NODES AND ASSUMPTIONS

IN BLOCKCHAIN

NODES

NODES

- 블록체인 네트워크에는 서로 다른 목적을 가진 다양한 종류의 노드들이 존재

NODES

유형		네트워크	저장할 데이터	지갑 기능			채굴 기능
				주소 확인	잔액 확인	송금	
풀노드 (Full Node)		메인 네트워크	검증에 필요한 데이터	필요 없음			필요 없음
아카이브 노드 (Archive Node)		메인 네트워크	모든 데이터	필요 없음			필요 없음
경량 클라이언트 (Lightweight Client)		메인 네트워크	검증에 필요한 최소한의 데이터	필요 없음			필요 없음
지갑 (Wallet)		메인 네트워크	필요 없음	필요			필요 없음
채굴자 (Solo Miner)		메인 네트워크 채굴 네트워크	검증에 필요한 데이터	필요	필요 없음	필요 없음	수행
채굴 풀 (Mining Pool)	서버	메인 네트워크 채굴 네트워크 클라이언트-서버	검증에 필요한 데이터	필요	필요 없음	필요	할당
	클라이언트	클라이언트-서버	연산에 필요한 데이터	필요	필요 없음	필요 없음	수행
라우터 (Router)		메인 네트워크	필요 없음	필요 없음			필요 없음

