Consensus Mechanism

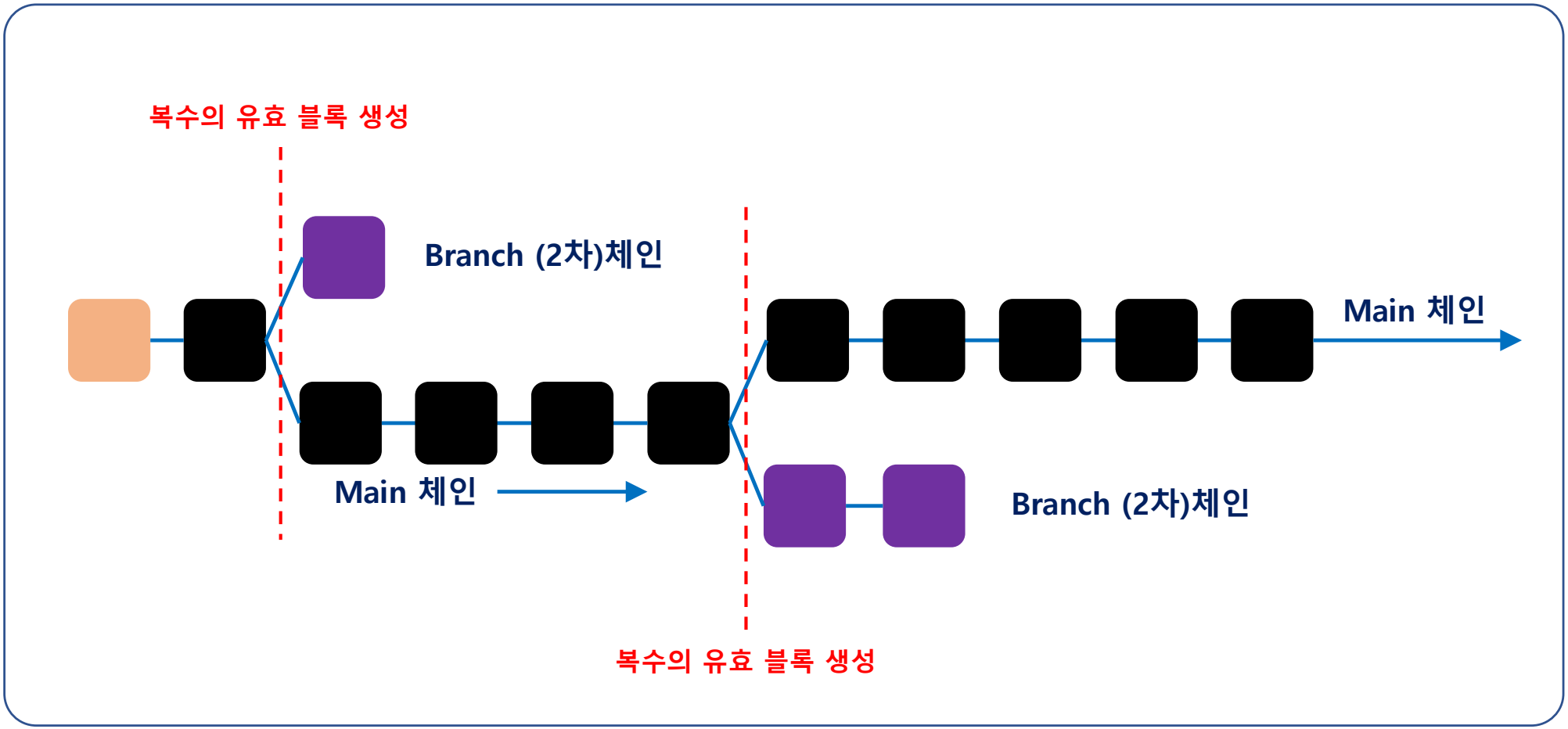
INDEX

Appendix I. 합의 알고리즘

Appendix I : 합의 알고리즘 개요

복수의 유효 블록 생성

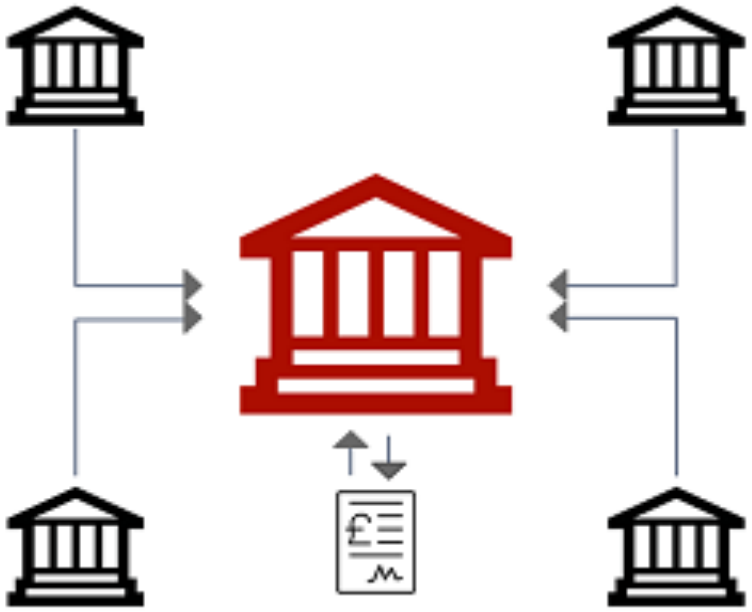
블록 생성과정에서 블록이 의도적으로 **조작된 경우** 혹은 **동일 Timestamp에 여러 블록**이 생성될 경우, 동일 번호에 **복수의 유효 블록**이 존재할 수 있습니다. 이때, 노드는 이후 **연결되는 블록의 수가 더 많은** 체인을 **Main 체인**으로 식별하게 됩니다. 따라서, 상대적으로 **먼저** 생성되어 전파된 블록이나 더 큰 **Hash Power**를 가진 노드에서 생성된 블록이 Main 체인으로 식별될 확률이 높습니다.



