하이퍼레저 패브릭 기반의 프라이빗 블록체인 개발 과정

2022-02-14

빅픽처랩㈜

안휘

강사 소개

- **블록체인기반 소프트웨어 시스템 설계 및 개발 전문가**
- **경력**
  - 빅픽처랩(주), CTO, 2018.5 ~ 현재
  - KAIST, 전산학부, 박사과정, 2012.3 ~ 현재
  - CMU, MSIT-SE, 석사, 2012
  - KAIST, 전산학과, 학사, 2010
- **대표 프로젝트**
  - **잇닷 & trust-chain**: 블록체인 기반 오피니언 보드 소프트웨어 서비스
    - Hyperledger Fabric, MongoDB, GoFiber(Go), Fastify(node.js), React, React Native
  - **it-chain**: 오픈소스 경량 블록체인 엔진
- **Background**
  - Software Architecture & Software Engineering

안 휘
E. hwi.ahn@bigpicturelabs.io
M. 010-2695-0232

전체 목차

- **하이퍼레저 패브릭 소개 (1 day)**
  - 블록체인이란?
  - 블록체인 기술들과 하이퍼레저 패브릭
  - 하이퍼레저 패브릭 구조 개요
  - 하이퍼레저 패브릭 구동 실습
- **하이퍼레저 패브릭 개발 (2 day)**
  - 하이퍼레저 패브릭 아키텍처
  - 스마트 컨트랙트 개요
  - 하이퍼레저 패브릭 스마트 컨트랙트 개발 실습

블록체인이란?

2022-02-14

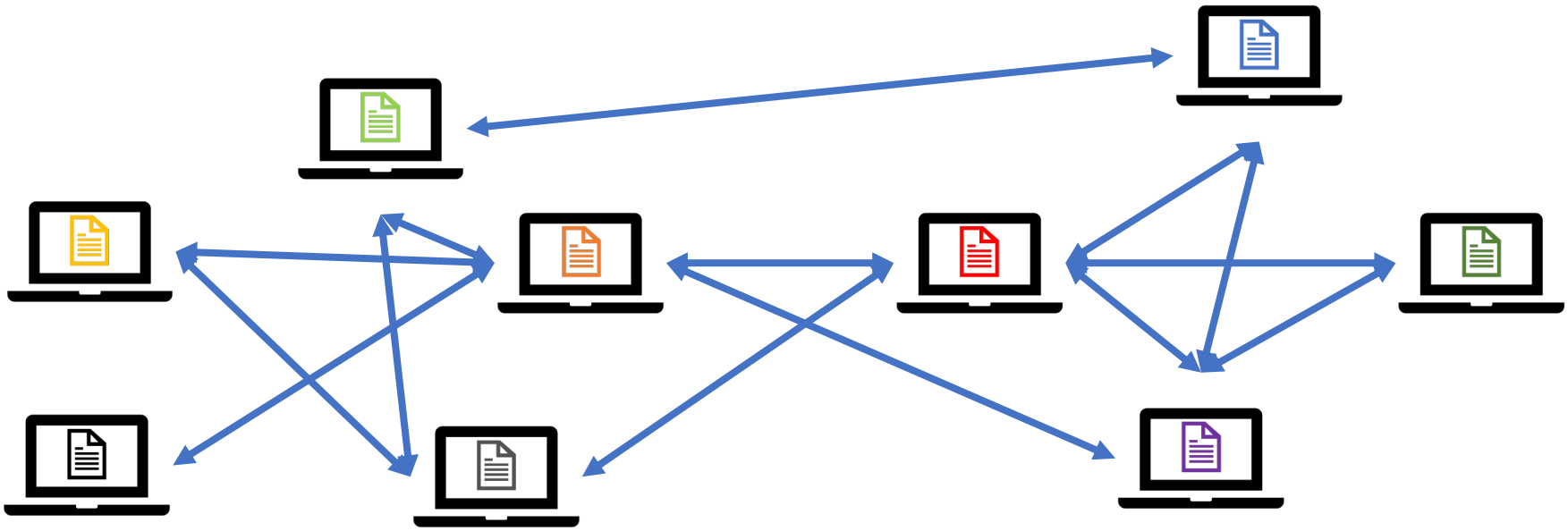
빅픽처랩(주)

안휘

목차

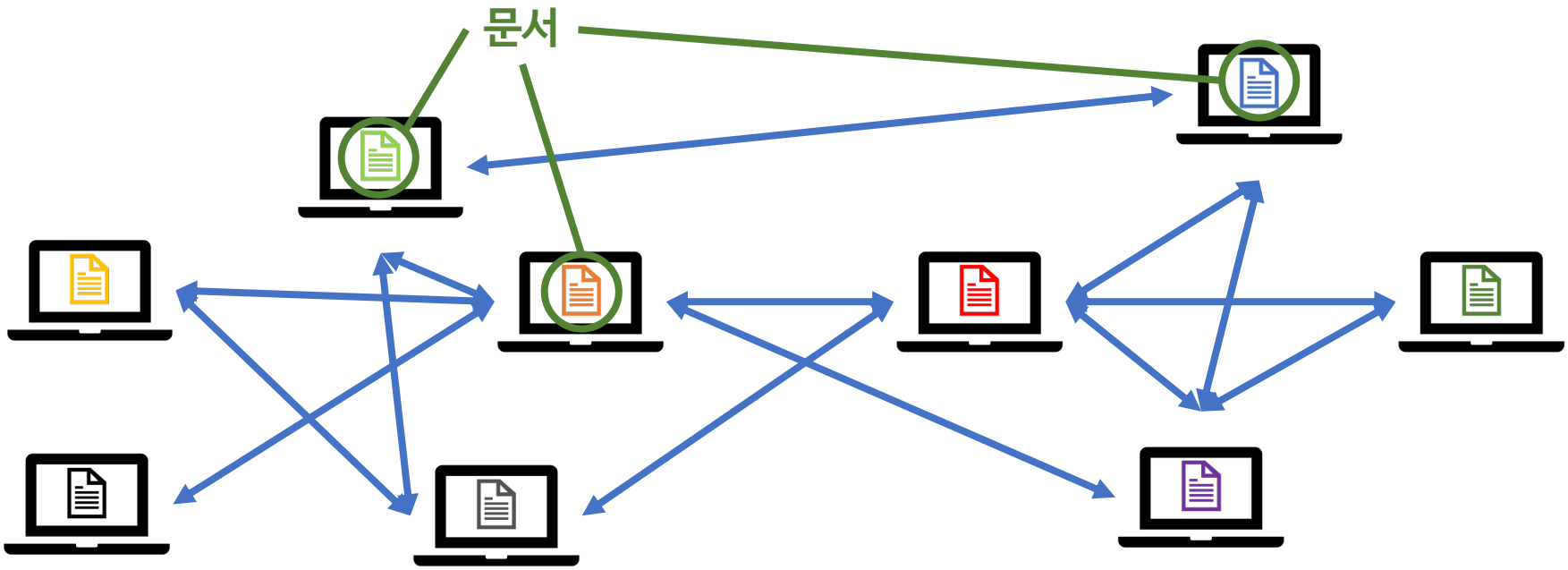
1. 인터넷

- **네트워크**
  - 컴퓨터들이 연결되어 있는 것
  - 왜? 서로의 **정보를 공유**하기 위해
- **인터넷: 네트워크의 네트워크**
  - 컴퓨터들 사이의 모든 네트워크를 연결한 거대한 네트워크



1. 인터넷

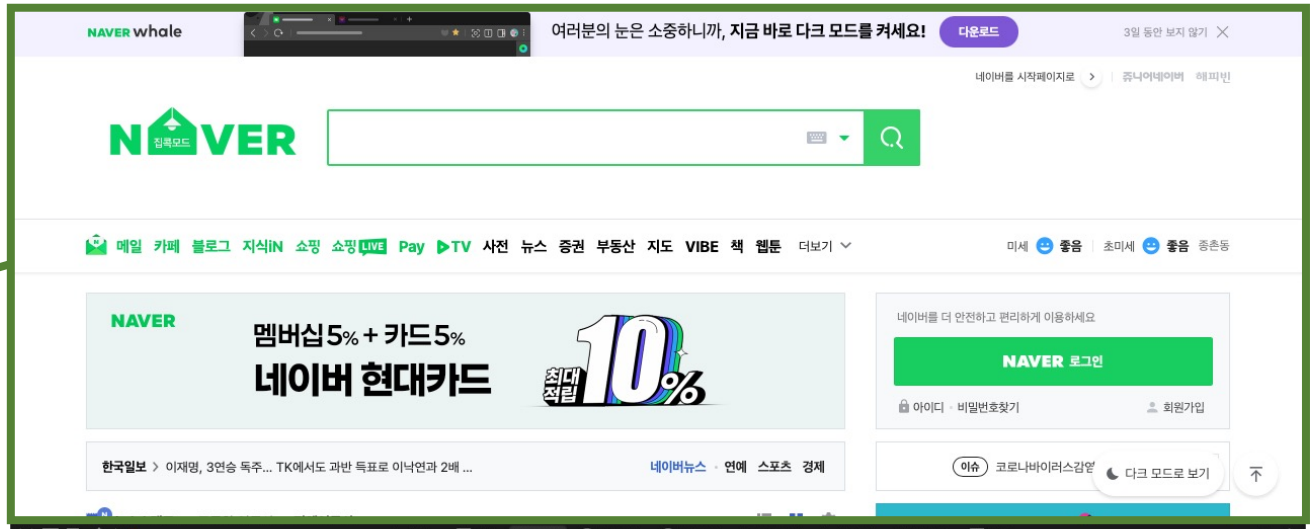
- **월드 와이드 웹(World Wide Web, WWW)**
  - 줄여서 “웹”
  - 가장 대중적인 인터넷 기반 정보 공유 공간
  - 정보는 HTML로 작성된 문서로 공유됨



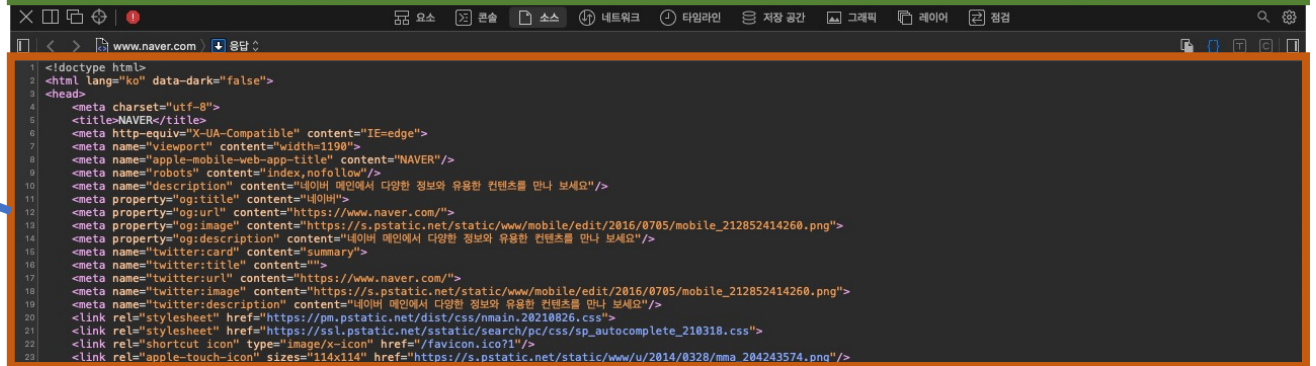
1. 인터넷

- 월드 와이드 웹(World Wide Web, WWW)
  - 줄여서 “웹”
  - 가장 대중적인 인터넷 기반 정보 공유 공간
  - 정보는 HTML로 작성된 문서로 공유되고 “웹 브라우저”를 통해 봄