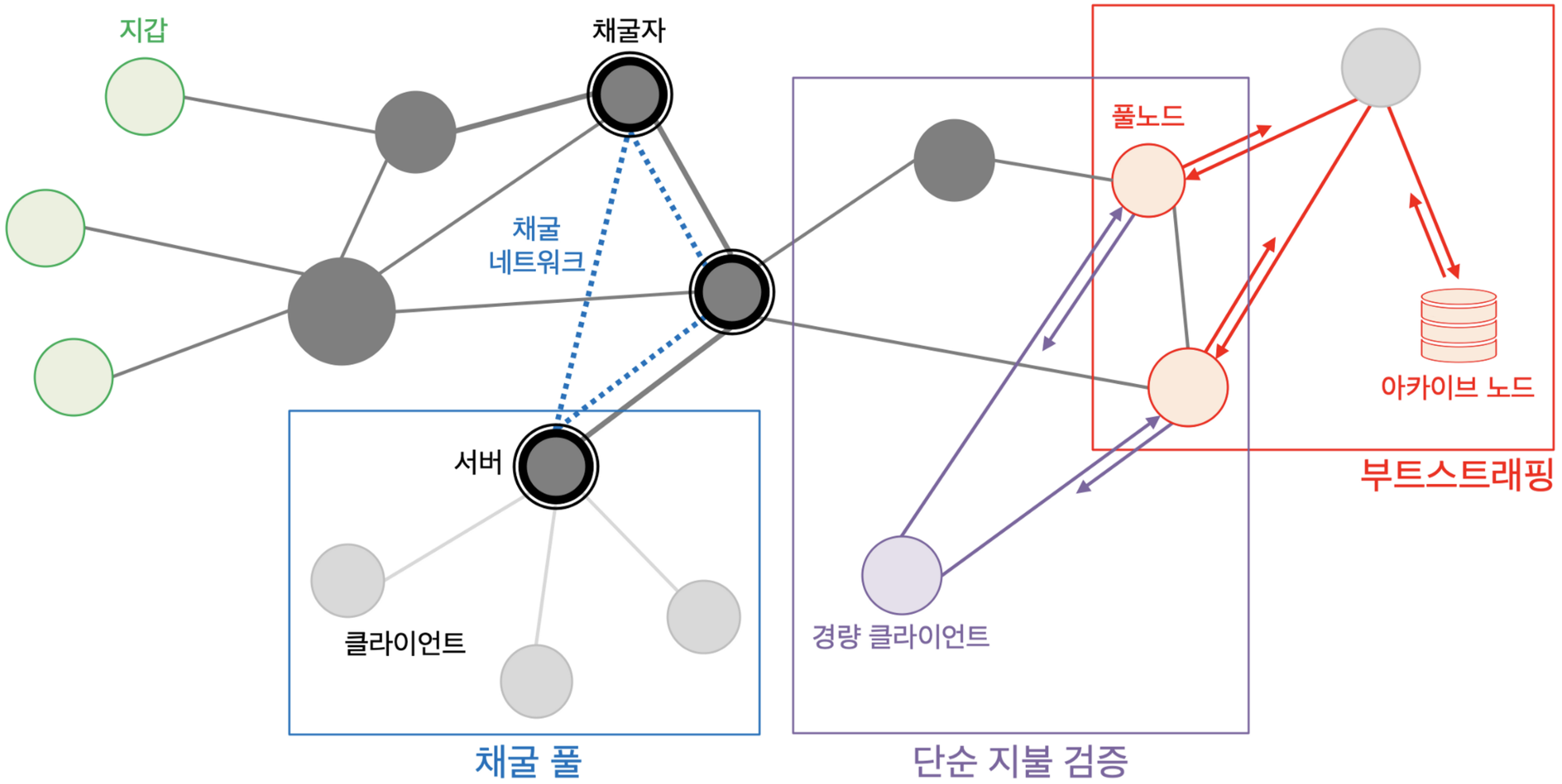
NODES

- **풀노드**: 현재 블록체인의 유효성 여부를 스스로 검증
- **아카이브 노드**: 모든 블록체인 데이터를 가진 노드
- **경량 클라이언트**: 블록 헤더의 유효성만을 검증해 체인을 형성
- **지갑**: 암호화폐를 운용하기 위한 기능을 갖춘 노드
- **채굴자**: 채굴을 수행하는 노드
- **채굴 풀**: 채굴 성공률을 높이기 위해 여러 채굴자가 모여 풀을 형성
- **라우터**: 트랜잭션 및 블록의 전파를 수행하는 노드

NODES

- 다양한 유형의 노드들이 모여 블록체인 네트워크를 형성
- 노드 간 상호작용의 관점에서 흥미로운 요소들이 존재

NODES



NODES

- **채굴 네트워크**: 채굴자 및 채굴 풀에서 보상의 기댓값을 최대화하기 위해 매우 짧은 대기 시간을 가지는 채굴 전용 오버레이 네트워크를 활용
- **부트스트랩(Bootstrap)**: 신규 노드가 블록체인 네트워크에 참여하기 위해 원장을 다운로드
- **단순 지불 검증(Simple Payment Verification, SPV)**: 경량 클라이언트가 풀노드의 보조를 받아 머클 패스 증명으로 어떠한 블록에 해당 트랜잭션이 존재함을 확인

FAULT TOLERANT

FAULT TOLERANT SYSTEM

- 이상적인 분산 시스템에서는 모든 노드와 네트워크가 정상 상황에 있음
- 현실은
  - 누구나 공격을 시도할 수 있어 서로 신뢰할 수 없음
  - 네트워크는 통신 장애나 극심한 지연 등의 상황에 부닥칠 수 있음
- 분산 시스템을 설계할 때 장애를 모델링하고 대비하는 것이 중요

FAULT TOLERANT SYSTEM

- 장애 허용 시스템(Fault tolerant system, 결함 감내 시스템)
  - 일부에서 장애가 발생하더라도 정상적으로 기능할 수 있는 시스템
  - 장애 허용이 고려되지 않으면 작은 결함만으로 전체 시스템의 구동이 중단될 수 있음
  - 선택한 장애 모델에 따라 설계 및 구현의 난이도도 달라짐

FAULT TOLERANT SYSTEM

장애 모델	메시지 도착 여부	메시지 정상 여부
실패-중단 (Fail-stop)	항상 도착함	항상 정상적
붕괴 (Crash)	도착하지 않으면 노드에 장애가 있다고 간주	항상 정상적
누락 (Omission)	T 시간 안에 도착하지 않으면 누락된 것으로 간주	항상 정상적
성능 (Performance)	도착하지 않는 상황과 늦게 도착하는 상황까지 상정	항상 정상적
입증-탐지가능한 비잔틴 (Authentication-detectable Byzantine)	도착하지 않는 상황과 늦게 도착하는 상황까지 상정	입증 및 탐지가 가능한 비잔틴 장애만을 고려
비잔틴 (Byzantine)	도착하지 않는 상황과 늦게 도착하는 상황까지 상정	임의의 모든 장애를 고려