Appendix I : 합의 알고리즘

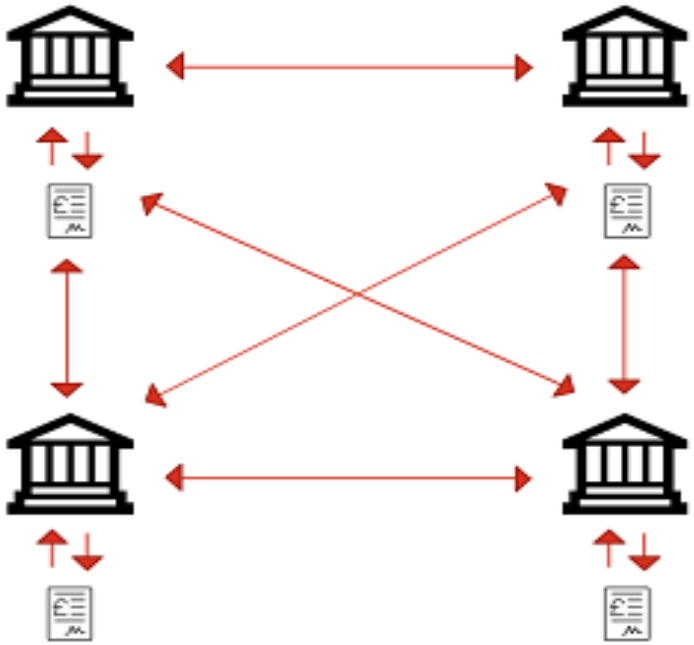
Blockchain Main Concept

중앙 집중 시스템



중앙화된 제3의 신뢰기관에서
데이터의 관리와 검증을 진행

분산 네트워크 시스템

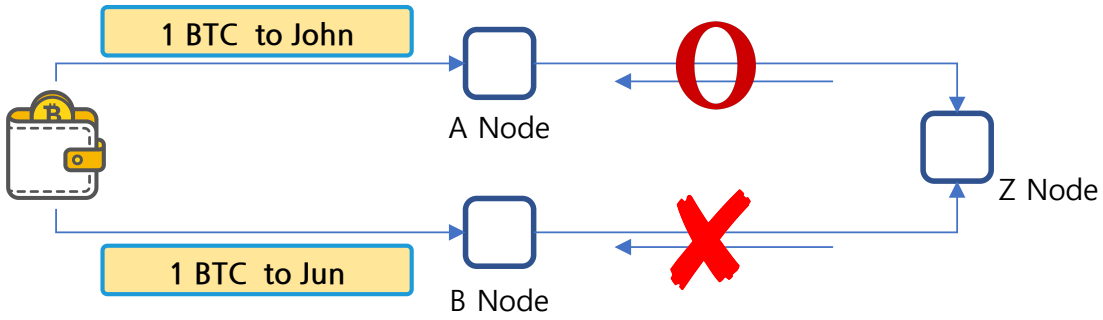


각 네트워크 참여주체가
데이터를 관리하고 원장을 공유

Appendix I : 합의 알고리즘

분산 네트워크와 신뢰 문제

Issue 1. Double-Spending



