## Quorum이란?

Quorum은 이더리움 기반의 분산원장 프로토콜으로, 이더리움에 프라이빗한 특성을 적용하여 금융서비스 산업에서 트랜잭션과 컨트랙트의 프라이버시를 보장받을 수 있도록 하기 위해 개발되었습니다.

Quorum은 geth라고 흔히 불리는 go-ethereum 클라이언트를 포크하여 최소한의 수정만 하였기 때문에, 이더리움 개발자 커뮤니티에서 구현된 모듈들의 활용이 가능합니다.

퍼블릭 이더리움에서 추가된 Quorum의 대표적 기능은 다음과 같습니다.

* 트랜잭션과 컨트랙트 프라이버시
* 다수의 투표 기반 합의 메커니즘
* 네트워크/개인의 허가 관리
* 향상된 속도

Quorum은 현재 다음과 같은 구성을 가지고 있습니다:

* Quorum 노드 (Geth 클라이언트에서 수정된 버전)
* Constellation/Tessera - 트랜잭션 매니저
* Constellation/Tessera - 엔클레이브(Enclave)

Quorum은 금융 서비스에 적용하는 것을 염두에 두고 설계되었지만 금융 서비스에만 국한되지 않고 상기 명시한 기본 기능을 필요로 하는 이더리움 활용에 관심이 있는 다른 산업들에도 적용 가능합니다.

### 참고 문서

Quorum의 설계와 배경에 대한 자세한 내용은 [**Quorum 백서**](https://github.com/jpmorganchase/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.1.pdf)를 읽거나 [Hyperledger deck](https://drive.google.com/open?id=0B8rVouOzG7cOeHo0M2ZBejZTdGs) 혹은 2016년 9월 22일 하이퍼레저 프로젝트 기술 운영위원회 회의에 제공된 [프레젠테이션](https://drive.google.com/open?id=0B8rVouOzG7cOcDg4UkxqdTBacm8)을 참고하십시오.

## 논리적 아키텍처

[[https://github.com/jpmorganchase/quorum-docs/blob/master/images/Quorum%20Architecture.JPG]]

## 구성요소

### Quorum 노드

Quorum 노드는 지속적으로 성장하고 있는 이더리움 커뮤니티의 연구 및 개발 결과물과 호환성을 이룰 수 있도록 의도적으로 geth에 최소한의 수정만을 하는 형태로 설계되었습니다. 이러한 설계를 바탕으로 Quorum은 geth 릴리스에 맞추어 함께 업데이트할 것입니다.

Quorum 노드는 geth를 다음과 같이 수정하였습니다.

1. 합의는 작업증명방식(PoW) 대신 Raft 또는 이스탄불 BFT(Istanbul BFT) 합의 알고리즘을 사용합니다.   
2. 허가된 노드와의 연결만 허용하도록 P2P 레이어를 수정하였습니다.   
3. 블록 생성 로직에서 ‘global state root’를 ‘global public state root’로 대체하였습니다.   
4. 블록 검증 로직에서 블록 헤더의 ‘global state root’를 ‘global public state root’로 대체하였습니다.  
5. 상태전이 패트리샤 트리 (State Patricia trie)를 퍼블릭 상태 트리와 프라이빗 상태 트리로 분리하였습니다.   
6. 블록 검증 로직에서 ‘프라이빗 트랜잭션’을 처리하도록 수정하였습니다.   
7. 데이터 프라이버시 보호를 위하여 트랜잭션 데이터를 암호화된 해시로 대체하도록 트랜잭션 생성 로직을 수정하였습니다.   
8. 가스의 개념은 남아있지만 가스 가격(gas price)은 제거하였습니다.

### Constellation 및 Tessera

[Constellation](https://github.com/jpmorganchase/constellation)과 [Tessera](https://github.com/jpmorganchase/tessera)는 안전한 방식으로 정보를 전달하기 위한 범용 시스템으로 Haskell 및 Java 로 구현하였습니다. 이 시스템들은 메시지를 PGP(Pretty Good Privacy)로 암호화하는 MTA(Message Transfer Agents, 메시지 전송 에이전트) 네트워크와 비교할 수 있습니다. 해당 시스템들은 블록체인에 국한되는 것은 아니며, 다양한 이해관계자가 존재하는 네트워크에서 개별적으로 암호화된 메시지 교환을 원하는 다양한 애플리케이션에 적용할 수 있습니다. Constellation과 Tessera 모듈은 다음 두 개의 하위 모듈로 구성되어 있습니다.   
\*노드(Quorum의 PrivateTransactionManager에서 사용)  
 \*엔클레이브(enclave)

#### 트랜잭션 매니저

Quorum의 트랜잭션 매니저는 트랜잭션의 프라이버시를 담당합니다. 트랜잭션 매니저는 암호화된 트랜잭션 데이터의 저장 및 접근 제어, 다른 트랜잭션 매니저와 암호화된 페이로드 교환을 수행하지만 트랜잭션을 암/복호화하는 프라이빗 키에는 접근할 수 없습니다. 또한 트랜잭션 매니저는 암호화 기능을 위해 엔클레이브를 사용합니다. (엔클레이브가 선택적으로 트랜잭션 매니저 자체에서 호스팅되는 것도 가능합니다.)

트랜잭션 매니저는 restful/stateless 서비스로 쉽게 부하 분산이 가능합니다.

트랜잭션 매니저가 엔클레이브와 어떻게 상호작용하는지에 대한 더욱 자세한 내용은 [[트랜잭션 처리 및 프라이버시|트랜잭션 처리]]를 참조십시오.

#### 엔클레이브 (Enclave)

분산원장 프로토콜은 일반적으로 트랜잭션 유효성 검증, 참여자 인증, 과거 데이터 보존 (즉, 암호화된 해시 데이터 체인 사용)을 위해 암호화 기술을 활용합니다. 이러한 암호화 기술을 적용함에 있어서 성능향상 및 모듈간 명확한 역할 분담을 위하여 암호화 기능들의 병렬 처리 및 대칭 키 생성, 데이터 암/복호화와 같은 대부분의 암호화 작업은 엔클레이브에서 처리합니다.

엔클레이브는 트랜잭션 매니저와 상호 통신하며 트랜잭션 데이터를 처리합니다. 이때 트랜잭션 매니저와는 분리된 방식으로 암/복호화를 관리함으로써 트랜잭션의 프라이버시를 강화합니다. 엔클레이브는 다른 구성 요소들과 격리된 “가상 HSM”로 프라이빗 키를 보관하고 있습니다.

엔클레이브에 대한 더욱 상세한 내용은 [[트랜잭션 처리 및 프라이버시|트랜잭션 처리]]를 참조하십시오.