FAULT TOLERANT SYSTEM

- **실패-중단(Fail-stop)**:
  - 장애가 발생한 노드는 실행이 중단되고 상태의 업데이트가 일어나지 않음
  - 요청에 대해서는 장애가 발생하기 전 상태를 기준으로 응답
  - 메시지가 도착하지 않는 상황을 상정하지 못함

FAULT TOLERANT SYSTEM

- **붕괴(Crash)**:
  - 요청에 응답하지 않으면 노드에 장애가 발생한 것으로 간주
  - 메시지가 도착하지 못하는 상황을 다룰 수 있지만, 그 원인을 노드의 장애로만 한정

FAULT TOLERANT SYSTEM

- **누락(Omission)**:
  - 메시지가 도착하지 않는 상황에 대해 원인이 무엇이든 단순히 누락된 것으로 간주
  - 이 모델에서 메시지는 일정 시간 안에 도착하거나, 그렇지 않으면 영원히 도착하지 않음

FAULT TOLERANT SYSTEM

- **성능(Performance)**:
  - 여러 이유로 메시지는 매우 늦게 도착할 수 있음
  - 이 경우 누락 장애 모델은 메시지가 도착하지 않는 것으로 간주
  - 반면 성능 장애 모델은 메시지가 도착하지 않는 상황은 물론이고 늦게 도착하는 상황까지 고려
  - 그러나 메시지가 비정상적인 상황을 다루지는 못함

