블록체인 서베이

https://youtu.be/NU5yA8u7jRU

작업증명, 지분증명, PBFT, Sharding

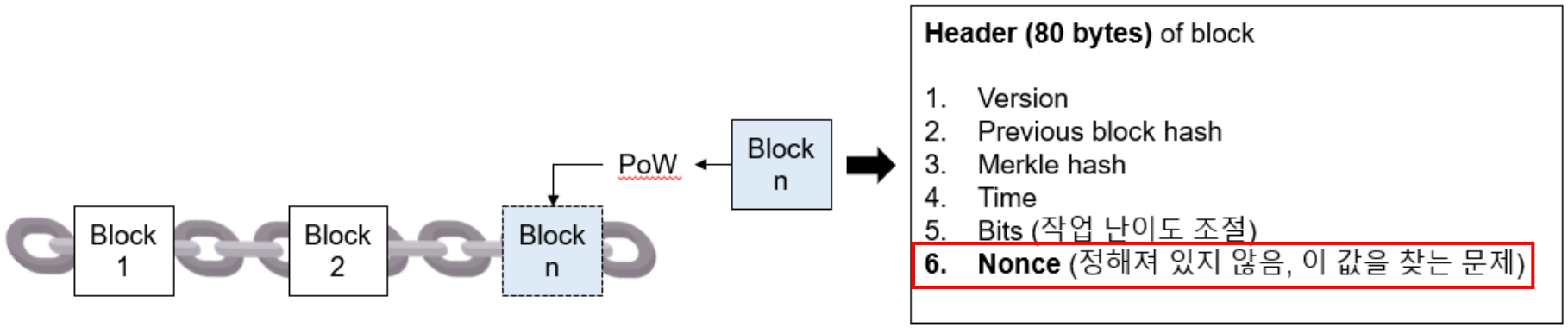
Layer1

Layer2

ZK-STARK

작업 증명(Proof of Work, PoW)

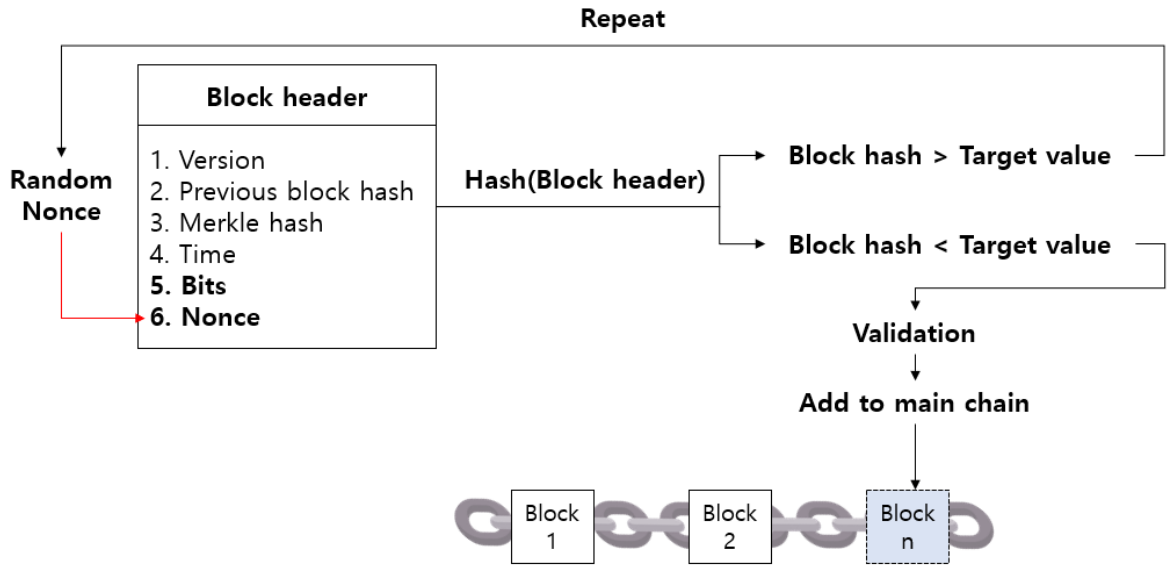
- **PoW란?**
  - 블록체인의 보안성 확보를 위해 블록의 **특정 해시값**을 찾는 합의 알고리즘
  - 블록 헤더에 속하는 6개의 정보 중, 특정 조건을 만족하는 유효한 **Nonce** 값을 찾는 문제 → 채굴
  - 블록 헤더의 Bits라는 요소로 **난이도 조절** 가능 → 블록 해시가 **Bits** 보다 작아야 함
  - 채굴자가 유효한 Nonce 값을 찾은 후에 다른 채굴자들이 해당 Nonce 값이 유효한지 **검증**
    → 이러한 검증 과정을 거친 후, 블록이 메인 체인에 추가되고 채굴자는 보상을 받음



작업 증명(Proof of Work, PoW)

- **동작 방식**

1. **nonce 값**을 변화시키며 해시 함수에 대입
2. 목표 값보다 **큰 값**이 나오면 1번의 과정을 다시 수행
   목표 값보다 **작은 값**이 나오면 해당 nonce를 적용했을 때의 해시 값이 해당 블록의 해시 값이 됨
3. 다른 노드들은 해당 nonce를 **검증**
   → 블록 헤더에 해당 nonce 값을 입력하여 해시하고, 목표 값보다 작은 해시 값이 나오면 검증 성공
4. 검증된 블록은 **메인 체인에 추가**되고 채굴자는 보상을 받음



작업 증명(Proof of Work, PoW)

- **작업 증명의 장점**
  - 모든 노드에게 검증을 받아야하므로 거래 내역을 속이기가 어려움

- **작업 증명의 단점**
  - 채굴 난이도가 높아지면서 작업 증명을 위한 연산에 **높은 컴퓨팅 파워**를 필요로 함
    → 고사양 장비가 필요하며, 과도한 전력 소모로 에너지 낭비 발생
  - 작은 블록체인 네트워크일 경우, 높은 컴퓨팅 파워를 가진 채굴자들이 담합(Mining pool)할 경우
    → **51% 이상의 지분을 차지**하여 탈중앙화 기반의 검증시스템이 무력화 되어 기존 거래 내역에 대한 **위변조 발생 가능**
  - **속도가 느림**
    →노드 수가 늘어나면서 트랜잭션 처리 속도가 느려짐 (**확장성 저하)**

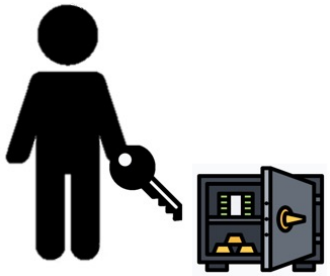
작업 증명(Proof of Work, PoW)

- **PoW를 기반으로 하는 알고리즘**
  - 이중 작업증명
  - 경과 시간증명
  - 스펙터
  - 지연 작업증명
  - 균형 작업증명

지분 증명(Proof of Stake, PoS)