웹 브라우저로 본 문서의 모습

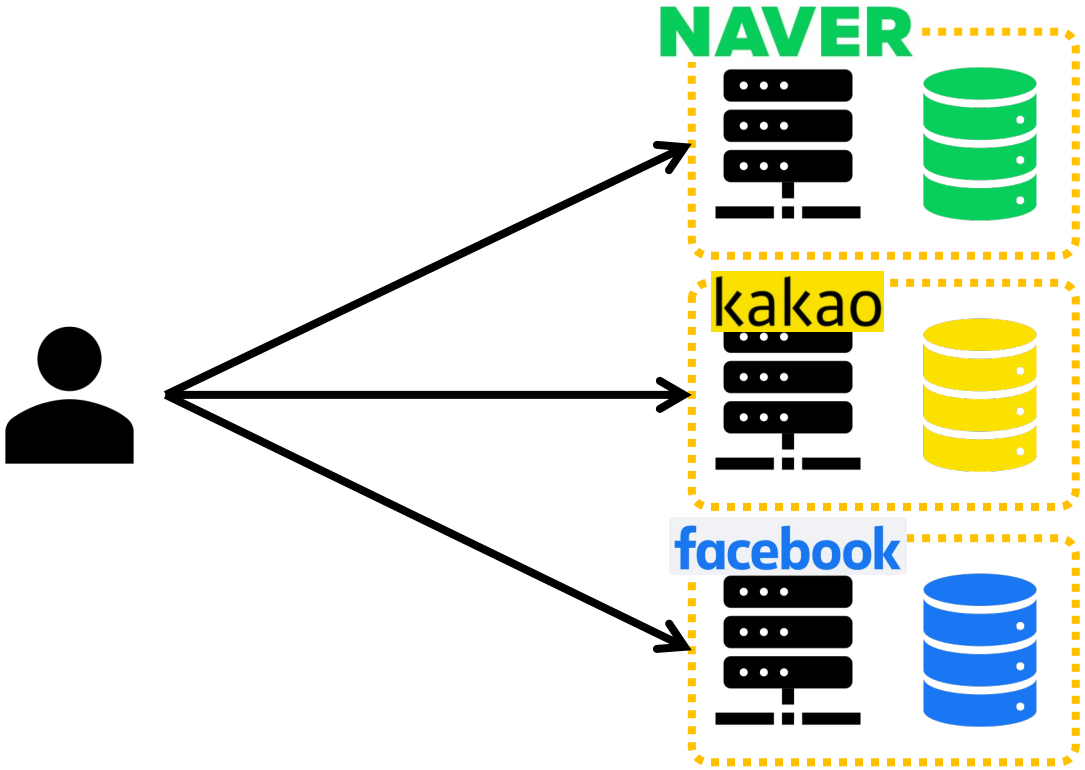


HTML로 작성된 문서



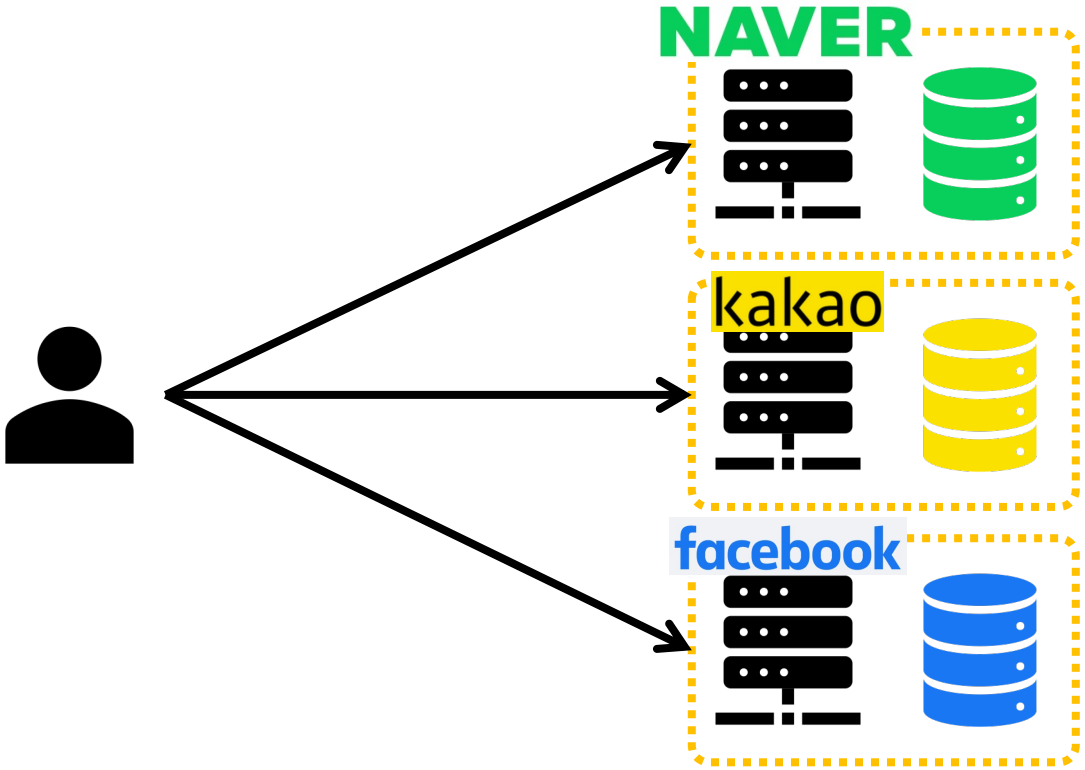
1. 인터넷

- **우리가 인터넷으로 하는 일은?**
  - 다른 컴퓨터가 갖고 있는 데이터를 읽고, 쓰고, 고치고, 지우는 것
    - 데이터: 나의 데이터? 그들의 데이터?



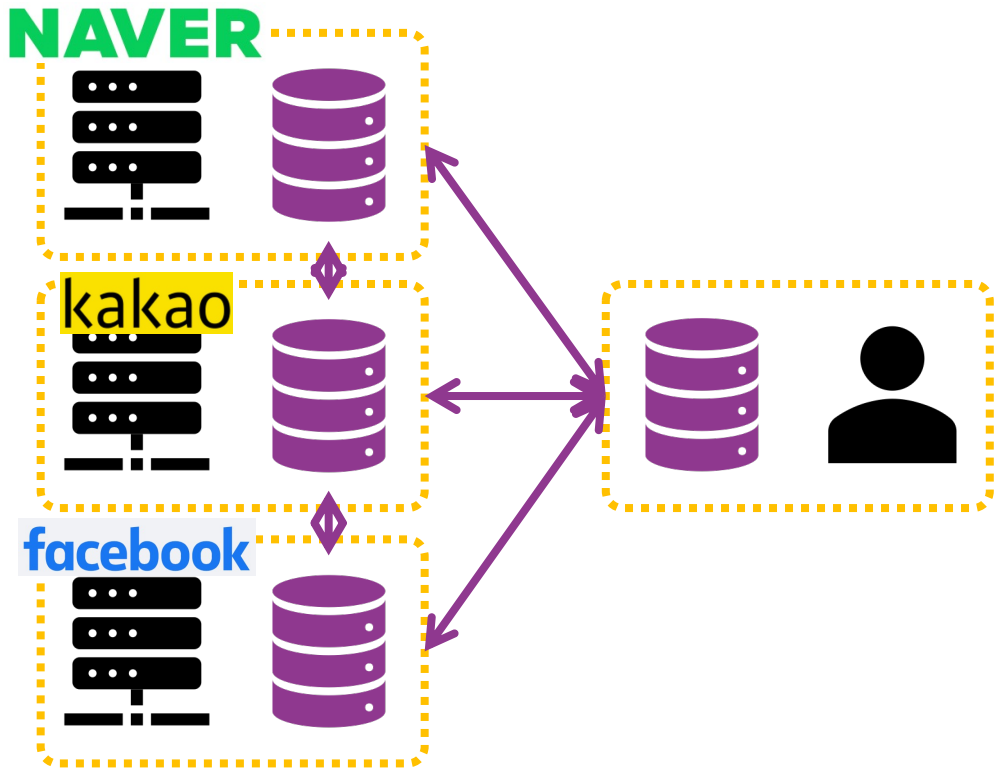
1. 인터넷

- **현재 인터넷에서의 데이터**
  - 각자의 컴퓨터에서 알아서 저장하는게 당연
  - 내 컴퓨터에 있는 데이터는 (실제 주인과 관계없이) 내 것