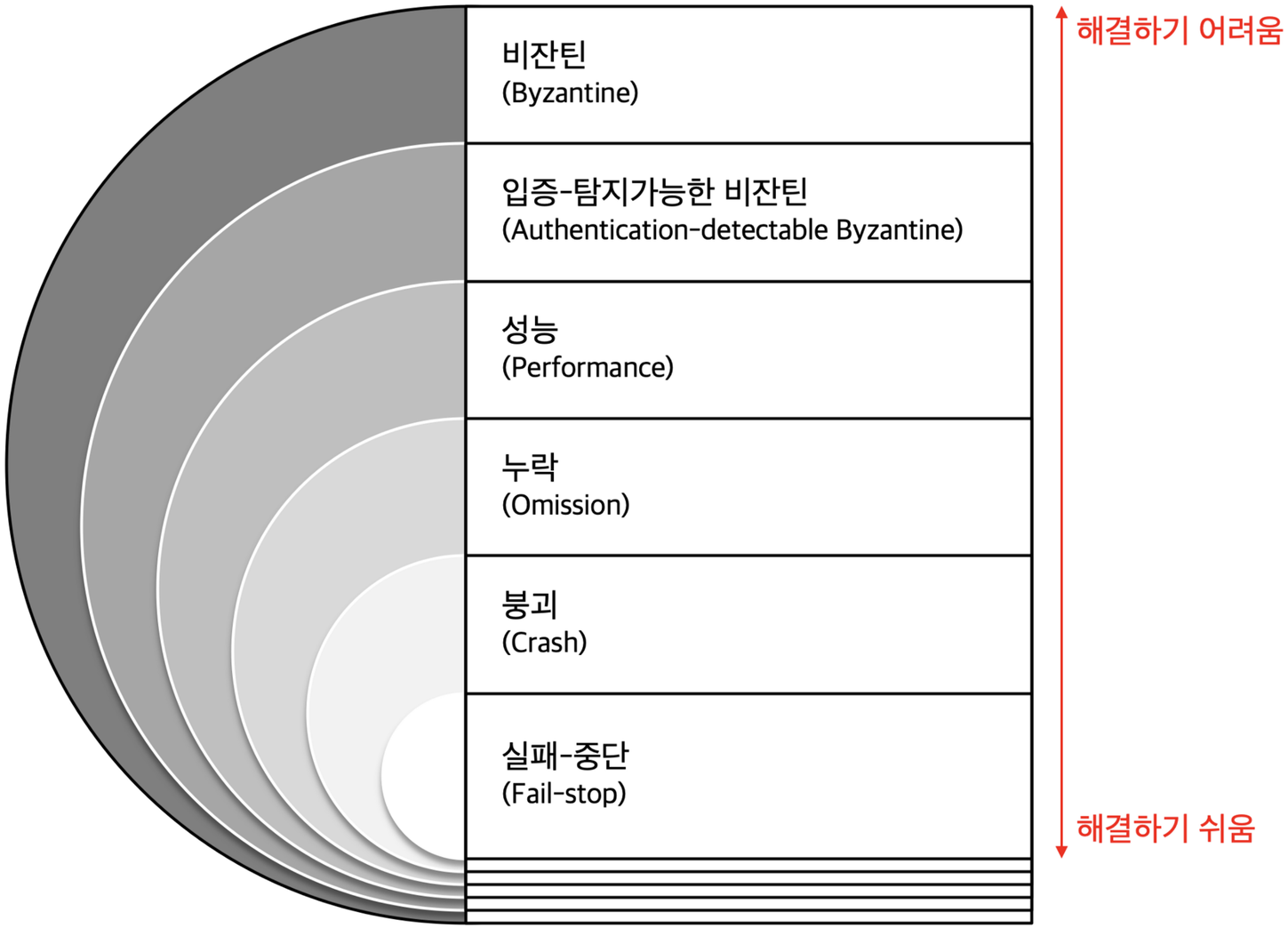
FAULT TOLERANT SYSTEM

- **입증-탐지가능한 비잔틴(Authentication-detectable Byzantine)**:
  - 송신자가 악의적이거나, 공격자로부터 위/변조된 경우,
    혹은 네트워크 장애로 인해 변조된 경우 메시지가 비정상적일 수 있음
  - 입증할 수 있고 탐지가 가능한 비잔틴 장애만을 고려

FAULT TOLERANT SYSTEM

- **비잔틴(Byzantine)**:
  - 비잔틴 장애 모델은 분산 시스템에서 가장 해결하기 어려운 장애
  - 메시지 도착 및 정상 여부의 모든 경우를 취급
  - 임의의 장애(Arbitrary Failure) 모델

FAULT TOLERANT SYSTEM



FAULT TOLERANT SYSTEM

- 블록체인은 언제나 등장할 수 있는 비정상적 메시지를 다루는 비잔틴 장애 허용 시스템
  - 이성적 참여자는 비록 그 행동이 악의적일지라도, 이익을 최대화하는 방향으로 행동
  - 합의 알고리즘과 인센티브를 잘 설계함으로써 비잔틴이 되지 않도록 유도

ASSUMPTIONS

ASSUMPTIONS

- 블록체인은 가장 난이도 높은 비잔틴 장애 모델을 효과적으로 다루는 시스템
  - 실제 구현체인 비트코인이 10년이 넘는 시간 동안 큰 문제 없이 구동됨으로써 유효성을 입증하고 있음
- 블록체인 네트워크에는 몇 가지 암묵적인 가정이 존재

ASSUMPTIONS

- **이성적 참여자**:
  - 현실의 네트워크 구성원들은 모두가 이성적이지 않음
  - 사람의 심리, 클라이언트 프로그램과 이를 수정할 능력의 한계, 프로토콜 밖에서 일어나는 거버넌스(governance) 등으로 인해 행동을 예측하기 어려움
  - 모든 참여자가 이성적이라 가정하는 것으로 단순화

ASSUMPTIONS

- **궁극적 동기화**:
  - 네트워크는 궁극적으로 동기화돼 있음
  - 일시적으로 분할 상황에 처할 수는 있으나, 영구적인 분할은 일어나지 않을 것으로 간주

ASSUMPTIONS

- **담합(collusion)**:
  - 노드들은 서로 신뢰하지 않으므로 담합이 일어날 수 없음

ASSUMPTIONS

- **뇌물(bribery)**:
  - 어떤 노드의 행동을 제어하기 위해 뇌물을 수수하는 경우는 없음

ASSUMPTIONS

- **정직한 참여자**:
  - 네트워크에서 적어도 하나의 노드는 정직해야지만 유효한 합의를 이룰 수 있음

NODES AND ASSUMPTIONS

IN BLOCKCHAIN