- PoW의 문제점을 해결하기 위해 도입된 알고리즘
  → 컴퓨팅 파워가 아닌 네트워크 참여자가 가진 지분(Stake)에 비례하여 블록 생성 권한을 위임하는 방식
- Staking이란?
  → 사용자가 가진 암호화폐의 일정량을 블록체인 네트워크 운영을 위해 활용될 수 있도록 보증금으로 맡기고 그에 대한 대가로 보상을 받음
- 가장 많은 암호화폐 소유자가 블록 생성 권한을 독점하는 것을 막기 위해 **무작위 시스템**을 통해 다음 블록 생성자를 결정

PoW (작업증명)



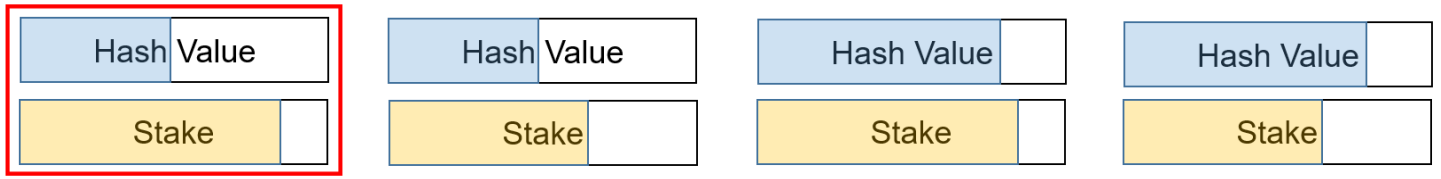
PoS (지분증명)

지분 증명(Proof of Stake, PoS)

- **무작위 시스템의 종류**

1. 무작위 블록 선택

- 가장 낮은 해시 값과 가장 높은 지분(Stake)의 조합을 가진 노드를 검증자로 선택
- 지분의 크기는 모두에게 공개되어 있으므로 노드들은 일반적으로 다음 검증자를 예측 가능



2. 코인 나이에 따른 검증자 선택

- 블록 생성에 참여하고자 하는 노드들이 네트워크 상에 일정량의 코인을 자신의 지분으로 staking
- 코인의 나이 = 코인이 stake 된 일 수 x 코인의 수
- 만약 노드가 블록을 생성한다면 코인의 나이는 다시 0으로 초기화
- 따라서 다시 블록을 생성하기 위해서는 코인의 나이가 많아질 때까지 일정 시간이 소요

→ 가장 많은 지분을 차지한 노드가 네트워크를 지배하는 것을 방지

지분 증명(Proof of Stake, PoS)

- **PoS의 장점**
  - 모든 노드에게 검증을 받지 않아도 되므로 PoW보다 **거래 처리 속도가 빠르고 에너지 소비가 적음**
  - PoW에 비해 **탈중앙화에 유리**
    → PoW의 경우 자본이 100배면 100배 이상의 컴퓨팅 파워를 가질 수 있으나, PoS의 경우 자본이 100배면 100배 이상의 지분을 가질 수 없어서 분권화에 유리
- **PoS의 단점**
  - 전체 토큰의 **51%**를 한 명의 참여자가 소유할 경우 **중앙화**
  - 다수의 코인을 보유해 staking 할수록 권한은 커지고, 검증 보상으로 코인을 받는 것이 반복됨
    → **빈익빈 부익부**
  - **Nothing at Stake :** 체인에 포크가 발생할 때, 지분을 가진 참여자들이 양쪽 체인에 투표하여 블록 체인의 정당성을 해치는 상황
    - 자신의 지분 증명을 위한 한계 비용이 없으며, 두 체인에 투표해도 손해보는 것이 없으므로 발생함
    - 양쪽 체인 모두에서 지분을 가질 수 있게 되어 고의로 체인 포크를 노리는 참여자 발생 가능
    →이를 해결하기 위해서는 지분 증명 시 **보증금**을 내고, 잘못될 경우 보증금의 일부를 잃도록 함

지분 증명(Proof of Stake, PoS)

- **PoS를 기반으로 하는 알고리즘**
  - 리스지분증명
  - 하이퍼 위임증명
  - 위임지분증명
  - 마스터 노드 지분증명
  - 포크능력증명

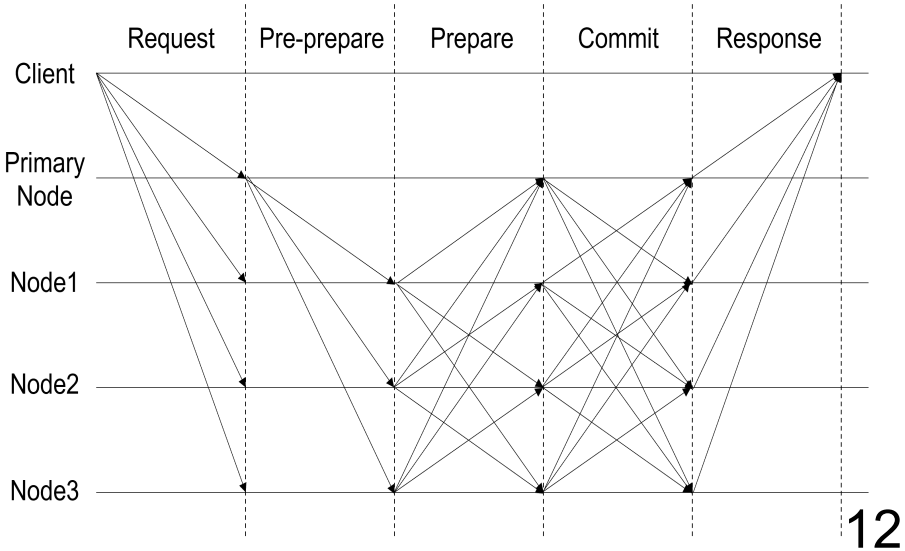
Practical Byzantine Fault Tolerance, PBFT

- 비잔틴 장군 문제를 해결하기 위해 제시
- **악의적인 노드가 네트워크 내에 존재하여도 합의를 도출**할 수 있는 알고리즘
- 네트워크 내에 악의적인 노드가 f개일 경우 총 노드의 개수 (n)가 $n = 3f + 1$개 이상일 경우 성공적인 합의가 이루어질 수 있다는 이론에 근거

Practical Byzantine Fault Tolerance, PBFT

- **PBFT 알고리즘의 동작 방식**
  - Pre-prepare, Prepare, Commit 단계로 구성

1. 클라이언트가 Primary 노드에게 **상태 변환을 요청하는 메시지 전송**
2. Primary 노드는 클라이언트의 **요청을 정렬** → 요청에 대한 결과**(pre-prepare 메시지)를 기입 → 브로드캐스트**
3. 백업 노드들은 수신한 **pre-prepare 메시지를 브로드캐스트**함과 동시에 다른 노드들로부터 $2f$**개의 메시지 수집** (prepared certificate)
4. 다른 노드들로부터 **가장 많이 받은 동일한 메시지 확인** → 해당 메시지를 다른 노드들에게 **브로드캐스트**
5. 이때, 다른 노드들로부터 $2f+1$ 개의 메시지를 수집한 경우를 **committed certificate**라고 함
6. 클라이언트에게 **Reply 메시지 전송**
7. 최종적으로 **모든 노드가 합의를 이룬 같은 메시지를 가질 수 있음**