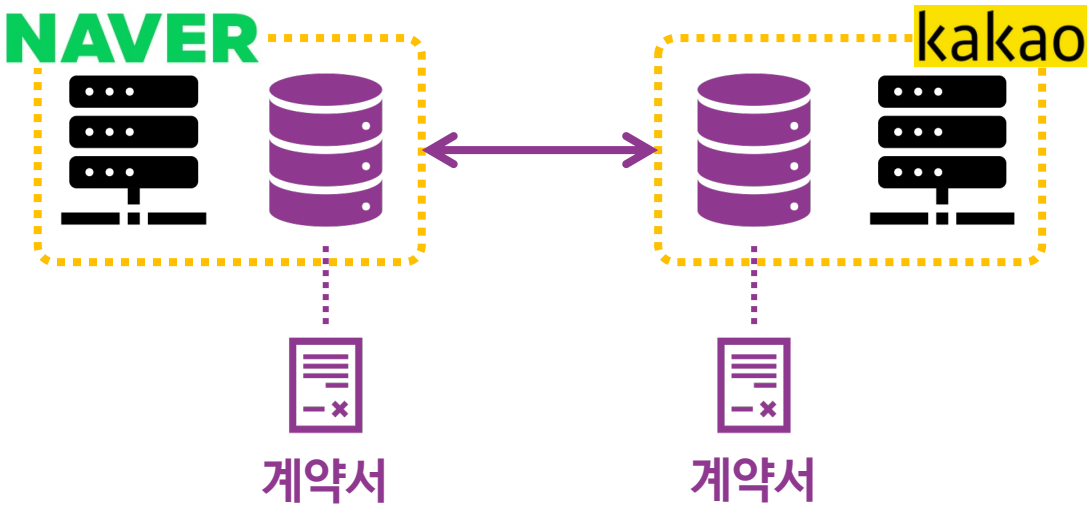
2. 블록체인 기술

- **블록체인 기반 인터넷의 데이터**
  - 각자의 데이터 -〉 모두의 데이터
  - 내 데이터는 나만 읽고, 쓰고, 고치고, 지울 수 있음



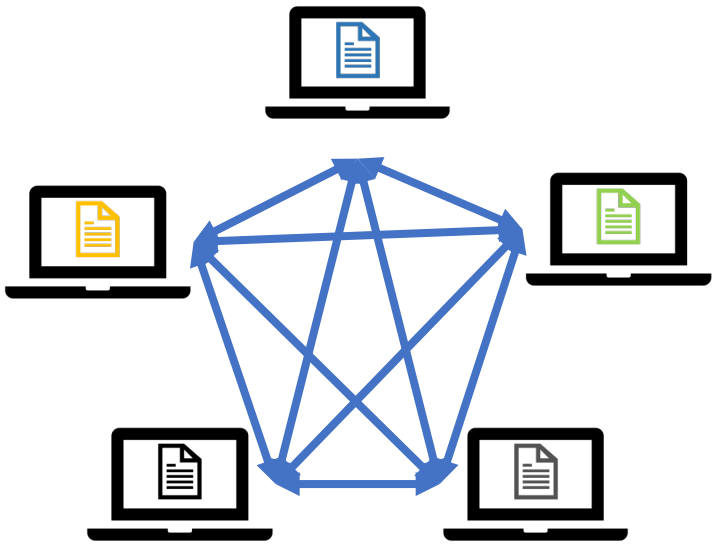
2. 블록체인 기술

- **블록체인**
  - 관계자 간에 **신뢰할 수 있는 방법**으로 **정보를 공유**하는 기술

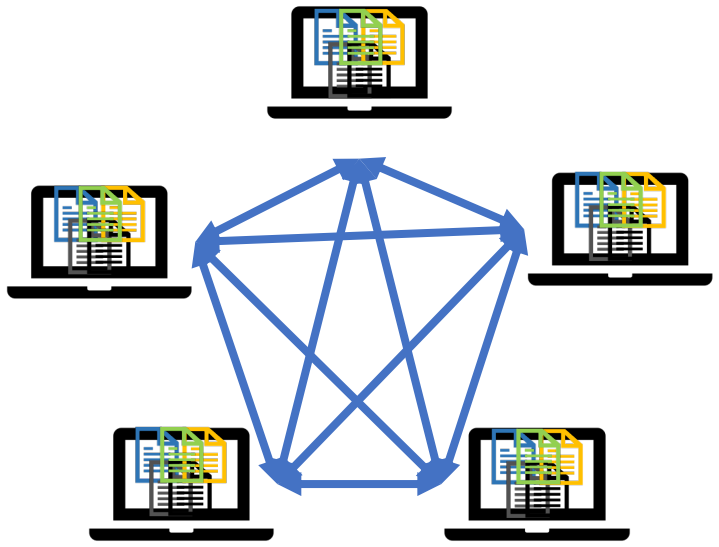


2. 블록체인 기술

- **블록체인**
  - 관계자 간에 **신뢰할 수 있는 방법**으로 **정보를 공유**하는 기술



기존 인터넷



블록체인 기반 인터넷

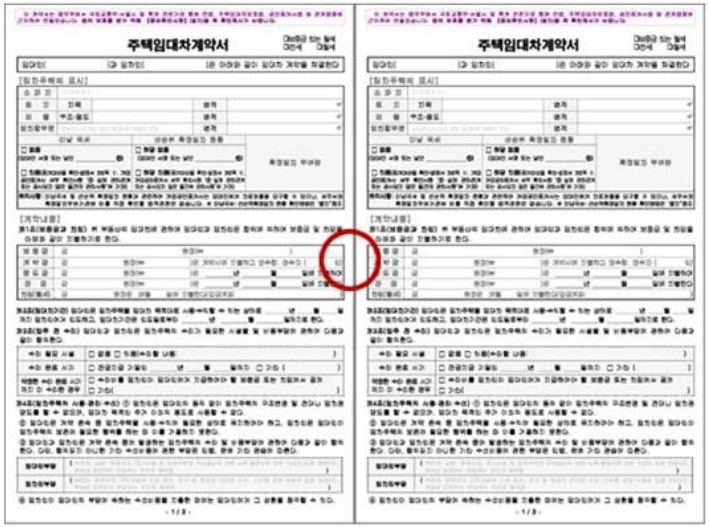
2. 블록체인 기술

인간적 신뢰가 구축되지 않은 관계자들 사이에서
중요한 데이터를
신뢰할 수 있는 방법으로 공유(저장)하는 기술

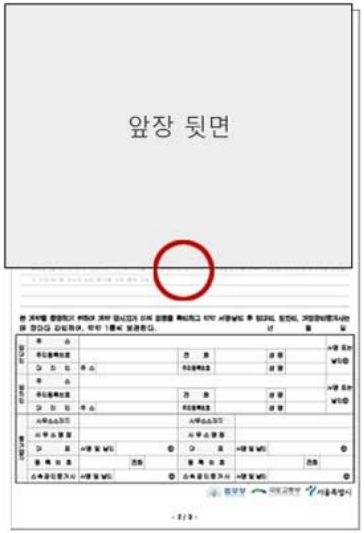
분산 데이터베이스 기술

3. 블록체인 기술의 구성요소

- **부동산 계약서 구성요소**
  - **인감도장**: 내가 바로 내가 맞는 것을 증명
  - **참여자 숫자만큼 복사한 계약서**: 실제 데이터
  - **할인**: 참여자들 각각이 갖고 있는 데이터가 모두 동일함을 확인
  - **간인**: 데이터의 순서가 중간에 위변조되지 않음을 확인



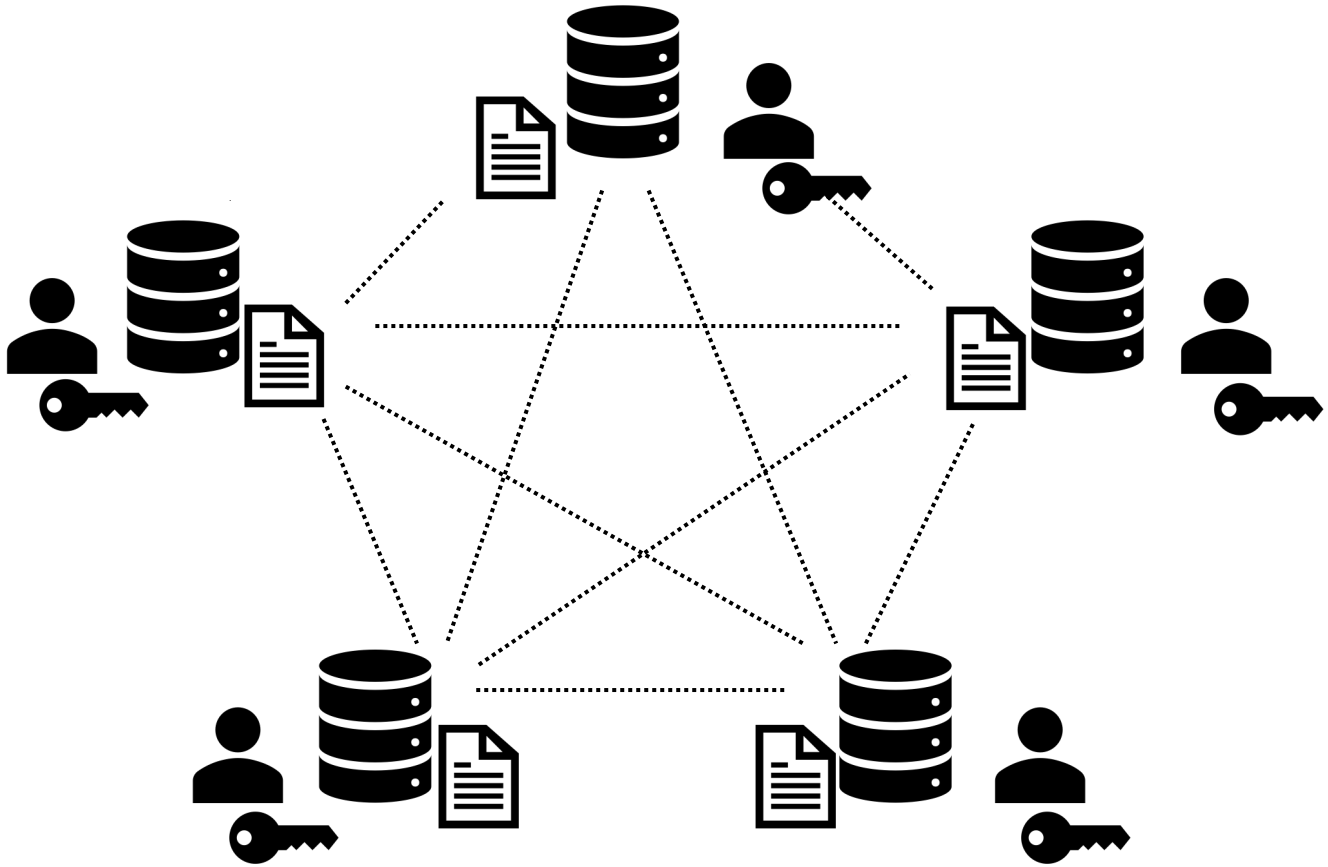
할인(割印)



간인(間印)

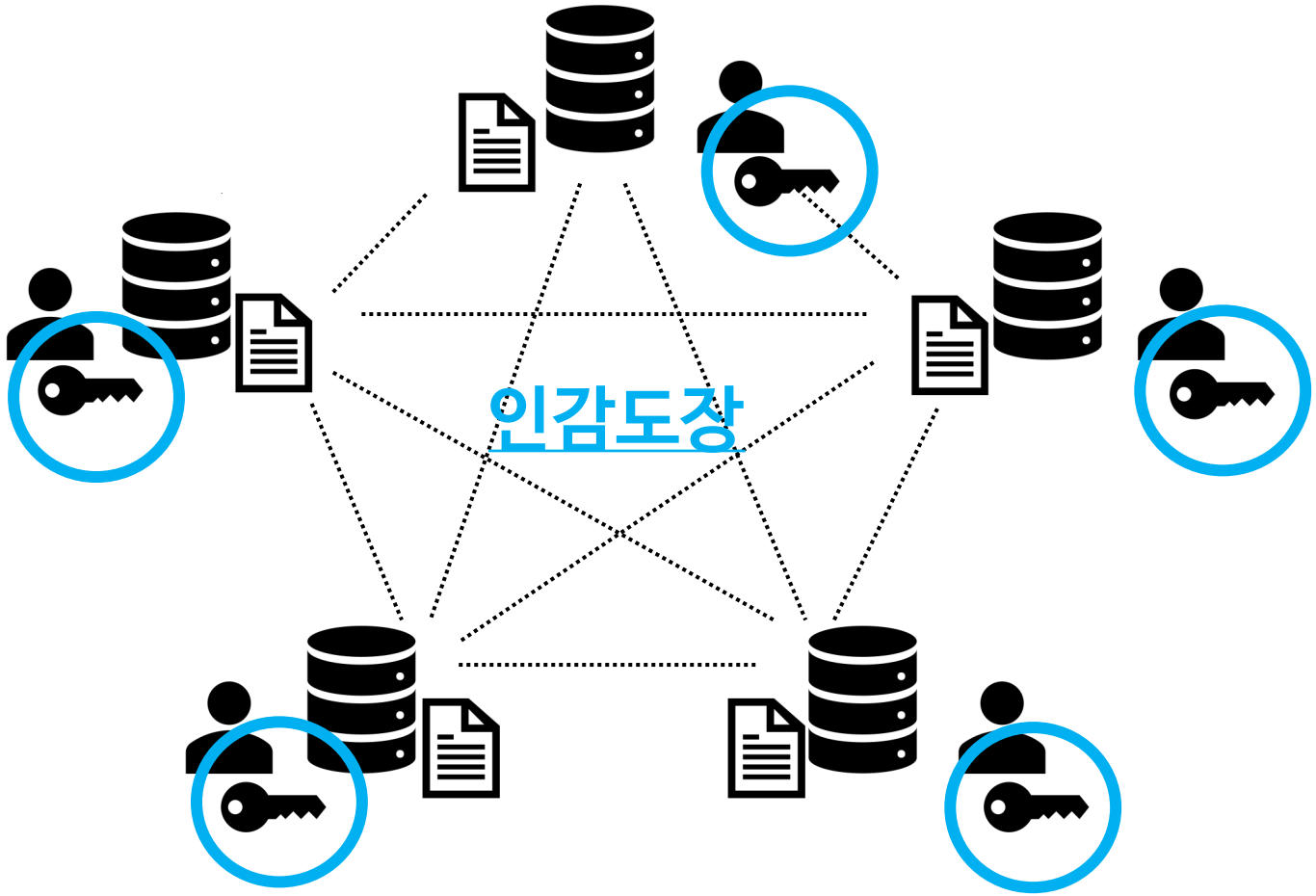
3. 블록체인 기술의 구성요소

- 인간적 신뢰가 구축되지 않은 관계자들 5명이 존재



3. 블록체인 기술의 구성요소

- 공개키 암호화 기술



3. 블록체인 기술의 구성요소

- **디지털 인감도장: 공개키 암호화 기술 (PKI)**

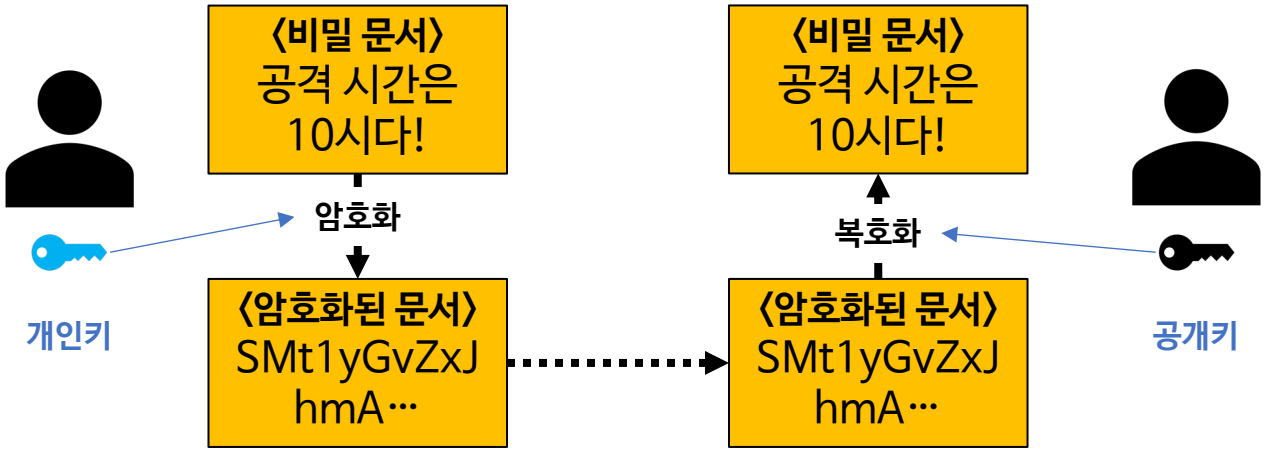
  - 암호화 기술
    - 단방향 암호화 기술
      - 복호화가 불가능

