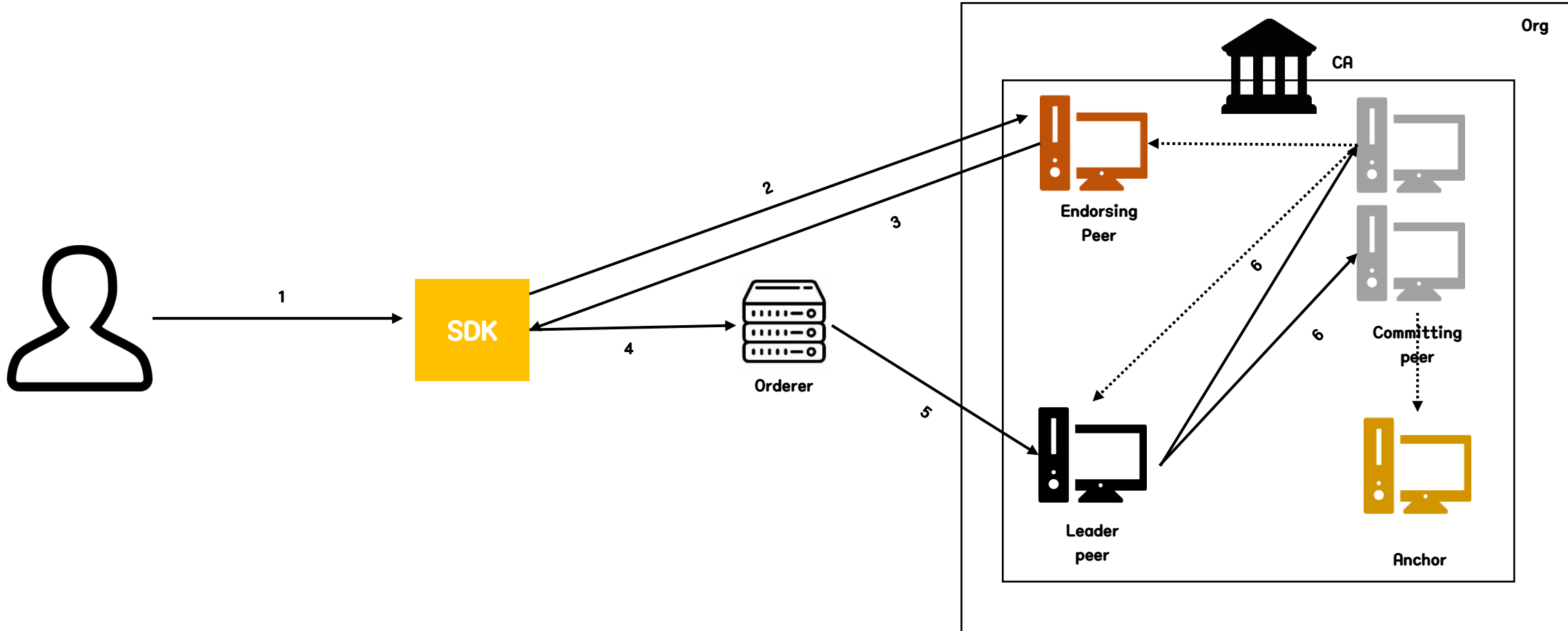
Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (7)



Hyperledger Fabric 트랜잭션 처리 과정

트랜잭션 흐름 (8)



Q & A