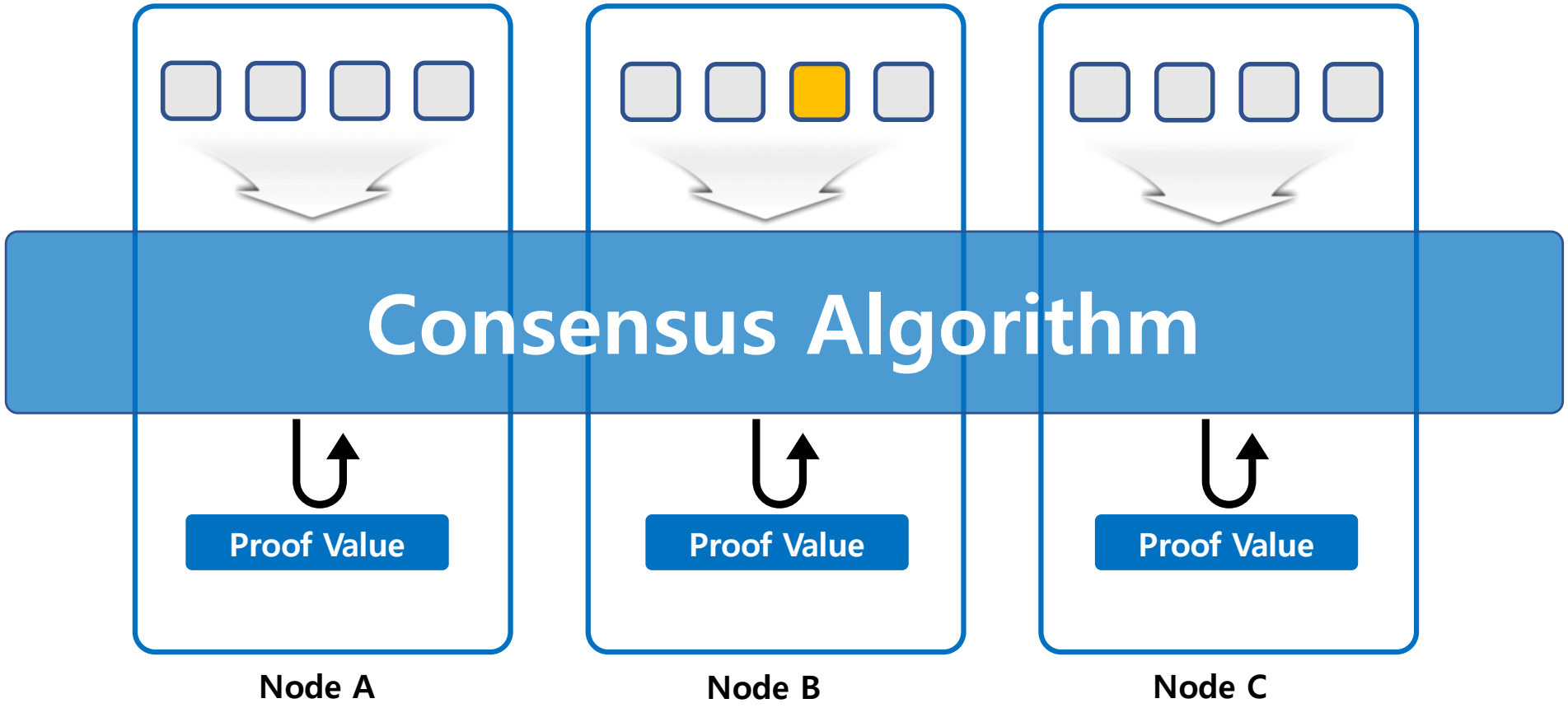
단일 입력값에 대하여 분산된 각 시스템이 동일한 결과 값을 결정해야 함

Issue 2. Byzantine General Problem

분산 시스템상에 악의적 노드가 참여하여도 전체 시스템의 신뢰도를 제공하여야 함

합의 알고리즘

합의 알고리즘은 **분산 네트워크** 상에서 서로 **신뢰관계가 없는 노드**들이 특정하게 정의된 절차를 통해 **수학적으로 계산된 결과값**을 상호 **검증**함으로써 시스템의 **무결성을 보장**하는 알고리즘입니다.