    - 비밀키 암호화 기술
      - 하나의 키로 암호화/복호화 모두 수행
      - 해당 키가 절대 유출되면 안되기 때문에 “비밀키” 암호화 기술

    - 공개키 암호화 기술
      - 고유한 2개의 키(암호화용 키, 복호화용 키)가 쌍으로 존재
      - 하나의 키로 암호화한 내용은 다른 키로만 복호화 가능

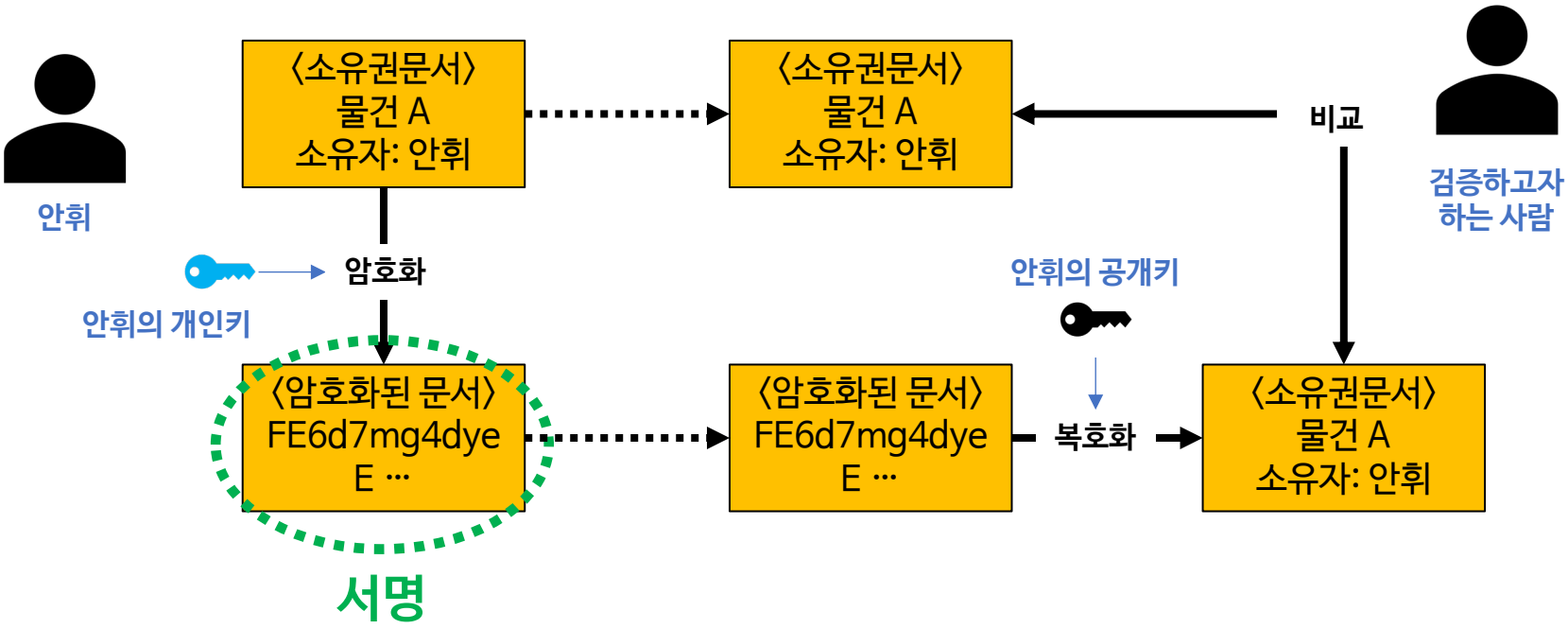
3. 블록체인 기술의 구성요소

- **디지털 인감도장: 공개키 암호화 기술 (PKI)**
  - PKI는 암호화와 복호화를 하기 위한 키를 2개 준비
    - 고유한 한 쌍의 키: 암호화용 키 1개, 복호화용 키 1개
    - 개인키로 암호화하면, 복호화는 같은 쌍의 공개키로만 가능



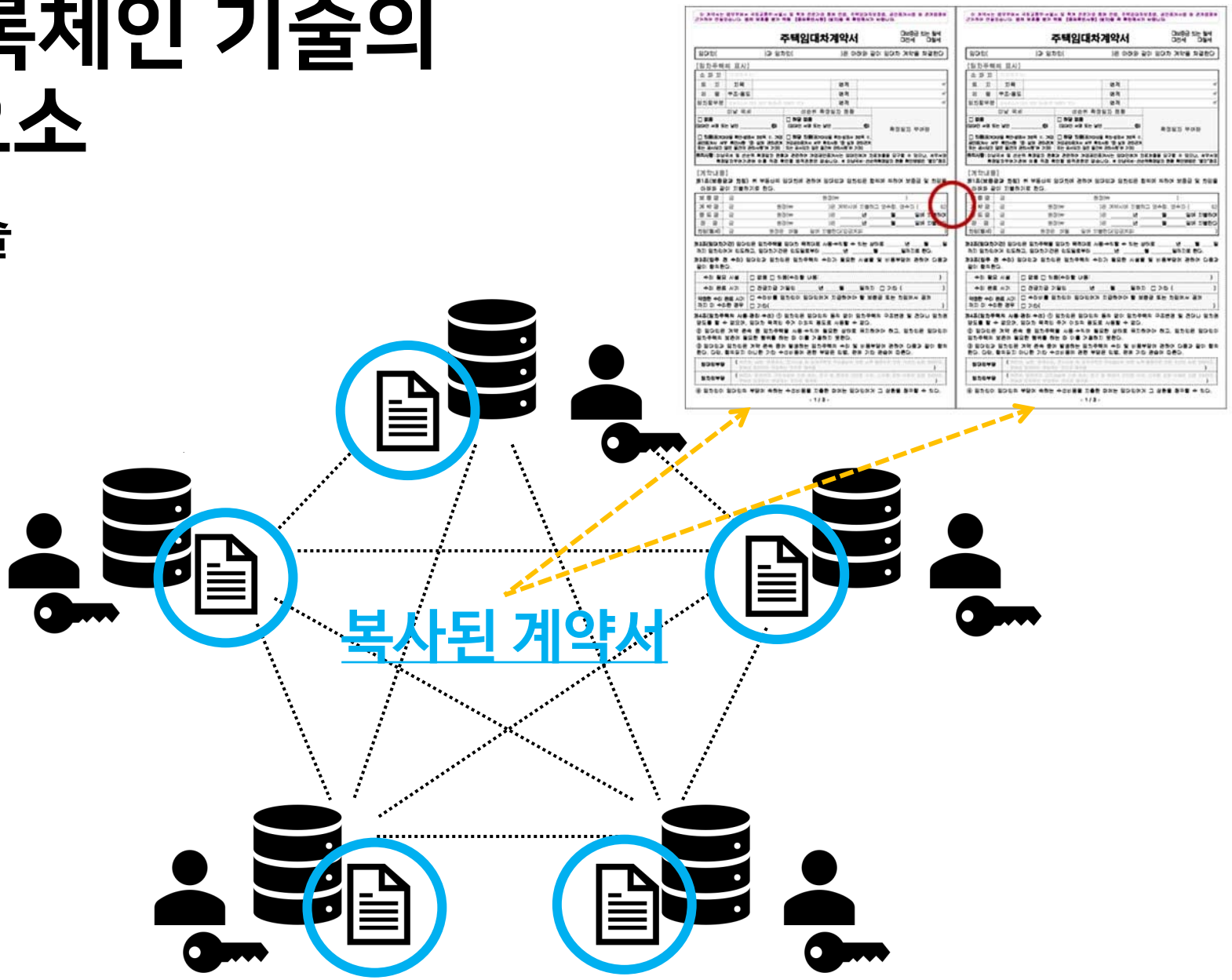
3. 블록체인 기술의 구성요소

- **디지털 인감도장: 공개키 암호화 기술 (PKI)**
  - PKI를 이용한 본인인증: "나"임을 인증하는 것 = "Private Key" 소유자임을 인증