Practical Byzantine Fault Tolerance, PBFT

- **신뢰도가 높은 알고리즘**
  - 다른 노드들이 Primary 노드의 행동을 보고 악의적인 행동을 한다고 판단된다면,
    다수결을 통해 Primary 노드 교체 가능

- 악의적인 노드가 $f = \frac{n-1}{3}$보다 많을 경우에는 **정상적인 합의 불가**

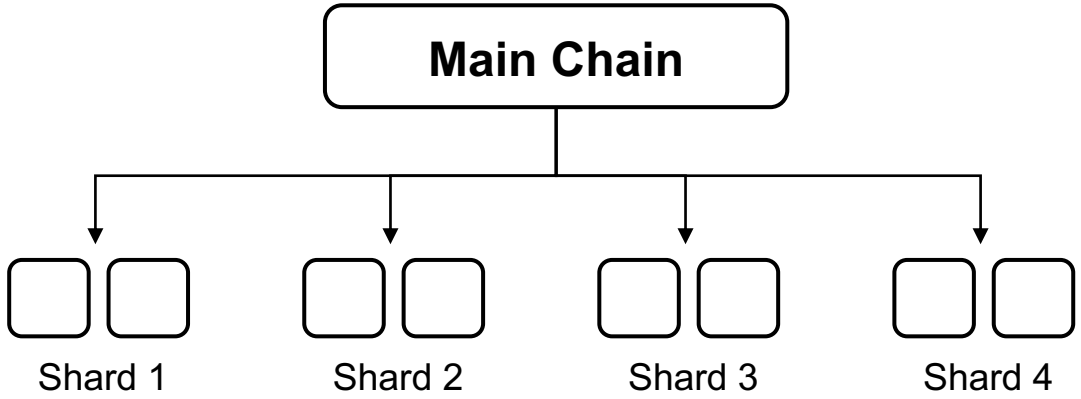
- **통신비용 증가, 확장성 저하**
  - 모든 노드에 대해 2번씩 발생하는 브로드캐스트로 인해

Sharding

- **Sharding이란?**
  - 데이터베이스에서의 샤딩

    → 대용량의 데이터를 처리하기 위해 테이블을 수평 분할하여 데이터를 분산하여 저장하고 처리함으로써 속도를 높이는 방법
  - 블록체인에서의 샤딩

    → 전체 네트워크를 여러 개의 네트워크로 분할함으로써 병렬로 처리해 네트워크 속도를 높이는 방법



Sharding

- **Sharding의 요소**

  - **Proposer**

    - 샤드내에서의 트랜잭션들을 모아서 Collator에게 Proposal을 전달

    (Proposal은 아직 검증되지 않은 Collation = Collation shard에서의 Block과 같은 개념)

  - Collator

    - Proposer로부터 전달받은 Proposal을 검증하며, 하나의 샤드에는 여러명의 Collator가 무작위로 배치

  - Executor

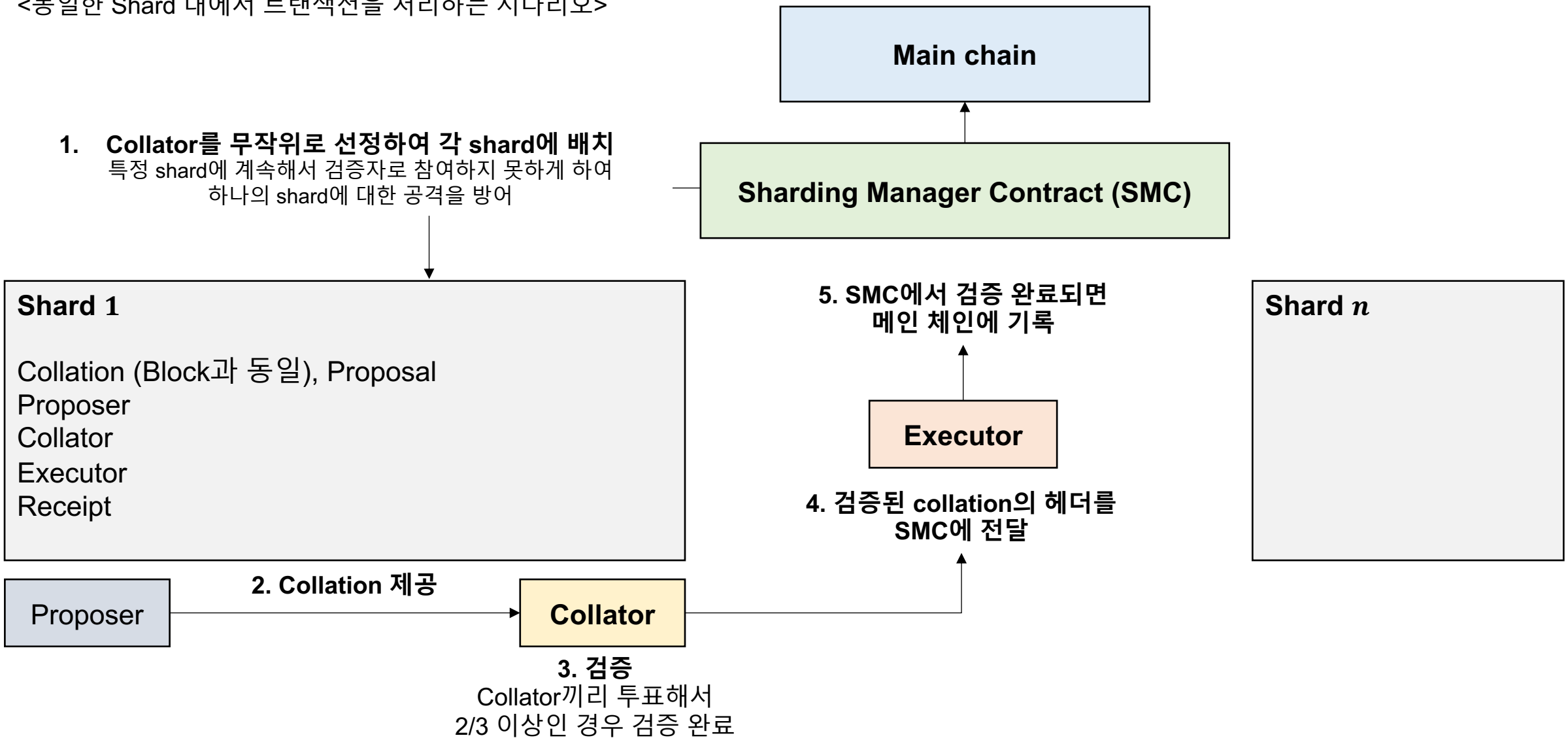
    - Collation(shard에서의 Block과 같은 개념)의 헤더를 메인체인에 있는 Sharding Manager Contract에 전달