Appendix I : 합의 알고리즘

합의 알고리즘 종류

Public Blockchain의 합의 알고리즘

비트코인의 경우 **작업증명** 알고리즘을 사용중이며, **이더리움**의 경우 2017년 8월부터 작업증명과 지분증명 알고리즘을 **Hybrid 형태**로 사용하고 있으며, 2018년중에 **지분증명으로의 전환**을 목표로 하고 있습니다.

✓ **작업증명 (Proof of Work, PoW)**

- 블록체인에서 가장 보편적으로 사용중인 합의 알고리즘으로 컴퓨팅 파워를 이용하여 특정 난이도의 해시값을 역함수 해시화 하여 Nonce값을 계산해내고 이를 검증하는 것으로 합의를 도출함

✓ **지분증명 (Proof of Stake, PoS)**

- PoW의 컴퓨팅 파워 낭비 문제를 해결하고자 개발된 합의 알고리즘으로 노드가 보유한 자산을 기준으로 권한을 분배하여 합의를 도출하고 보상을 분배하는 알고리즘

✓ Proof of Elapsed Time (PoET) 등의 다양한 알고리즘 존재

Private Blockchain의 합의 알고리즘

Private Blockchain의 경우 보편적으로 **PBFT와 PAXOS** 알고리즘을 사용하고 있으며, Enterprise Ethereum의 대표 프로젝트인 **Quorum**의 경우 **Raft** 알고리즘을 채택하여 사용하고 있습니다.

✓ **PAXOS**

- 가장 일반적인 합의 알고리즘으로 Leader를 선정하고 과반수의 동의에 의해 합의를 이룸

✓ **Practical Byzantine Fault Tolerance (PBFT)**

- 비잔틴 장군 문제를 해결하고자 고안된 합의 알고리즘으로 투표 메카니즘을 도입한 3단계 프로토콜을 이용한 합의 도출로 프라이빗 블록체인에서 널리 이용됨