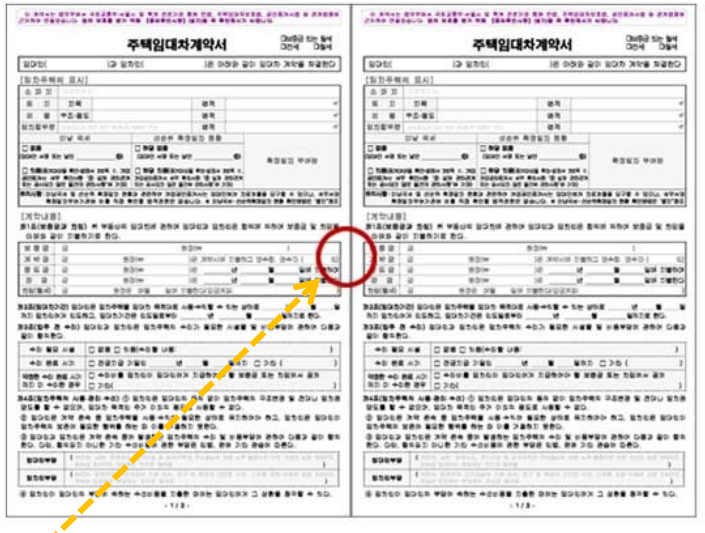


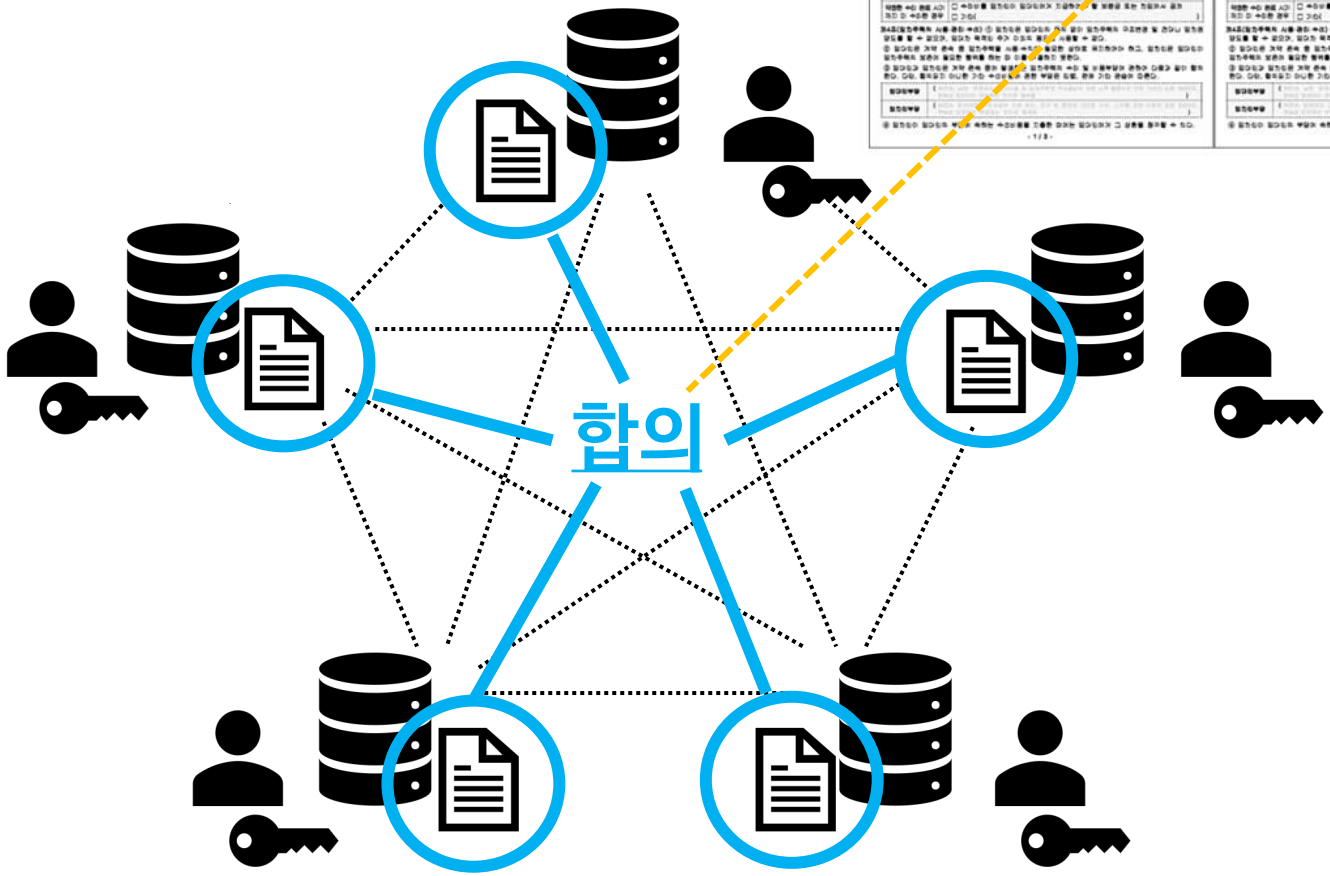
3. 블록체인 기술의 구성요소

- 합의 기술



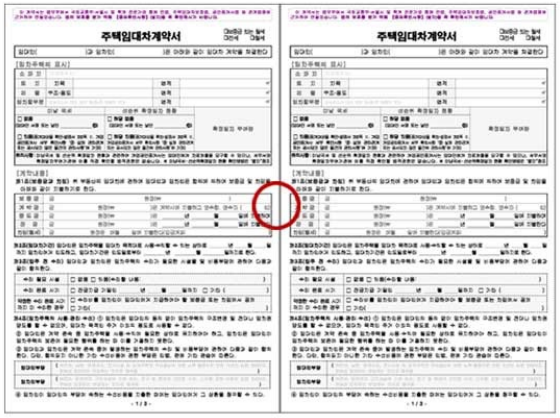
3. 블록체인 기술의 구성요소





3. 블록체인 기술의 구성요소

- **복사한 계약서 및 할인: 합의 기술 (Consensus)**
  - Proof of Work: 노력에 의한 합의
  - Proof of Stake: 지분에 의한 합의
  - Fault Tolerance Algorithms: 리더에 의한 합의



3. 블록체인 기술의 구성요소

- **Proof of Work: 노력에 의한 합의**
  - 특정 조건에 맞는 숫자(Nonce, Number used once)를 찾는 작업
    - 빠르게 찾을 수 있는 방법 없음
    - 해당 조건에 맞는 숫자를 찾을 때까지 오로지 하나하나 해보는 방법 밖에 없음

    - 먼저 정답을 찾는 사람이 갖고 있는 계약서가 진품 (블록 생성 권한을 갖음)
      → 나머지 참여자들은 해당 내용을 복사해서 가지고 있음

3. 블록체인 기술의 구성요소

- Proof of Work: 노력에 의한 합의

  - …근데 왜 이 짓(?)을 할까?
    - “블록생성권한”을 갖고, 블록을 만들면 정해진 양 만큼 보상이 주어짐
      → 보상을 “채굴”한다라고 표현
    - 네트워크가 유지되기 위해서는 누군가 문서를 검증하고, 확인하는 작업을 해주어야 함
    - 보상은 네트워크 유지에 대한 보상임
  - 비트코인, 이더리움 등 대부분 이 방식을 따르고 있음

3. 블록체인 기술의 구성요소

- **Proof of Stake: 지분에 의한 합의**
  - 토큰 보유량에 따라 블록생성권한을 조정함
  - 구현에 따라 여러 모습이 존재
    - 토큰 보유량에 따라 블록생성 및 검증을 수행할 대표자를 선정
    - 토큰 예치 등을 통해 권한 부여
    - …
  - Proof of Work 의 과도한 에너지 사용 문제를 해결
  - 이더리움이 향후 이 방향으로 진화할 예정

3. 블록체인 기술의 구성요소

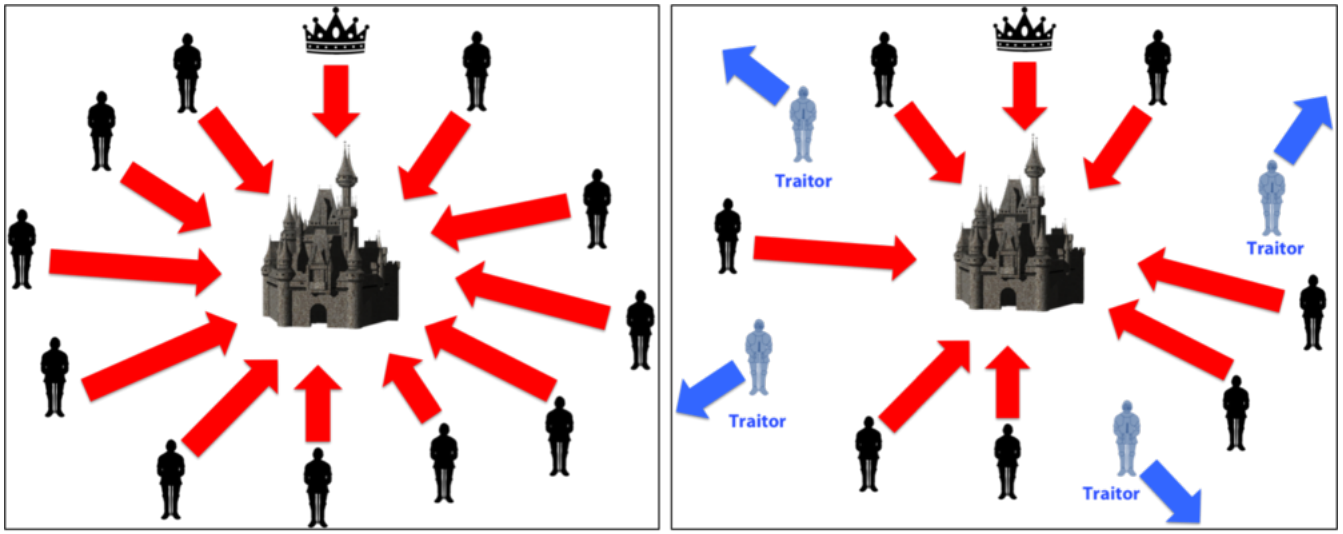
- **Fault Tolerance Algorithms: 리더에 의한 합의**

  - Fault Tolerance Algorithms
    - 참여자 중 일부가 불능(fault) 상태가 되더라도, 네트워크를 유지시키는 알고리즘
    - 예:
      - 3대의 컴퓨터가 동일한 내용을 검증하여 저장 중
      - 1대의 컴퓨터가 갑자기 불능이 됨
      - 그래도 내용이 올바르게 검증되었다는 것을 보장함

    - Crash fault vs Byzantine fault
      - Crash fault: 참여자 중 일부가 불능 상태에 빠짐
      - Byzantine fault: 참여자 중 일부가 “배신자”가 됨

3. 블록체인 기술의 구성요소

- Fault Tolerance Algorithms: 리더에 의한 합의
  - Byzantine Fault



Coordinated Attack Leading to Victory　　Uncoordinated Attack Leading to Defeat

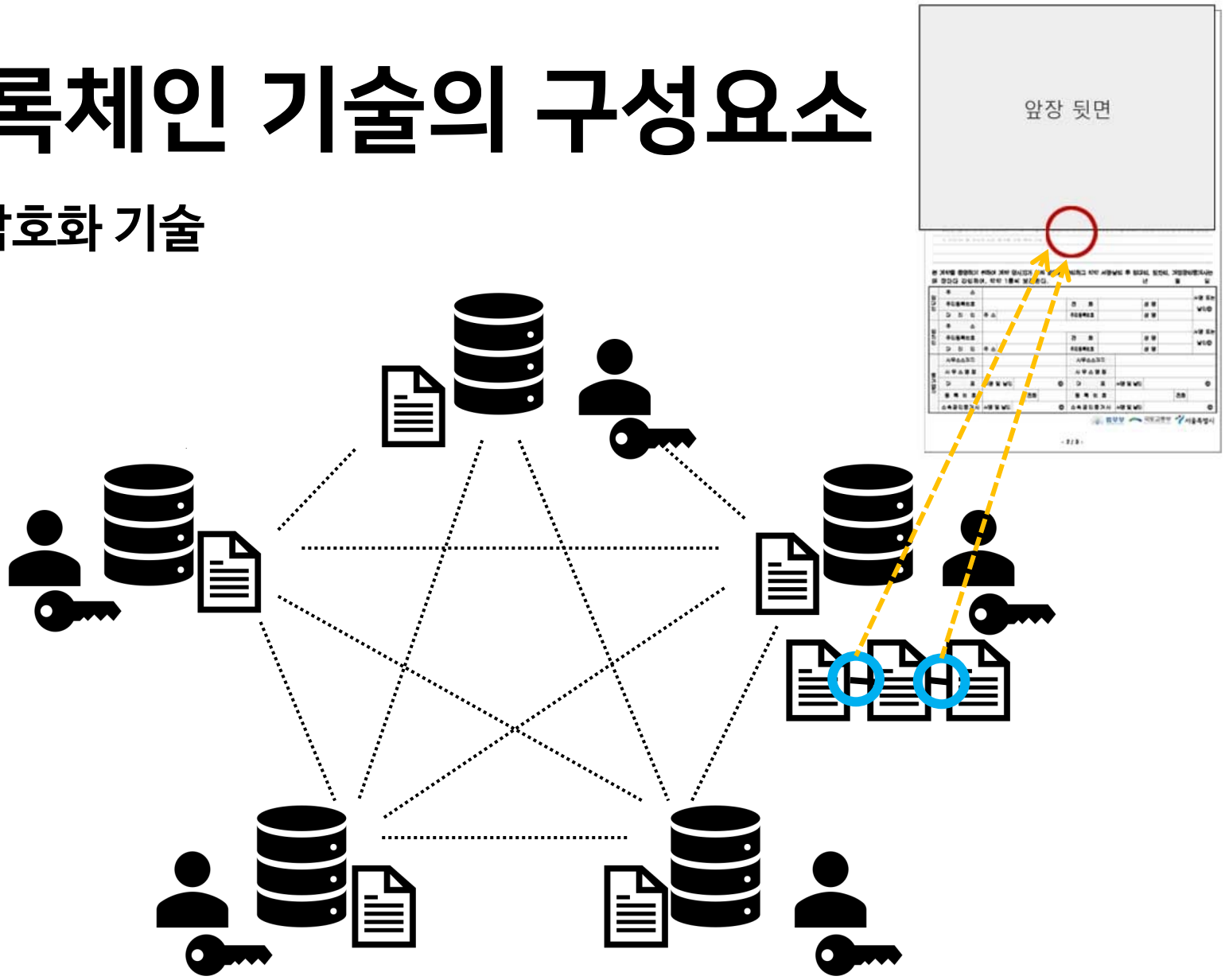
3. 블록체인 기술의 구성요소

- **Fault Tolerance Algorithms: 리더에 의한 합의**

  - Fault Tolerance Algorithms
    - 일반적으로 1/2(Crash fault), 또는 1/3(Byzantine fault)의 참여자가 불능상태에 빠져도 합의가 올바르게 진행되는 것을 보장함
    - 소수의 참여자들이 빠르게 합의하기 때문에 다른 합의 대비 매우 빠르고, 에너지 소모가 거의 없음
    - 보통 참여자 숫자가 20~30 정도면 한계에 도달
    - 프라이빗 블록체인에서 주로 사용
    - 현재 기술은 Crash Fault 수준