  - Sharding Manager Contract

    - Collator로부터 받은 예치금 관리
    - 악의적 행동 방지를 위한 Collator 무작위 배치
    - Collation 검증 및 체인에 기록
    - 블록 생성을 위한 투표 관리
    - receipt 관리

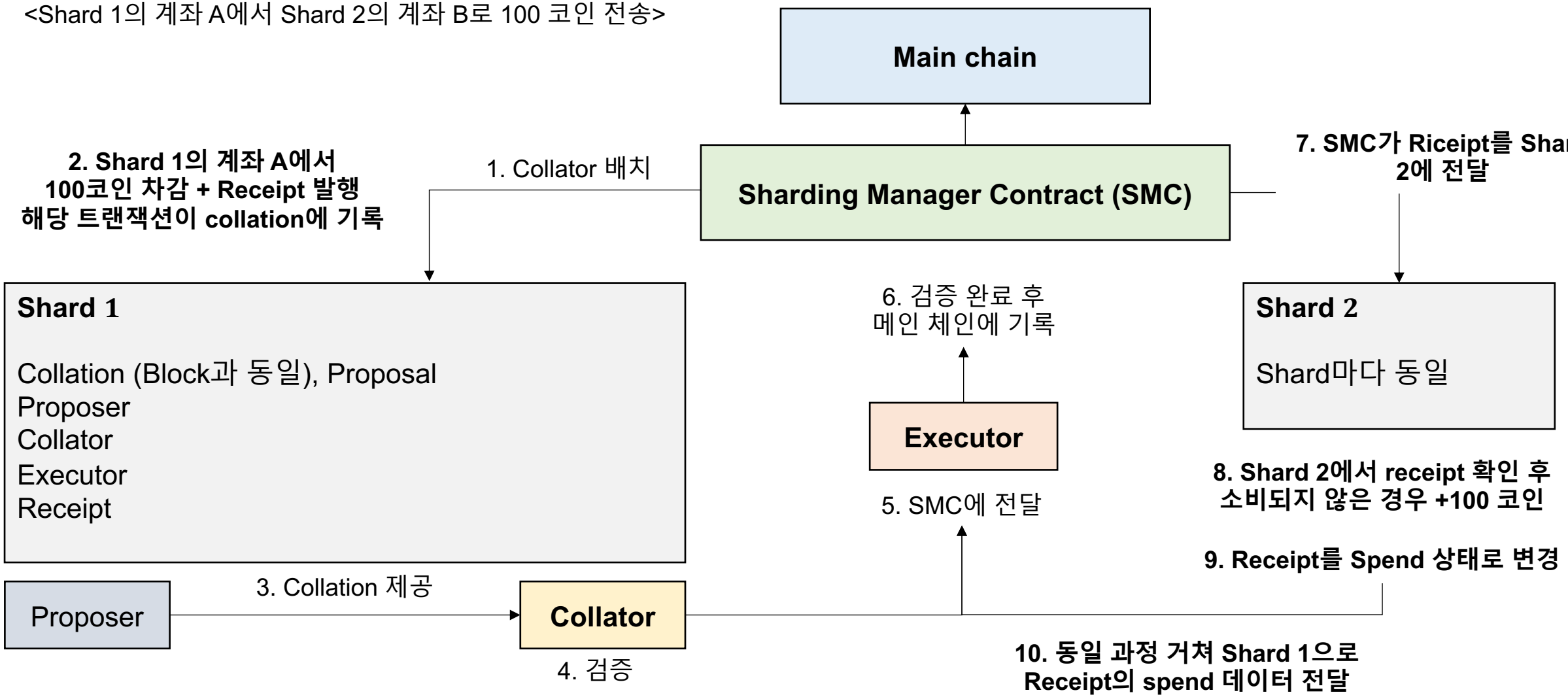
Sharding

<동일한 Shard 내에서 트랜잭션을 처리하는 시나리오>



Sharding

<Shard 1의 계좌 A에서 Shard 2의 계좌 B로 100 코인 전송>



Sharding

- **장점 : 확장성 향상**

  1. 분산화 되어 **노드별 부담이 저하** 및 네트워크 **과부하 가능성 적음**

  2. **Shard의 수에 따라서 TPS(Transaction per second) 증가**

     - 기존 블록체인은 검증 노드 수가 증가하면 보안성이 향상되지만 처리 속도가 감소 → 확장성 저하

     - Sharding이 적용된 블록체인은 노드 수가 증가하여도 collator의 수가 증가하며 여러 shard에서 병렬적 검증 가능 → **확장성 저하 방지**

- **단점**

  1. 너무 많이 분산시킬 경우

  - 정보 손실로 인한 무결성 문제

  → 다수의 shard에서 트랜잭션을 나누어 처리하므로 모든 노드에서 해당 정보를 가지고 있지 않은 경우 정보 손실 가능성 → 무결성 훼손

  - 네트워크의 악의적 공격 문제 발생 가능

  → 악의적인 사용자가 특정 데이터를 기록하지 않고 기록했다고 주장한 후, 나중에 데이터를 기록하는 악의적 행동 가능

  → 반대로 이상 없는 블록을 생성했으나, 악의적인 사용자가 데이터가 포함되지 않았다고 허위 사실 유포 가능

  2. 다른 shard의 데이터를 가지고 있지 않음

  - shard 간 데이터 참조 및 검증 방법에 관한 문제 발생 시 알고리즘이 복잡해짐

Layer1

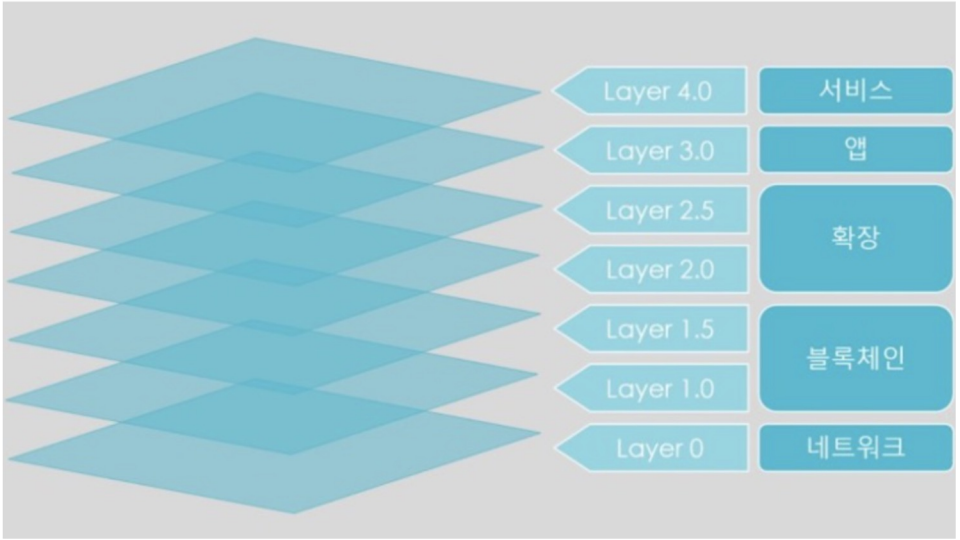
- **Layer1이란?**
  - 비트코인, 이더리움, 솔라나 등과 같이 일반적으로 알고 있는 기본 블록체인
  - 트랜잭션의 처리속도(TPS)가 증가하지 않는 확장성 문제
  - 대표적으로 비트코인에서 사용되는 PoW 합의 알고리즘은 탈중앙화와 보안을 보장하지만 많은 연산량을 필요