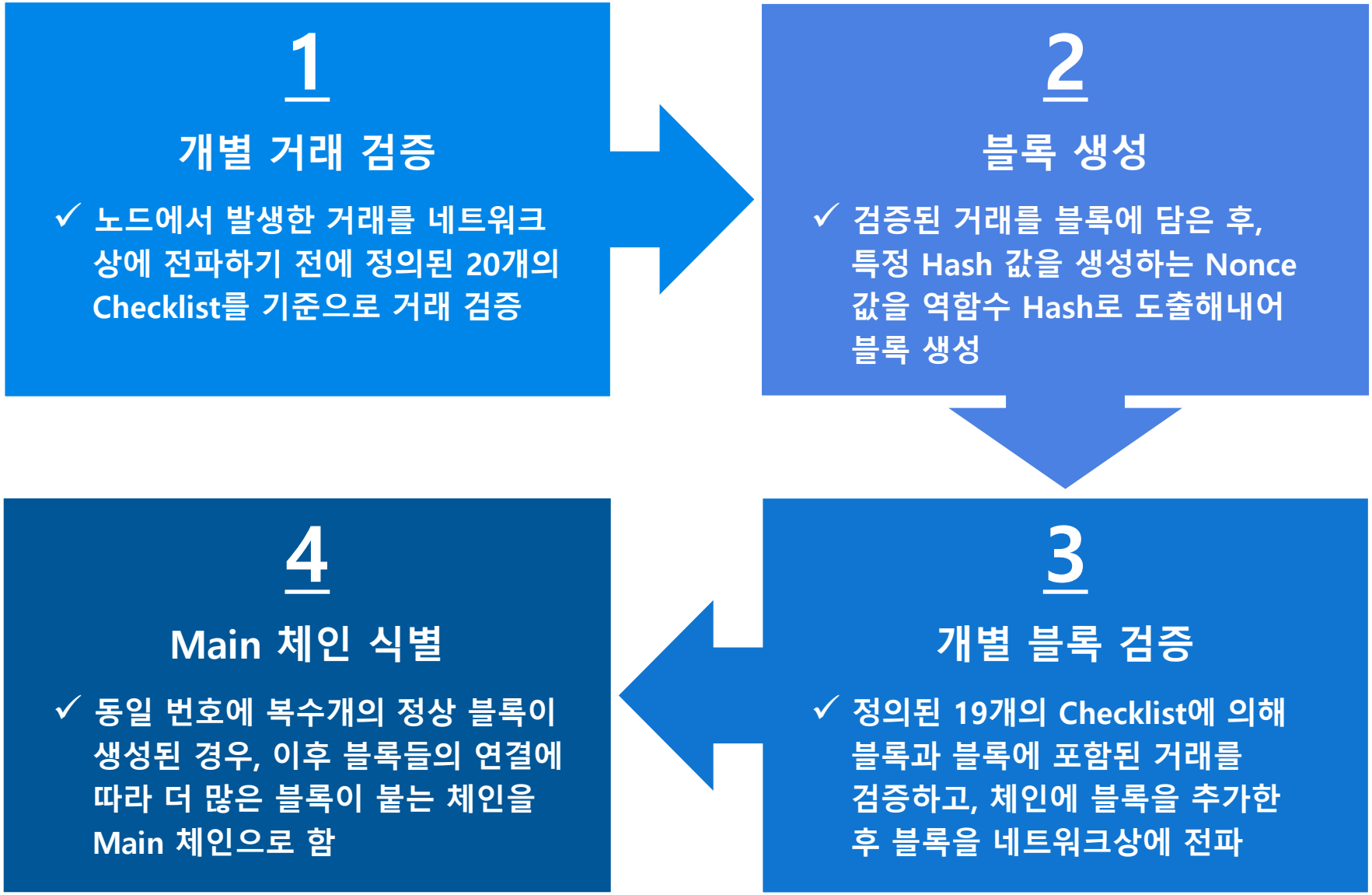
✓ **Raft**

- PAXOS를 보완한 형태로, 투표와 랜덤 타임아웃을 통한 리더 선출로 절차를 단순화 하는 것이 특징

✓ SBFT, Tendermint 등의 다양한 알고리즘 존재

Appendix I : 합의 알고리즘 개요

분산화된 합의 – 작업증명 알고리즘 (PoW)



Appendix I : 합의 알고리즘 개요

coinplug

Hash Function의 이해 / 합의 알고리즘(Pow : Proof of Work)

▶▶ 해시 함수(Hash Function)

- ✓ 입력 x Value에 대해서 고정된 길이의 출력 y Value로 변환 시켜주는 함수
- ✓ SHA : Secure Hash Algorithm, 공개 표준 해시 함수로 결과값은 160비트로 고정
- ✓ 비트코인 : SHA-256 함수 적용, 256비트 크기

x Value 무결성 검증

f(x) = y

- 역함수 계산 :: x value 반복 대입 :: Hash Rate
- y가 160비트이면, 최대 2^{160}번 시도

CPDAX

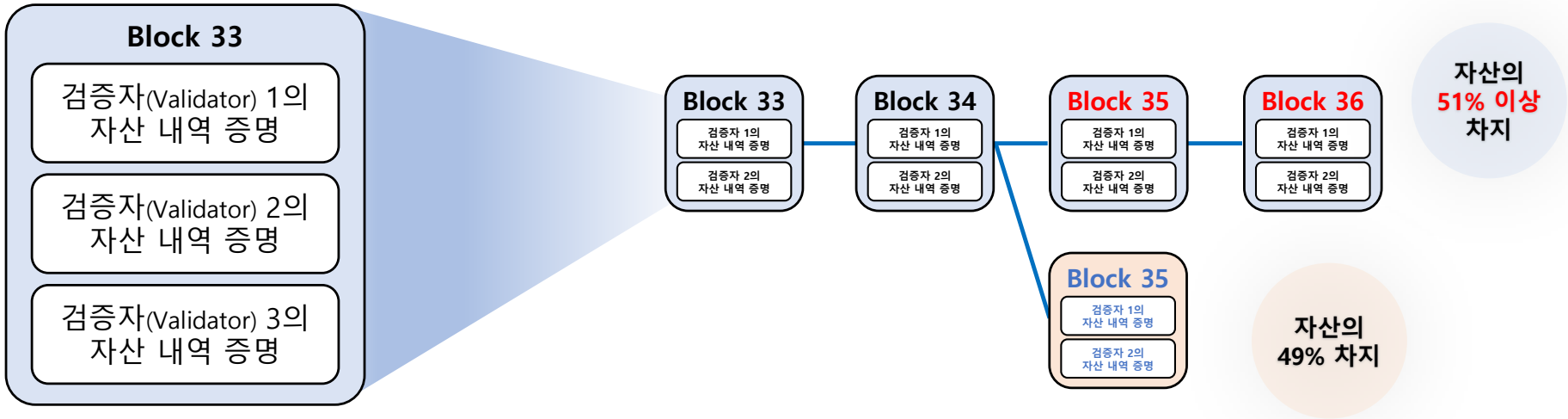
Appendix I : 합의 알고리즘 개요

분산화된 합의 – 지분증명 알고리즘 (Proof of Stake)