3. 블록체인 기술의 구성요소

- 단방향 암호화 기술



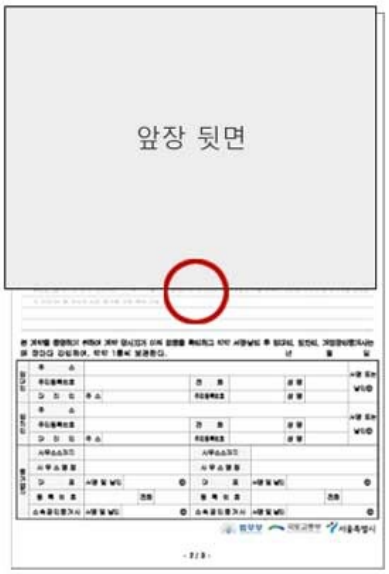
3. 블록체인 기술의 구성요소

- 간인: 해시 기술 (Hash)
  - 해시
    - **단방향 암호화** 기술
    - 특정 문서를 복구 불가능한 문자열로 암호화해줌

Hello, World! → (해시!) → DFFD6021BB2BD5B0AF676290809EC3A53191DD81C7F70A4B28688A362182986F

Hello, World? → (해시!) → F16C3BB0532537ACD5B2E418F2B1235B29181E35CFFEE7CC29D84DE4A1D62E4D

- 원래는 비밀번호 저장, 문서 위변조 방지에 사용됨



3. 블록체인 기술의 구성요소

- 간인: **해시 기술 (Hash)**
  - 이전 데이터의 해시값을 함께 저장함

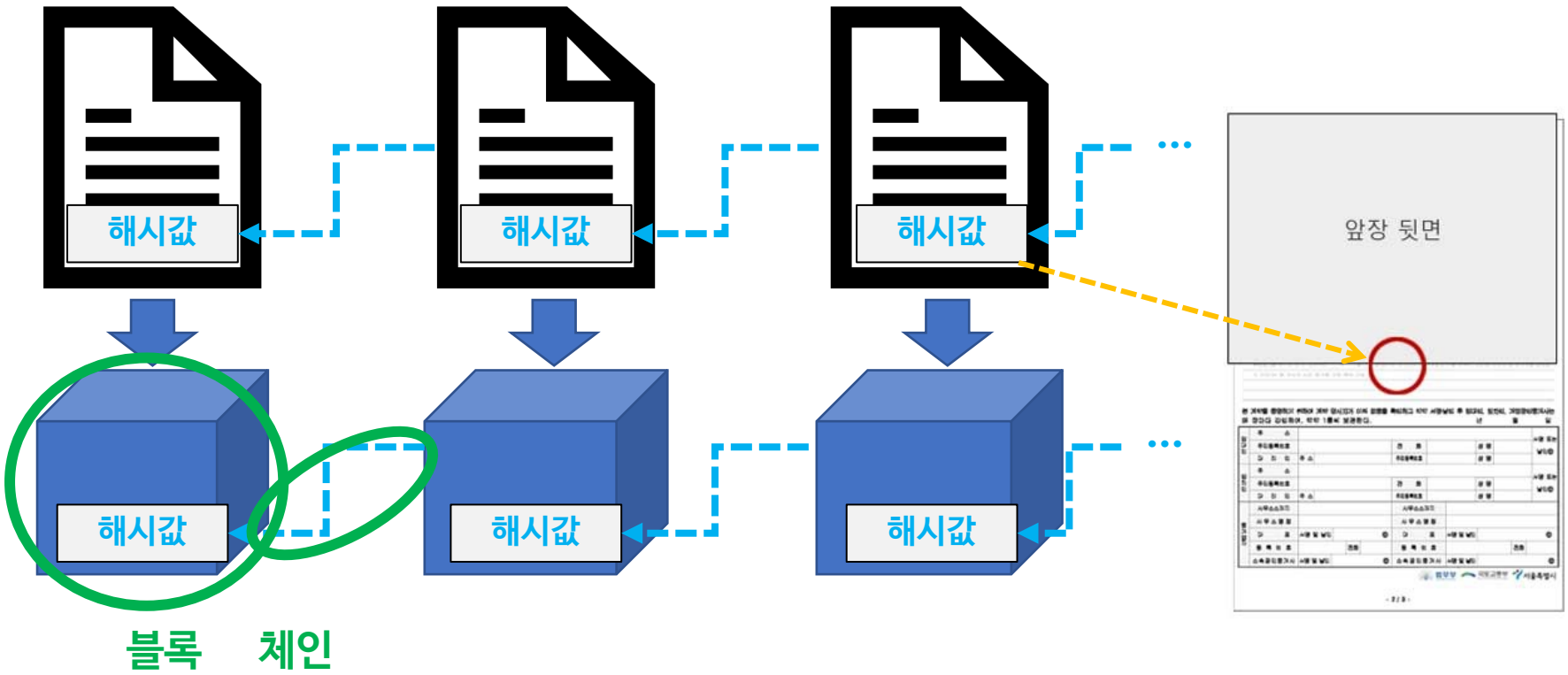
3. 블록체인 기술의 구성요소

- 간인: 해시 기술 (Hash)

- 저장된 데이터를 위조하려면, 그 뒤에 저장한 모든 데이터를 위조해야 함
  -〉 위변조가 현실적으로 불가능

3. 블록체인 기술의 구성요소

- 간인: 해시 기술 (Hash)



4. 블록체인 기술이란

- 인감도장: 공개키 암호화 기술 공인인증서 기술
- 복사한 계약서 및 할인: 합의 기술 네트워크 기술
- 간인: 단방향 암호화 기술 비밀번호 저장 기술

- 모든 건 이미 있던 기술… 혁신은 아무것도 없는 곳에서 갑자기 오는 것이 아니라, 이미 있던 것들이 조합되어 하나의 서비스가 될 때 이루어짐.
- “내가 그의 이름을 불러주기 전에는 그는 다만 하나의 몸짓에 지나지 않았다. 내가 그의 이름을 불러주었을 때 그는 나에게로 와서 꽃이 되었다” (김춘수, 꽃)

4. 블록체인 기술이란

- **장점**
  - 위변조 불가
    - 저장된 데이터는 위변조에 매우 강함
    - 사람이 갖는 공간적, 시간적 한계를 뛰어넘기 때문에, 체인 형태로 구성된 데이터를 모두 위조하는건 “현실적”으로 불가능
    - 공공 목적으로 활용될 때 가장 고려할만한 장점

  - 투명성
    - 모든 참여자들이 데이터를 나누어가짐
    - 지금도 전세계 비트코인 거래 내역은 투명하게 모두가 볼 수 있음
    - 암호화폐 거래를 통해 자신을 숨길 수 있다는 것은 허상
      (자신의 공개키를 아는 순간 전세계 모든 사람이 거래내역을 파악할 수 있음)
    - 프라이버시와 다름을 명확히 이해할 것

4. 블록체인 기술이란

- **한계**
  - 프라이버시
    - 사실 블록체인과 관계없음.
    - 프라이버시는 서비스 설계에서 논할 일이지, 블록체인을 쓴다고 해결되는 문제가 아님!
    - 그래서 투명성이 장점인 블록체인에 개인정보를 올리는건 “매우 섬세한” 서비스 설계가 필요한 일

  - 성능
    - 블록체인은 성능 관점에서 제약이 심한 시스템: 느리고, 자원을 많이 차지함.
    - 민간 기업 도입이 무산되는 가장 큰 이유
    - 투명성, 위변조 불가를 통해 공공의 목적을 달성할 경우, 성능 제약에도 도입 가능
      =〉 블록체인이 공공 영역에서 더 논의되어야 하는 이유!

블록체인 기술들과 하이퍼레저 패브릭

2022-02-14

빅픽처랩(주)

안휘

목차

1. 비트코인
2. 이더리움
3. 하이퍼레저 패브릭

1. 비트코인

Bitcoin: A Peer-to-Peer Electronic Cash System

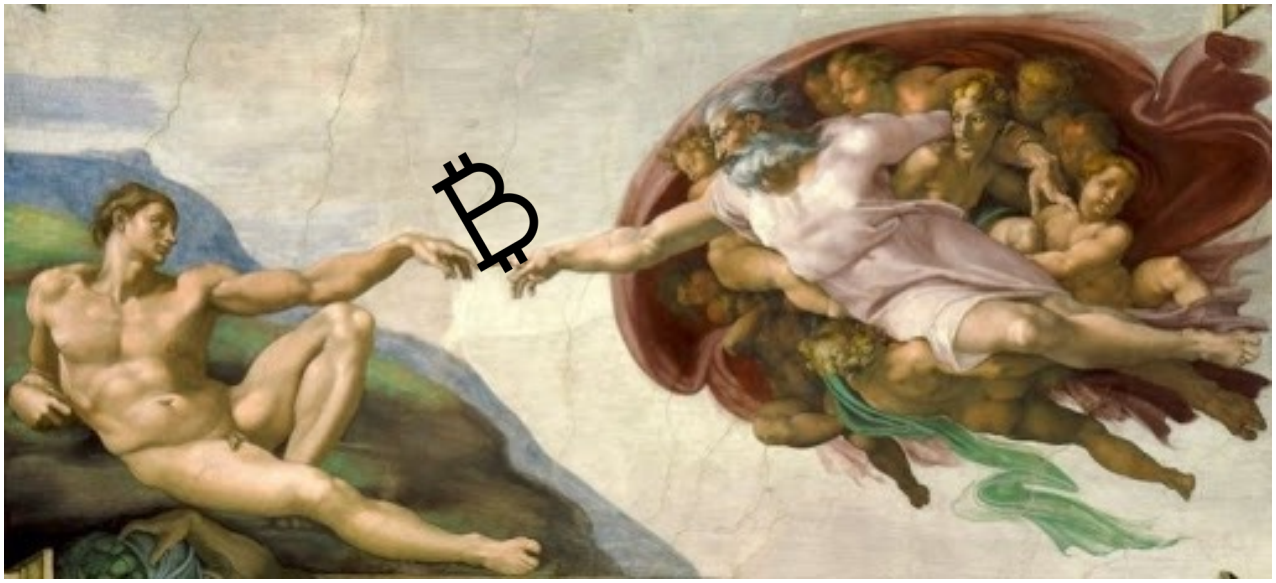
Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the

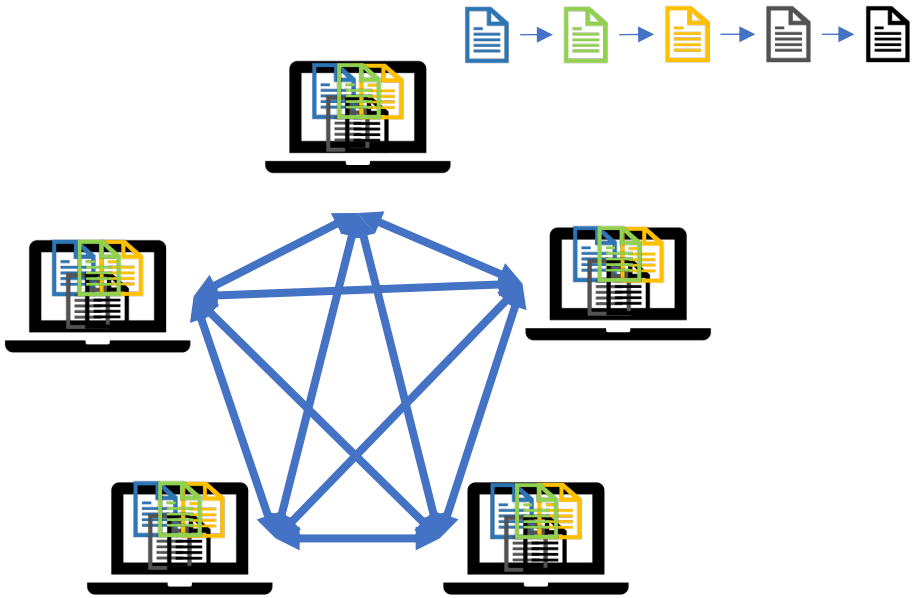
1. 비트코인



- 2008년 10월, 논문이 보안 학회 쪽 메일링 리스트에 링크됨
- 2009년 1월, 오픈소스로 첫 비트코인 프로그램이 개발되어 공개됨
- 2009년 1월, 비트코인의 첫 블록이 사토시 나카모토에 의해 생성됨

1. 비트코인

- **“신뢰할 수 있는 방법으로 정보를 공유하기 위한” 사토시 나카모토의 제안**
  - 정보 저장 원칙 -〉 모두가 동일한 카피를 갖는다
  - 정보 저장 형태 -〉 체인 형태로 위변조를 막는다
  - 정보 공유 형태 -〉 채굴이라는 합의 과정을 통해 공유한다



1. 비트코인

- **그래서… 그런 기발한 방법으로 공유할 정보는 무엇이죠?**
  -〉 역시 가장 신뢰가 필요한 중요한 정보는…

나의 은행 잔고!