    → 초당 약 3~7건의 거래만 처리 가능

    → Visa의 VisaNet 전자 지불 네트워크에서 초당 24000건의 거래를 처리하는 것과 비교하였을 때 매우 적은 수치
  - Layer 2는 이러한 블록체인에서 확장성과 효율성을 향상시키기 위한 Layer

Layer 2

- **Layer 2**
  - 기존 블록체인의 확장성을 향상시키기 위한 레이어
  - 기존 레이어(Layer 1) 위에 추가적인 레이어(Layer 2)를 쌓아,
    기존 레이어의 일부 기능을 추가적인 레이어에서 수행하게 된다.



- **Layer 2 Solution (Scaling Solution)**
  - Layer 2에서의 수많은 트랜잭션을 묶어서 처리한 후
    Layer 1(블록체인)에 가끔씩만 쿼리하는 방식
    ex) 중첩 블록체인, 상태 채널, 사이드체인, 롤업, etc...

Lightning Network

- 비트코인의 확장성 문제를 해결하기 위한 오프체인 기반 Layer 2 지불 프로토콜
  - 오프체인 : 메인 블록체인 이 아닌 외부에서 트랜잭션 발생
- 거래 당사자들간의 양방향 지불 채널을 생성하여 즉각적인 트랜잭션 가능
- 직접적으로 지불채널을 생성하지 않아도 충분한 자금을 보유한 네트워크 경로가 존재한다면 트랜잭션 가능
- 지불 채널을 열고 닫는 것은 온체인 기반
  - 온체인 : 블록체인 위에 기록
- 다중 서명과 해시타임락 기술을 사용
  - 다중서명(멀티시그)
    - 하나의 주소에 n개의 개인키가 설정되어 있어, 해당 주소에서 인출을 하기 위해서는 일정 개수 이상의 개인키가 요구되는 기법
    - 다중서명주소를 통하여 수천 개의 트랜잭션을 한 번에 전송 가능하도록 한다.

      => 채널을 열고 닫을 때에만 수수료를 내면 된다.

    - 2-of-2 다중서명 주소
  - 해시타임락
    - 암호화폐의 결제 기술 중 하나로, 계약을 일정시간까지로 제한한 타임락(time lock)과 일정한 해시값이 제시되어야 계약이 성사되는 해시락(hash lock)이 결합한 형태
    - 거래의 신뢰성 제공

Lightning Network

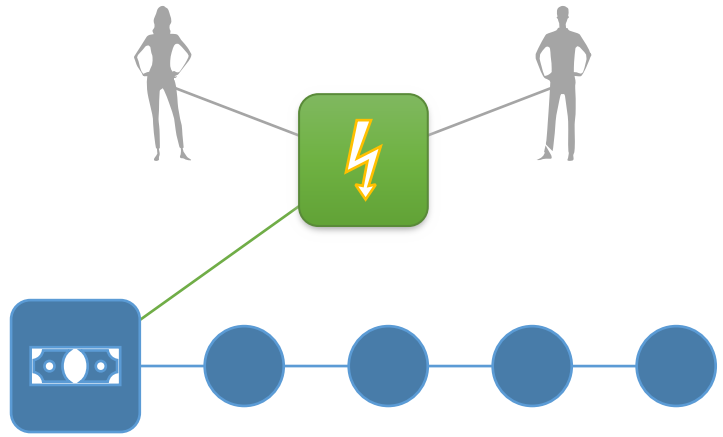
- Lightning Network의 거래 과정

  1) 당사자들간의 다중 서명 지갑 설정

  2) 지갑은 두 당사자 모두의 개인키를 통하여 접근

  3) 트랜잭션 이후, 각자가 보유하고 있는 자금을 기록하고 있는 잔고 증명서 사본에 서명 후 업데이트

  4) 트랜잭션을 모두 마친 후 지불 채널을 닫고 잔고 증명서를 비트코인 블록체인에 전송



Raiden Network

- 이더리움 블록체인에서 빠른 속도로 거래를 처리하기 위한 오프체인 방식의 네트워크 솔루션

- Lightning Network와 굉장히 유사

- **상태 채널 기반 1:1 양방향 거래**

- Raiden Network와 Lightning Network의 차이점
  - 글로벌 합의 없이 이용자들 간의 안전한 토큰 거래를 위하여 잔액 증명(balance proofs) 사용
  - ERC-20 토큰 거래를 위한 이더리움 특화 기술

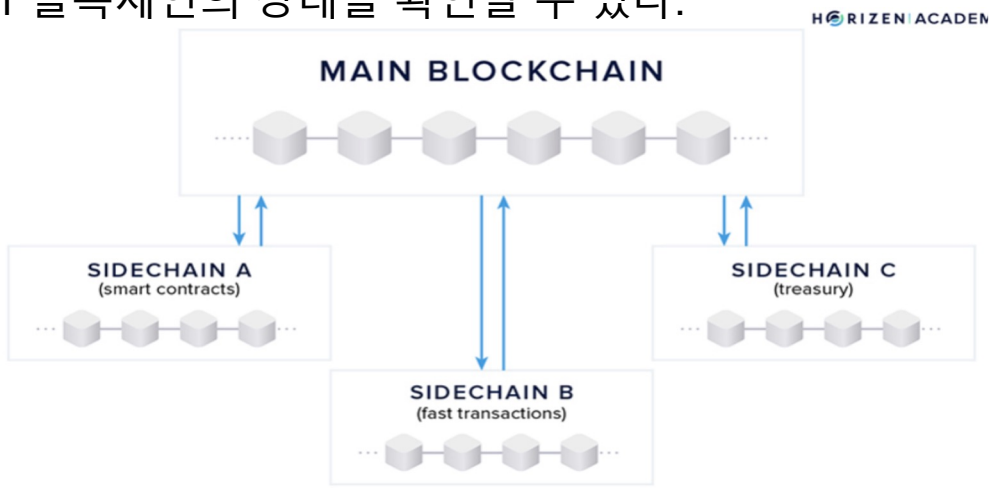
    → **Raiden Network는** 비트코인의 Lightning Network와 비슷한 **PLASMA 존재**

Scaling Solution (1)

- **Side-chain**
  - 메인넷에 대해 독립적으로 작동하는 분산 원장
  - IBC 프로토콜을 기반으로 한 양방향 연결을 통해 Layer 1 블록체인의 확장성 문제를 해결
  - 트랜잭션 확인 및 처리, 트랜잭션 작성, 합의 유지, 보안 담당
  - 자체적인 보안 및 합의 프로세스

- **Side-chain 장점**
  - Layer 1에서 사이드체인의 블록헤더만을 검증함으로써 다른 Layer 1 블록체인의 상태를 확인할 수 있다.