작업증명 알고리즘(PoW)은 **Hashing Power(연산능력)에 비례하여** 블록에 데이터 업데이트 권한을 획득했다면,
지분증명 알고리즘(PoS)은 더 많은 **지분(해당 코인)을 가지고 있을수록 그에 비례하여** 블록에 데이터 업데이트 권한이 부여됩니다.

합의(동의)하는 블록에 검증자는 각자의 자산 보유 내역(자산 리스트)을 포함하여 증명

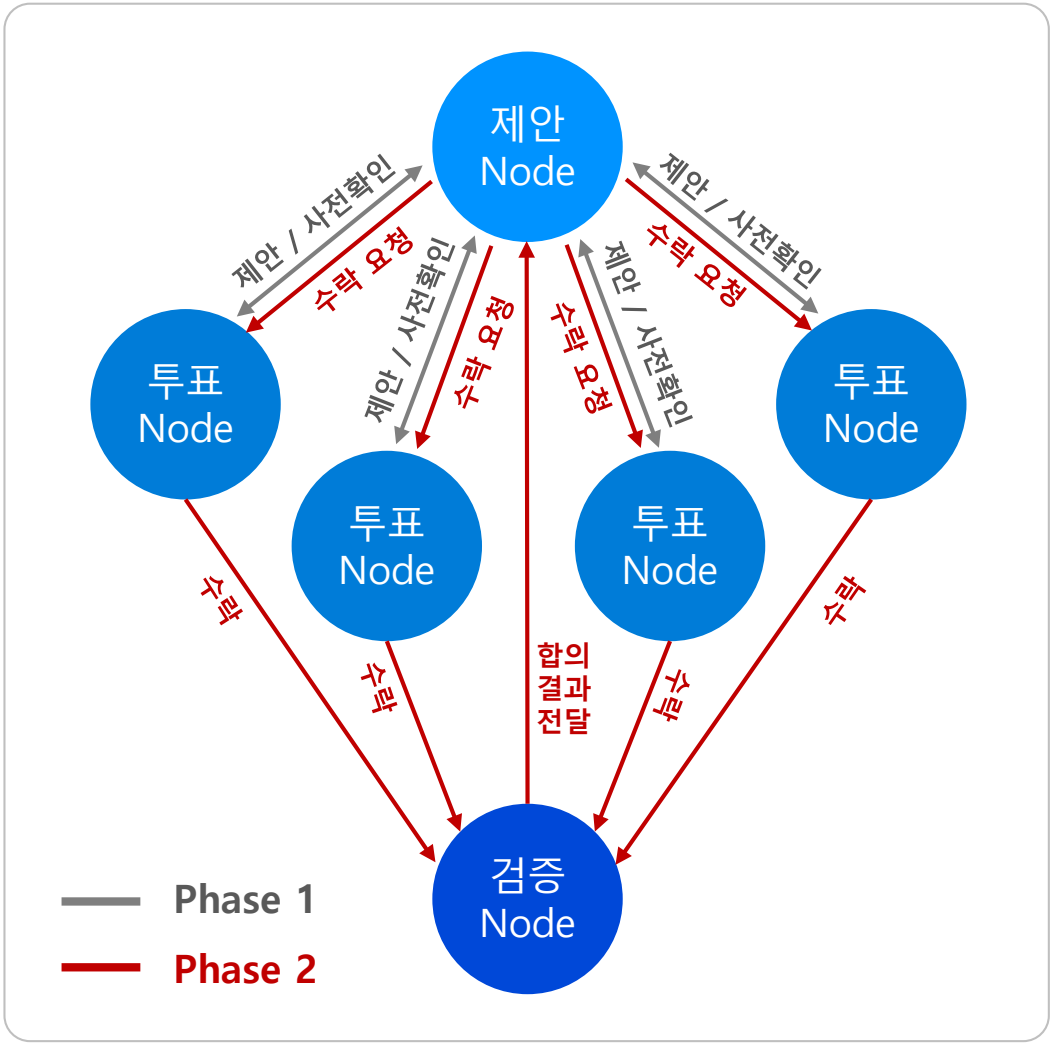
Forking시 더 많은 자산을 보유한 체인으로 합류



✓ PoW(Proof of Wark) 와 같이 PoS(Proof of Stake)도 **블록이 생성될때 보상이 지급**되는데 **지분에 대한 이자의 개념**으로 보상 지급