- 은행 잔고는 나의 입출금 내역의 합산
- 비트코인이 저장 & 공유하는 정보는 **“비트코인의 입출금 내역”**

2. 이더리움

- **이더리움**
  - 비트코인: 블록체인에 “입출금 기록”만 저장 가능
  - “무엇을 저장할지, 개발자들이 결정하도록 하자”
  - 블록체인 세상의 “**앱 스토어**”가 되겠다
    - 앱 = **스마트 컨트랙트**

- 2013년 비탈릭 부테린이 백서를 작성하여 개발을 제안
- 2014년 이더리움 재단 설립 및 개발 자금 펀딩
- 2015년 7월 30일 비탈릭 부테린에 의해 개발됨
- 2020 ~ 2021년: 이더리움 2.0 진화 예정

2. 이더리움

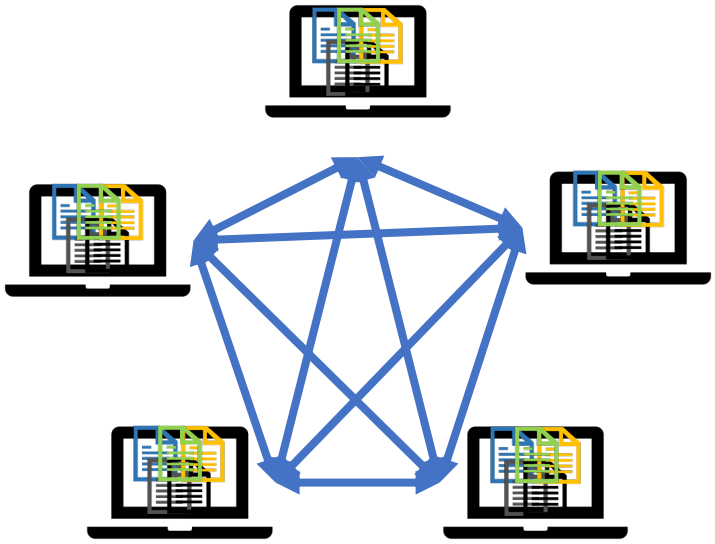
- **이더리움**
  - 저장 원칙, 형식, 공유 방식은 비트코인과 동일
  - 기반에 비트코인과 동일하게 이더 라는 암호화폐가 법정 화폐처럼 존재

  - 가장 개발자 친화적인 블록체인
  - 가장 다양한 시도가 이루어지고 있음
    - 대부분의 알트 코인들은 이더리움의 스마트 컨트랙트
    - DeFi, NFT 등도 모두 이더리움의 스마트 컨트랙트
    - 앞으로도 수많은 시도가 이루어질 것임

3. 하이퍼레저 패브릭

- **프라이빗 블록체인**
  - 비트코인, 이더리움:
    언제든, 누구든, 원하면 블록체인에 저장된 정보를 공유받고 자신의 컴퓨터에 저장할 수 있음
    -〉 퍼블릭 블록체인

  - 프라이빗 블록체인은 “정해진 컴퓨터들끼리만” 정보를 공유하고 저장함



3. 하이퍼레저 패브릭

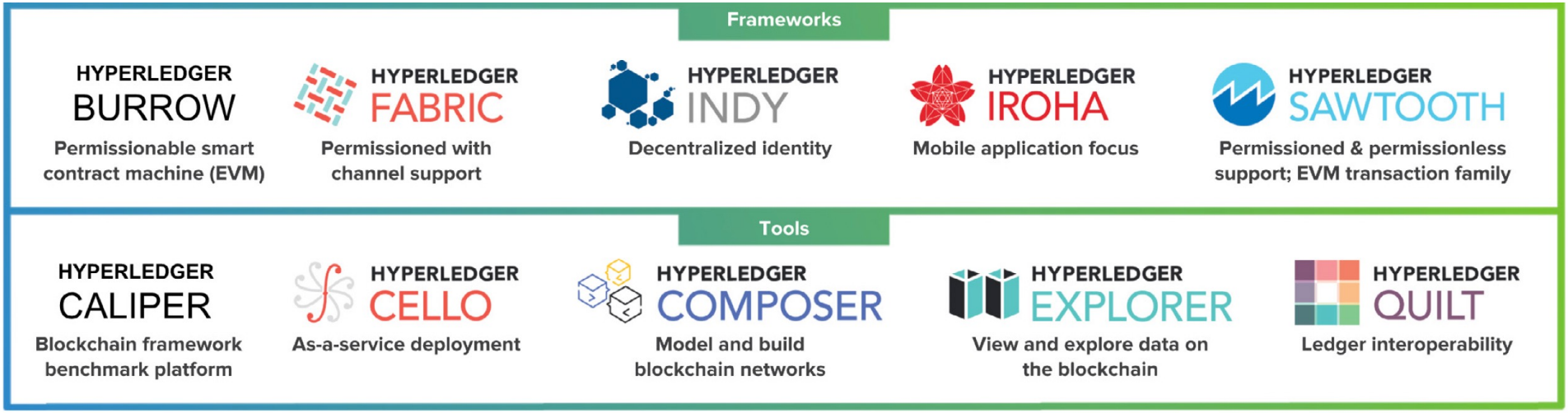
- **프라이빗 블록체인 소프트웨어**
  - RDBMS: PostgreSQL, MySQL, MariaDB, ...
  - NoSQL: MongoDB, Redis, ...
  - **Private Blockchain:** Hyperledger Fabric, GoQuorum, Hyperledger Besu, ...

- **데이터베이스 소프트웨어들이 개발자들에게 제공해주는 것**
  - 데이터베이스 인스턴스 실행, 관리
  - 데이터베이스 SDK
    - 데이터베이스 접속 SDK
    - 데이터베이스 모델 정의 SDK
    - 데이터베이스 CRUD SDK
    - ...

=〉 하이퍼레저 패브릭도 동일

3. 하이퍼레저 패브릭

- Hyperledger
  - 기업용 블록체인 기술을 연구, 개발하는 오픈소스 커뮤니티
  - 운영: Linux Foundations
  - 참여:
    - IBM, 인텔, JP 모건, 후지쯔, 히타치, 바이두, 화웨이, 보쉬, 다임러, LG CNS, 삼성 SDS, 월마트 등 174개 기업/조직/학교



이미지: Hyperledger Architecture, Vol II.

3. 하이퍼레저 패브릭

- “Hyperledger Fabric is an open source enterprise-grade permissioned distributed ledger technology (DLT) platform, designed for use in enterprise contexts, that delivers some key differentiating capabilities over other popular distributed ledger or blockchain platforms.”
- **하이퍼레저 안에서 가장 먼저 1.0 버전을 론칭 (2017.07)**
  - 1.0: 2017.07
  - 1.4 (LTS): 2019.01
  - 2.0: 2020.01
  - 2.1: 2020.04
  - 2.2 (LTS): 2020.7
  - 2.3: 2020.11
  - 2.4 (LTS): 2021.11
  - **2.4.2/2.2.5: 2022.01**

3. 하이퍼레저 패브릭

- **버전별 특징**
  - 1.4.x (현재 1.4.12 - 2021.04)
    - 각종 어드민 관리용 API들이 매우 막강함
  - 2.0 ~ 2.2 (현재 2.2.5 - 2022.01)
    - 2.0, 2.1: 쓰지 말것 (실험적인 릴리즈)
    - 체인코드(스마트 컨트랙트) 거버넌스 개편
    - Alpine Linux 기반 컨테이너
  - 2.3 ~ 2.4
    - 2.3: 쓰지 말것 (2.4에 모두 포함되어 있음)
    - 참여/합의 노드들 관리 개편
    - 트랜잭션 전송 플로우 개편
      - 매우 중요하여, 사실 2.4 버전 이전은 안 쓰는 것을 추천

3. 하이퍼레저 패브릭

- **하이퍼레저 패브릭의 특징**
  - Highly modular & configurable
    - Enterprise의 다양한 요구에 대응하기 위함
    - 하이퍼레저 패브릭을 복잡하게 하는 주 원인
  - 범용 언어로 스마트컨트랙트 작성이 가능
    - Go, Java, JavaScript
  - Permissioned
    - 참여 노드들의 멤버쉽 관리
  - Pluggable consensus protocol
    - 큰 의미 없고, RAFT 라고 생각하면 됨
  - NOT require native cryptocurrency
    - Ethereum 프라이빗 버전을 쓰지 않고 하이퍼레저 패브릭을 선택하는 주된 이유