    → 수평적 확장성 해결



Scaling Solution (1)

- **Side-chain 단계**
  - 블록 생성, 블록 헤더 전송, 블록 헤더 검증 및 기록

1) 블록 생성

사이드체인 노드 자체 합의 알고리즘에 따라 블록을 생성한다. 그 후, 사이드체인 내에서 거래를 진행한 후 해당 거래를 블록에 기록한다.

2) 블록 헤더 전송

생성된 블록체인의 블록 헤더를 주기적으로 Layer 1 블록체인에 전송 및 검증을 요청한다.

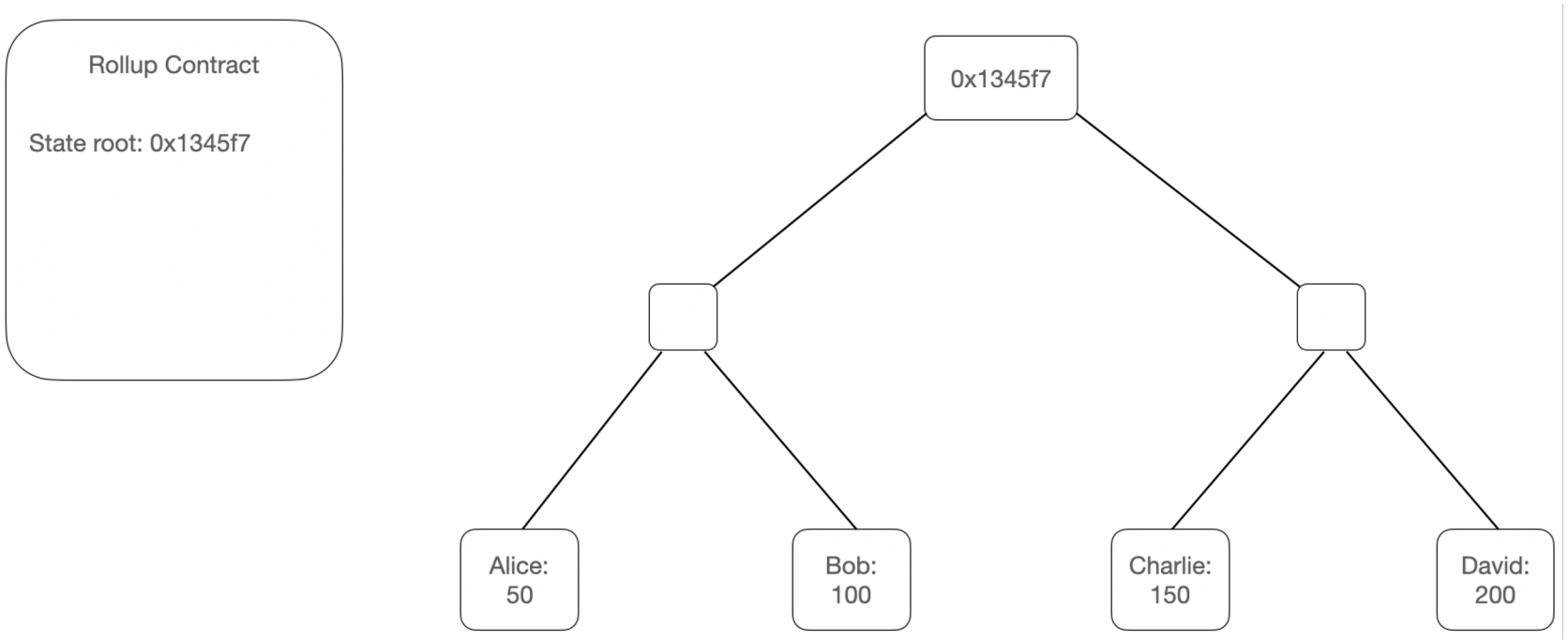
3) 블록 헤더 검증 및 기록

Layer 1 블록체인 내 노드는 사이드체인으로부터 전송받은 블록 헤더를 검증하고, Layer 1 블록체인에 이를 기록하여 트랜잭션의 검증을 완료한다.

Scaling Solution (2)

- **Rollup**
  - 메인 체인(Layer1 네트워크) 외부에서 트랜잭션을 실행하고 그 결과값만 메인 체인에 기록하는 솔루션.

  - 트랜잭션 처리량, 공개 참여, 가스 비용 측면에서 이점을 얻을 수 있다.



Scaling Solution (2)

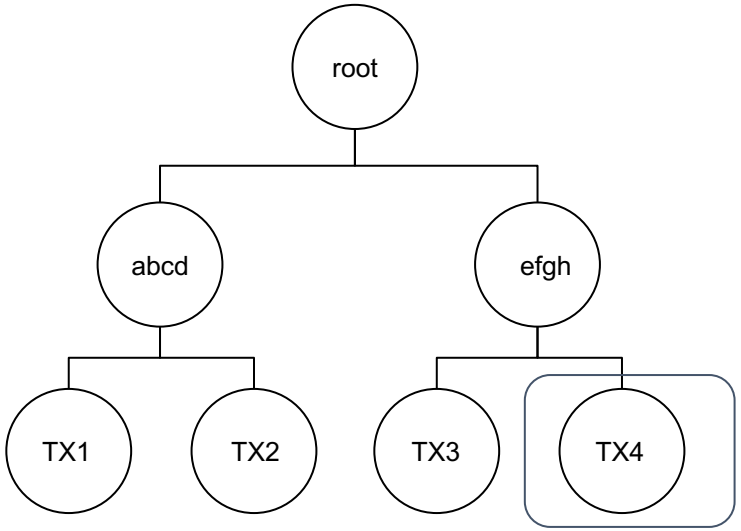
- Rollup 장점
  - 사이드체인은 반정기적으로 해시값을 전송하기 때문에 확장성은 뛰어나나 메인 네트워크와의 접점이 적기 때문에 보안은 감소한다.
  - Rollup은 사이드체인에 비해 트랜잭션 처리량은 적으나, 보안성이 우수하다.
  - 이론상, Rollup만으로도 4,807 TPS를 제공할 수 있으며, 데이터 샤딩을 통하여 최대 ~100,000 TPS까지 제공 가능하다.

- Rollup 문제점
  - 데이터 가용성 문제 (Data Availability Problem) 존재
  - 데이터 가용성 문제 해결 방식
    - **ZK-Rollup**: 오프체인(Layer2)에서 계산을 수행하고 체인에 유효성 증명을 전송한다.
    - **Optimistic Rollup**: 기본적으로 트랜잭션이 유효하다고 가정하고, 문제가 발생한 경우 위조 증명을 통해 계산만 실행한다.