Appendix I : 합의 알고리즘 개요

PAXOS



PAXOS 알고리즘 절차

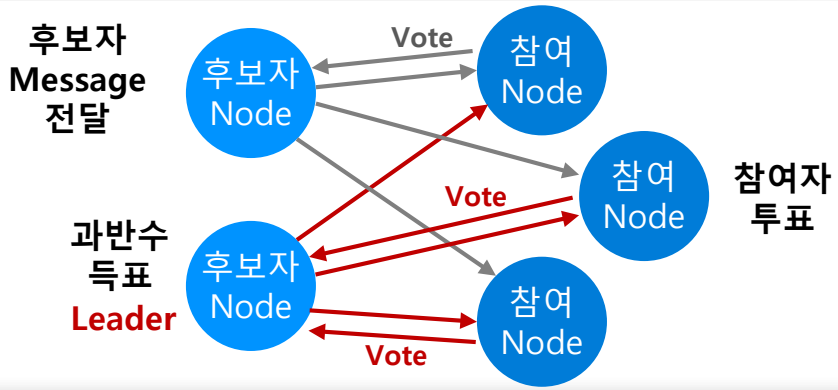
- ✓ **사전확인과 수락 처리 2단계 절차 진행**
- ✓ **Phase 1. 사전 확인**
  - 제안 Node가 합의 도출을 위한 제안 메시지를 각 투표 Node로 전달
  - 과반수의 사전 확인 후 수락 요청 처리
- ✓ **Phase 2. 수락 요청 및 최종 합의**
  - 제안 Node 수락 결과가 검증 Node에 취합되며 최종 합의 결과는 제안 Node에 전달하여 완료 함

Appendix I : 합의 알고리즘 개요

Raft

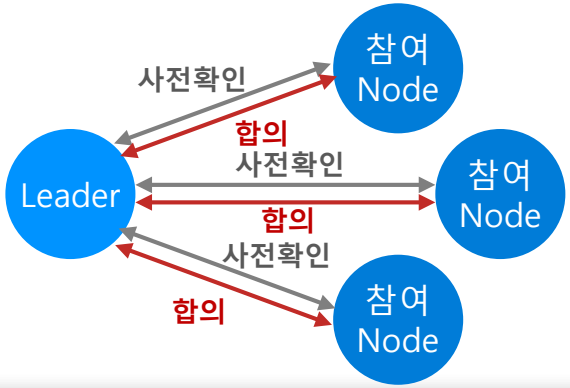
Raft는 **PAXOS** 알고리즘을 보완한 형태의 합의 프로토콜로써, **Leader를 선출**하고 해당 Leader로 하여금 거래 처리 및 합의 도출 과정을 전담하게 하여, 프로세스를 **단순화** 하고 **합의 속도**를 **향상**시켰습니다.

Phase 1. Election (Leader 선출)



- ✓ **한번에 하나의 Leader**만이 존재하며, 선출 주기는 별도로 지정되어있지 않고 **다음 Leader 선출 시 까지**로 함
- ✓ 참여자가 **heartbeat timeout** 시간내에 메시지를 받지 못하면 Leader가 없다고 판단하여 **후보자**가 됨
- ✓ Election 진행 시, **과반수**의 득표를 얻은 후보자가 없는 경우 **새로운 Election**을 다시 진행

Phase 2. Log Replication (합의)



- ✓ **PAXOS**와 유사한 구조로 합의 진행 / Leader만이 메시지를 전달하여 합의 절차를 진행할 수 있음
- ✓ **1차**적으로 메시지 수락에 대한 **사전 확인**을 **과반수**의 노드로 부터 회신 받고, **2차**적으로 **모든 노드**로 부터 **확정**을 회신 받음으로 합의 도출
- ✓ **Leader 부재** 시, 모든 과정은 중단되고 **Phase 1 진행**