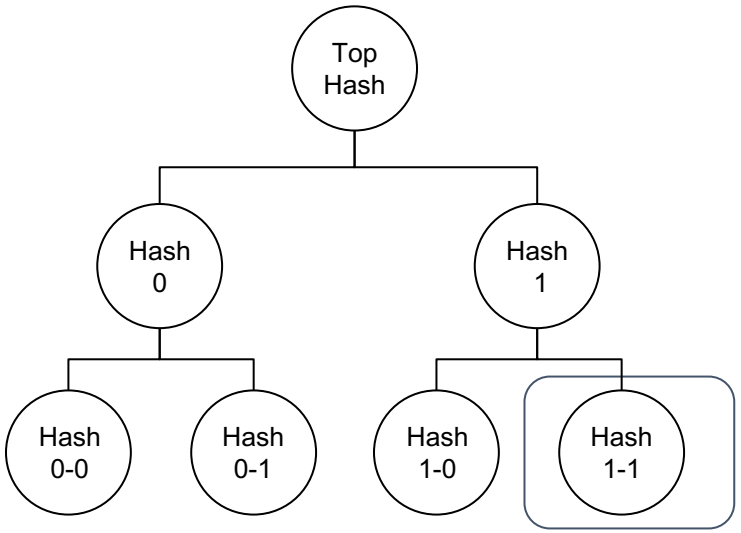
ZK-Rollup

- 이더리움 네트워크에 올라가는 트랜잭션과 ZK-SNARK 증명 함께 제공

- 트랜잭션 최적화를 통해 트랜잭션의 크기를 줄여 이더리움 네트워크에 전송
  - 주소 대신 위치 인덱스를 네트워크에 전송



1. 암호화폐를 주고 받은 주소 기록 (32-byte)



2. 주소 대신 위치 인덱스 네트워크로 전송 (4-byte)

ZK-Rollup

- ZK-Rollup 참여자 : Transactors Relayers

- Transactor(일반 사용자)
  - 전송 데이터 생성 및 생성된 전송 데이터를 보냄
    -전송 데이터에 포함된 정보? 수신인&발신인의 주소, 보내는 금액, 네트워크 수수료, Nonce(논스)
  - 발생되는 거래에 따라 변화되는 상태에 대해 새로 기록
  - 주소와 주소의 잔고에 대해서 머클 트리 형태로 기록 및 관리

- Relayers
  - Transactor로부터 받은 전송 데이터를 모아 하나의 트랜잭션을 합침
    - 이때, ZK-SNARK를 이용한 영지식 증명이 트랜잭션에 포함
  - 합쳐진 트랜잭션은 이더리움 위의 스마트 컨트랙트에서 검증
  - 영지식 증명에서 Prover 역할, 스마트 컨트랙트에서 Verifier 역할

ZK-Rollup

- 거래에 대해 사기가 아니라는 증명을 트랜잭션과 함께 전송
  - Rollup 내부 계좌가 옳다(사기가 아니다)에 대한 별도 검증기간 필요 x
  - 빠르게 결과값 확정 가능
  - 체인 내부에서 보안성이 보장되는 자금이 일정 금액으로 제한 x

Optimistic-Rollup

- 모든 트랜잭션을 이더리움에 전송하는 방법
  - 트랜잭션을 제대로 처리했는지에 대한 진위 확인
  - 모든 트랜잭션이 사실이라고 가정하고 처리
    (이때, 트랜잭션은 롤업 내부 블록에 올라감)
    - 블록을 생성하는 주체가 거래 내용을 감추거나 조작할 수 있기 때문에 사기 증명(Fraud Proof) 필수
  - 사기증명?
    - 롤업에 존재하는 검증자가 이더리움에 있는 모든 트랜잭션을 재실행하여 하나하나 값을 대조하는 과정
    - 사기를 밝혀낸 검증자에게는 보상
    - 사기라고 밝혀진 트랜잭션을 문제없다고 처리한 참여자는 처벌

- **Optimistic-Rollup에는 반드시 한 명이상의 사기 증명을 하려는(사기를 적발하려는) 참여자가 있어야 함.**

Optimistic-Rollup

- 이더리움에서 처리되는 트랜잭션을 대신 처리하고 결과만 이더리움에 전송하여 속도 향상

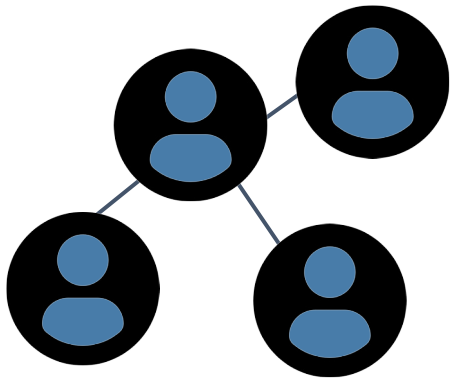
- 특정 데이터에 대한 사기 증명을 하기 위한 시간도 증명 가능한 기간이 정해져 있음
  - 트랜잭션 확정까지의 많은 시간 소요

- zk-rollup 과 다르게 내부에서 스마트 컨트랙트 구동 가능

- zk-rollup은 단순한 기능만 사용이 가능
  - zk-rollup이 스마트 컨트랙트를 지원할 수 있을 때까지 확장성 솔루션으로 각광받을 가능성이 높음

Commit Chain

- 트랜잭션들을 중계하는 하나의 운영자로부터 유지
  - 2명이상의 참가자들이 똑같은 상태를 갖고 있는 **채널과는 대조적**
- 운영자 채널에 중앙화
  - 하나의 운영자를 가진 구조에 최적화된 프로토콜
- 한번 실행하면 항상 실행중인(ongoing) 상태
  - 운영자가 Commit-chain을 시작한 후, 사용자는 운영자와의 계약을 통해 참여
- 운영자와 사용자들은 Commit-chain에서 트랜잭션 발생
- 사용자는 언제든지 자산을 가지고 부모 체인으로 출금하거나 탈출 가능
  - 신뢰할 수 없는 운영자이지만 커밋 체인을 사용하는 이유

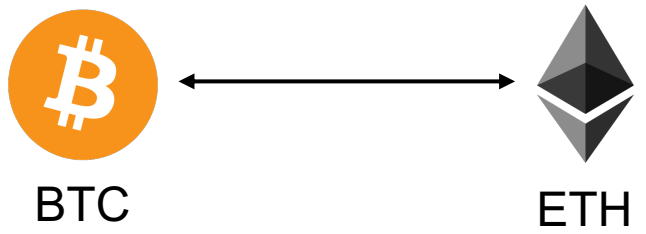


Cross Chain

- 블록체인 확장성을 높이기 위한 방법으로 서로 다른 블록체인을 연결시켜 주는 역할

  → 크로스체인 브릿지 = 환전소를 포함한 서로 다른 암호화폐를 연결하고 있는 브릿지

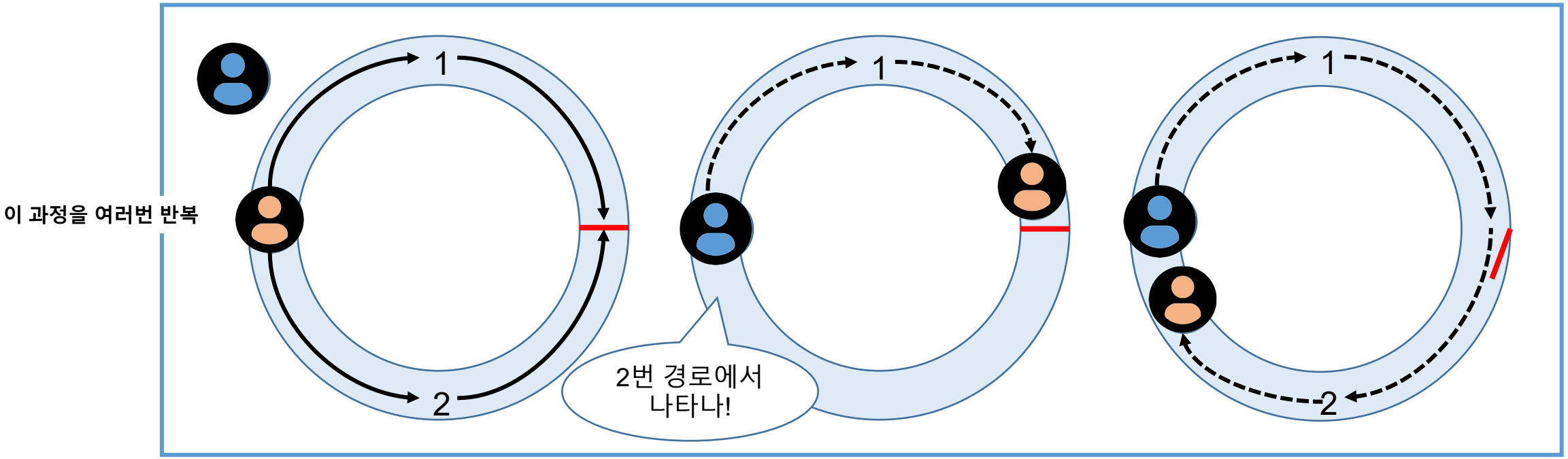
- 서로 다른 두 개의 암호화폐 간의 거래 및 교환을 가능



- 특징
  - 서로 다른 암호화폐의 교환 발생 → 복잡한 단계나 절차 없이 연결 가능
  - 암호화폐 간의 트랜잭션 발생 → 토큰 간의 교환 x, 데이터만 전송
  - 블록체인의 확장성을 높여 다른 블록체인 연결

ZK-SNARK

- Zero-knowledge Succinct Non-interactive ARgument of Knowledge의 약자
- Statement의 올바름에 대하여 어떠한 비밀 정보도 드러내지 않고 증명할 수 있는 기술
- Zcash에 의해 개발
  - Zcash? ZK-SNARK 기반으로 동작하고 있는 가장 큰 규모의 앱으로, ZK-SNARK를 적용한 최초의 암호화폐



ZK-SNARK

- 기존 ZK 간결하게(Succinct), 비상호적인 환경(non-interactive)에서 적용 가능하도록 변경
  - Succinct ? 증명의 크기가 작고 신속하게 확인 가능
  - non-interacitve ? 증명자와 검증자 사이의 상호작용이 거의 없거나 상호작용이 없음

- ZK-SNARK
  - 증명자와 검증자가 무작위 하나의 증거만 주고 받아오면 됨
  - 증명자와 검증자 사이의 신뢰할 수 있는 초기 설정에 의존
  - ZK-SNARK 활용한 트랜잭션의 경우, 수신자, 송신자 등의 정보 노출없이 트랜잭션 유효성을 다른 노드들에게 전달 가능

- 양자 컴퓨터는 ZK-SNARK의 위협으로 간주
  - ZK-SNARK는 정확한 컴퓨팅에 기반
  - 증명자의 컴퓨팅 능력이 제한적이라고 할 때, 정직하지 않은 증명자가 시스템 속일 확률 낮음
  - **But! 충분한 컴퓨팅을 갖고 있는 증명자는 가짜 증명 생성 가능**

ZK-SNARK

- 실제 블록체인 참여자가 블록의 내용을 알지못하더라도 전체 노드 중 증명 역할을 수행하는 노드가 블록의 위변조를 빠르게 검증 가능

- 타원곡선 알고리즘을 사용
  - 타원곡선 위에서는 정보는 숨기고 등식만 증명하는 것이 가능
  - 정보를 수정하는 부분인 Zero-Knowledge에 대해 구현하지 못하여 타원 곡선 알고리즘을 채택하여 사용
  - **빠른 속도로 연산을 할 수 있는 양자컴퓨터에 안전하지 않음**

ZK-STARK

- ZK-SNARK의 단점을 보완한 새로운 영지식 증명 기술
  - 더 신속하고, 저렴하게 기술 구현 가능
  - ZK-SNARK에 비해 증명의 크고, 수수료가 약간 비쌈

- Zero-knowledge Scalable Transparent Arguments of Knowledge의 약자
  - Scalable(확장 가능) : 검증에 소요되는 시간이 매우 짧음 (대규모 테스트를 진행하여도 즉각적으로 반응)
  - Transparent(투명) : ZK-SNARK와 다른게 초기 신뢰성 설정x

- 충돌 저항성 해시함수를 통해 더 희박한 대칭 암호화 의존
  - 초기 신뢰 설정 필요 x
  - 이론적으로 양자 컴퓨터에 의해 공격 받기 쉬운 ZK-SNARK의 정수론 가정 제거

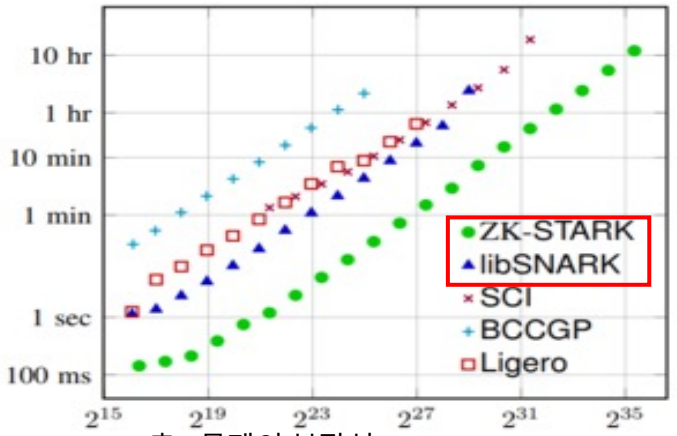
- 비교적 낮은 연산 능력을 요구하기 때문에 높은 확장성을 갖음

ZK-STARK

- 확장성
  - ZK-SNARK보다 높은 확장성

  - ZK-SNARK
    - 복잡성이 증가할수록 더 높은 연산처리 능력 필요
    - 증명 데이터와 검증 과정이 간결
    - **증명자와 증명을 생성하는데 시간이 오래 걸림**

  - 암호화 방식의 차이로 ZK-SNARK와 달리 영지식 증명의 복잡성 증명에도 연산 능력에 크게 영향이 없음
    - 증명자와 검증자 간의 communication 횟수와 계산 시간 일정



x축 : 문제의 복잡성
y축 : 증명 생성 시간

Q & A