Appendix I : 합의 알고리즘 개요

coinplug

PoW - 거래 / 블록 검증 Checklist

거래 검증 Checklist

- ✓ 거래의 구문과 데이터 구조가 정확해야 한다.
- ✓ 입력값이나 출력값 목록이 비어 있지 않다.
- ✓ 거래 크기가 Block Size보다 작다.
- ✓ 모든 출력값과 그 총합은 합법적인 금액 범위 내에 있어야한다.
- ✓ 출력값의 총합이 입력값의 총합보다 작아야한다.
- ✓ 풀이나 메인 브랜치에 있는 블록에 짝을 이루는 거래가 존재해야 한다.
- ✓ 각 입력값에 대한 해제 스크립트는 그에 해당하는 출력값 잠금 스크립트에 대해 검증해야 한다.
- ✓ 각각의 입력값에 대해, 출력값이 풀 내의 어떠한 거래 내부에 존재한다면 해당 거래는 거부되어야 한다.
- ✓ 각각의 입력값에 대해 출력값은 존재해야 하며, 이미 소비되었으면 안된다.

⋮ **20개**

블록 검증 Checklist

- ✓ 블록의 데이터 구조는 문법적으로 유효하다.
- ✓ 블록의 타임스탬프는 향후 2시간 이내이다.
- ✓ 블록 헤더 해시는 목표 난이도보다 작다.
- ✓ 블록의 크기는 허용할 수 있는 한도 내에 있다.
- ✓ 등록되는 제일 첫 거래는 코인베이스 생성거래 (채굴 보상을 위한 신규 Bitcoin생성 거래)이다.
- ✓ 블록 내에 있는 거래 전부는 '거래의 독립적 검증'에서의 Checklist를 이용하여 유효성 검증을 거쳐야한다.
- ✓ 블록에 포함된 거래 리스트가 비어있어서는 안된다.
- ✓ Nonce를 포함하고 Hash 화를 진행하여 작업증명을 검증하여야 한다.
- ✓ Merkel Hash를 검증하여야 한다.

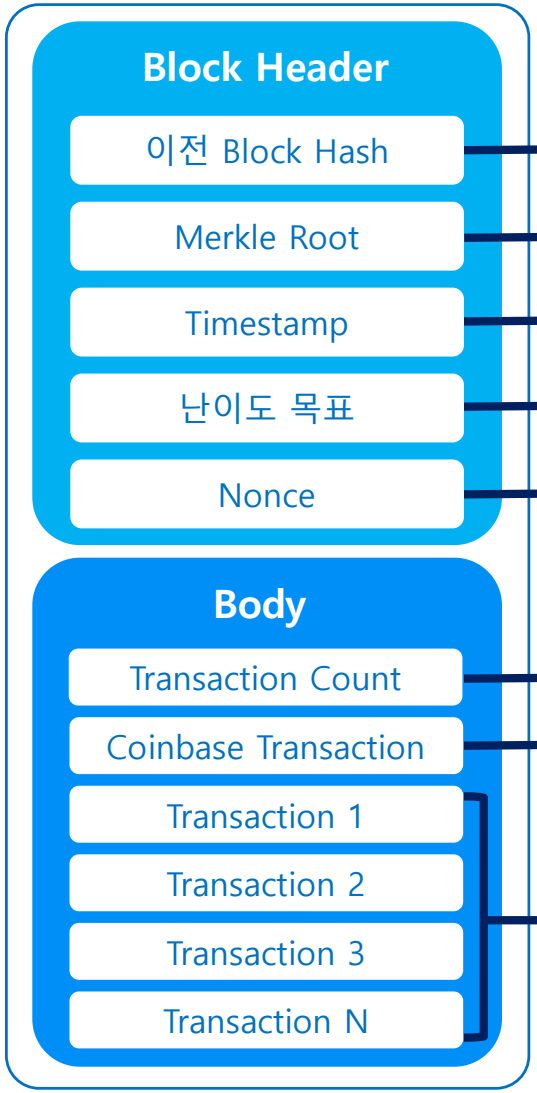
⋮ **19개**

Appendix I : 합의 알고리즘 개요

coinplug

Block의 구조

Block 구조



Data 상세 설명

- ✓ 이전블록에 대한 해시 참조값
- ✓ 블록에 포함된 거래들의 머클루트 해시값
- ✓ 블록 생성시간 (초단위)
- ✓ Bit값으로 블록의 작업증명 난이도 목표
- ✓ 작업증명 알고리즘에 사용되는 카운터
- ✓ 포함된 거래의 갯수
- ✓ 블록 생성 시 발생되는 비트코인에 대한 거래
- ✓ 10분동안 수집된 거래들의 정보

CPDAX

감사합니다.