하이퍼레저 패브릭 구조 개요

2022-02-14

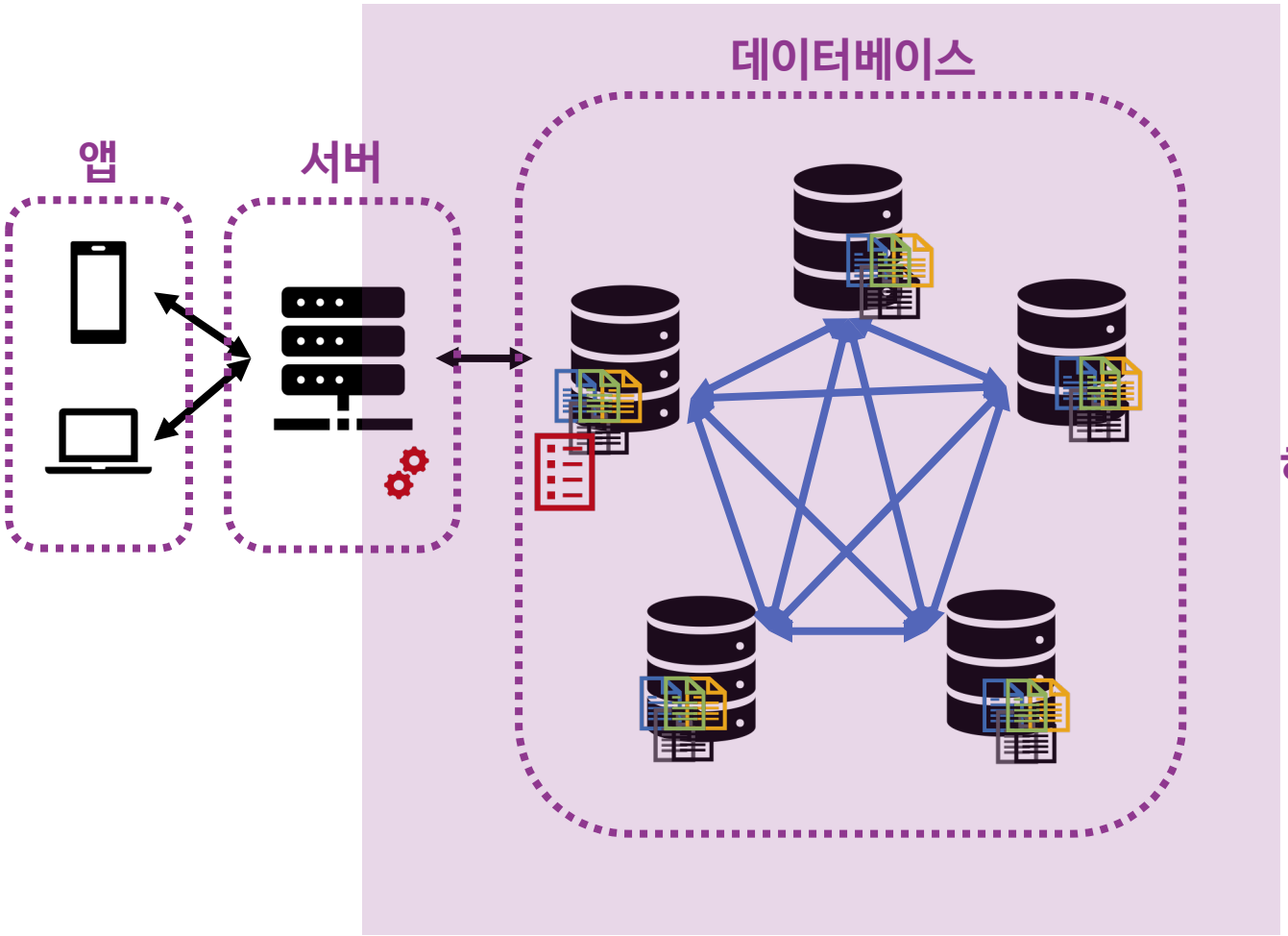
빅픽처랩㈜

안 휘

목차

- 하이퍼레저 패브릭의 주요 개념들
- 하이퍼레저 패브릭의 구성 노드
- 노드 멤버쉽
- 트랜잭션 플로우
- 데이터베이스
- 구동 환경

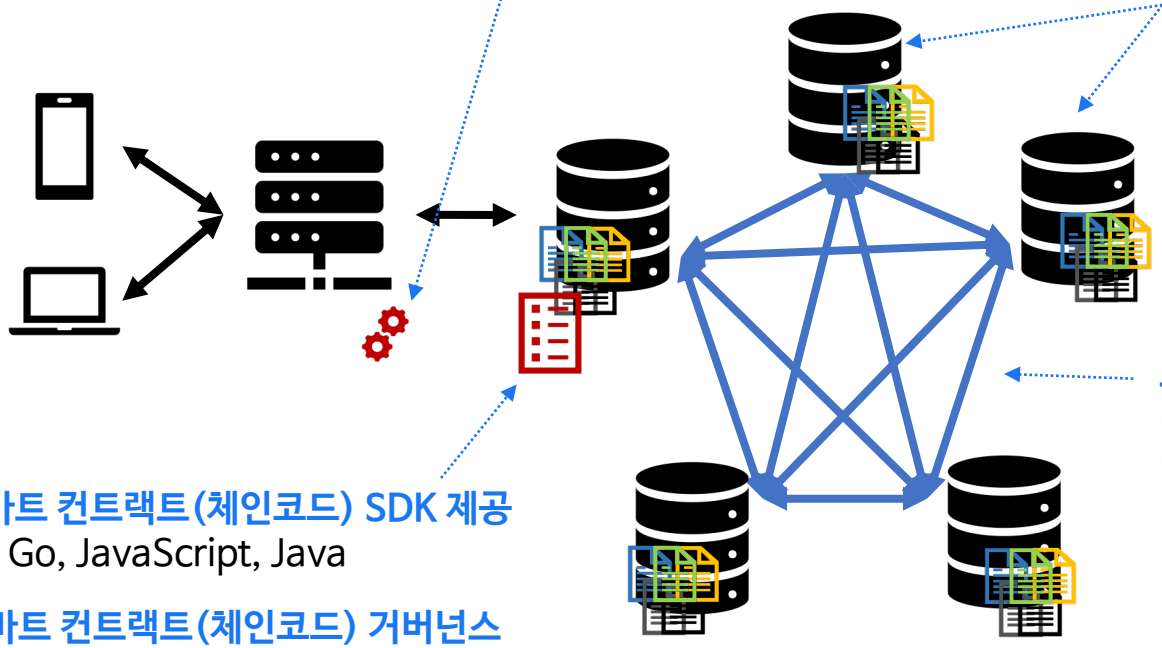
1. 하이퍼레저 패브릭의 주요 개념들



하이퍼레저 패브릭의 영역

1. 하이퍼레저 패브릭의 주요 개념들

서버 SDK 제공
- Go, JavaScript(node.js), Java



정해진 네트워크 노드들만 참여
- 멤버쉽 관리: MSP
- 채널

트랜잭션 생성과 합의
- RAFT 알고리즘
- Endorsement 기반 플로우

스마트 컨트랙트(체인코드) SDK 제공
- Go, JavaScript, Java

스마트 컨트랙트(체인코드) 거버넌스
- 합의된 배포

노드들의 데이터베이스
- 블록체인
- State DB (levelDB, CouchDB)

1. 하이퍼레저 패브릭의 주요 개념들

- **중요 개념들**
  - 노드 구성
    - 피어(Peer): 데이터 저장, 스마트 컨트랙트 실행 및 검증
    - 오더러(Orderer): 합의 수행
  - 노드 멤버쉽
    - Membership Service Provider: PKI 인증서 기반 검증
    - 채널(Channel): 데이터를 공유하는 노드들의 그룹
  - 트랜잭션 플로우
    - 합의: RAFT 알고리즘 (Crash Tolerance, ½ 보장 → 최소 3개 노드 요구)
    - Endorsement(승인) 기반 트랜잭션 전송 플로우
  - 데이터 저장
    - StateDB: 가장 최신 스냅샷. LevelDB 또는 CouchDB
    - 블록체인: 파일에 바이트로 직접 씀

2. 하이퍼레저 패브릭의 구성 노드

- **노드(Node)**
  - 네트워크로 통신하며 하이퍼레저 패브릭을 구성하는 '일종의' 서버들
  - gRPC로 통신하며, Go 언어로 작성된 고성능 서버
  - 모두 도커 이미지로 제공됨

피어 노드(피어, Peer)

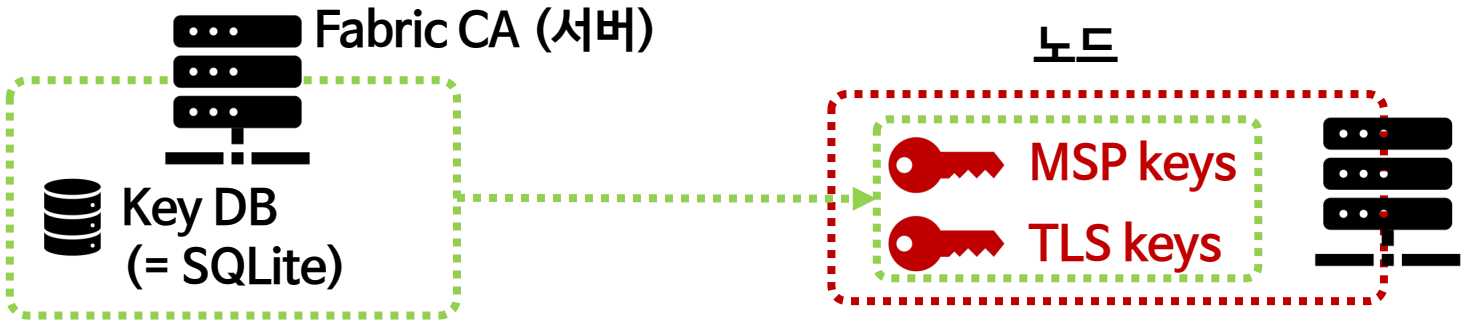
체인코드 실행
블록 내 트랜잭션 검증
블록 저장
State DB 에 대한 CRUD

Ordering Service 노드(오더러)

블록 생성
(트랜잭션들의 순서를 "합의"하여 블록을 생성)

3. 노드 멤버쉽

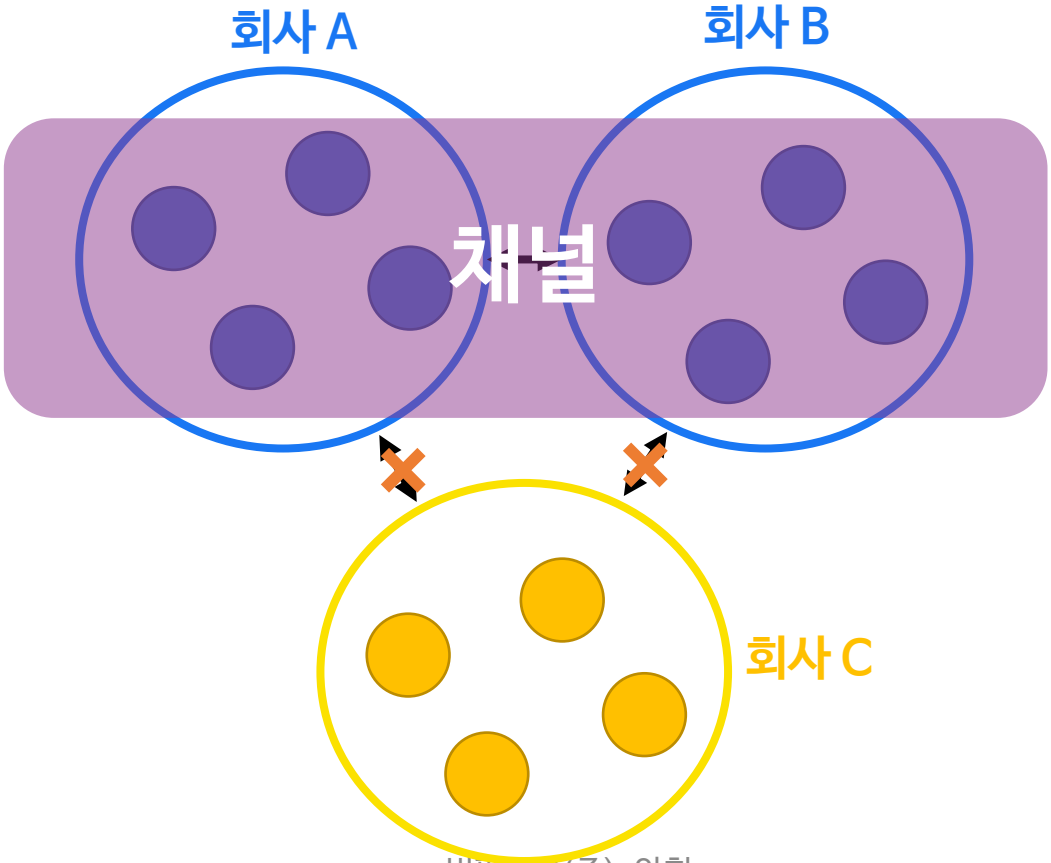
- Membership Service Provider (MSP)
  - 조직/기관이 하이퍼레저 패브릭 기반 프라이빗 블록체인 네트워크에 참여한다는 의미는?
  - 조직/기관이 자신의 노드(피어 또는 오더러)들을 갖고 네트워크에 참여한다는 것
  - 다른 네트워크 구성원들은 이 노드들이 허가된 조직/기관의 것임을 어떻게 알 수 있는가?
    =〉 공개키 암호화 기술 사용



  - 하이퍼레저 패브릭 노드에 멤버쉽을 제공해주는 서비스
  - MSP는 노드의 파일시스템에 .key, .cert 등과 같은 파일로 저장되며, 노드의 비밀키, 발급한 CA의 인증서(공개키) 등이 저장됨암호화된 통신(gRPCs)을 위한 TLS 키도 필수

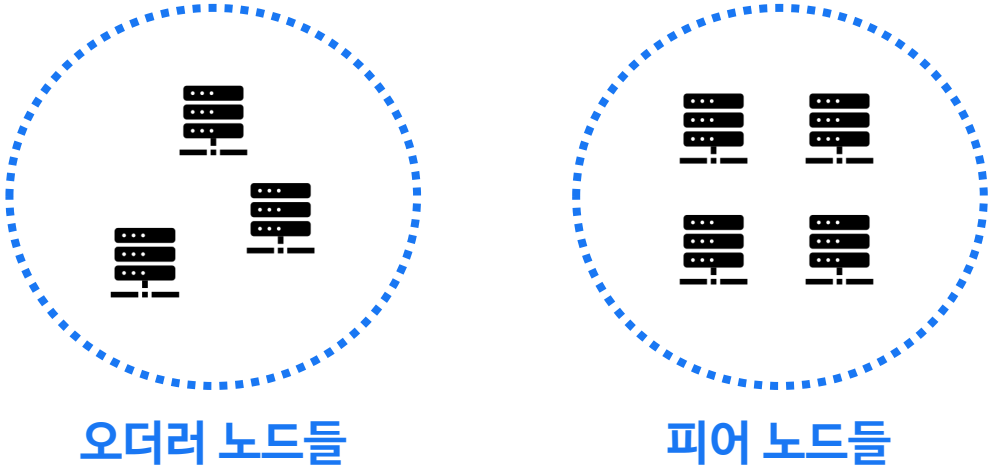
3. 노드 멤버쉽

- **채널 (Channel)**
  - 데이터를 공유하는 노드들의 그룹
  - 채널 블록체인의 제네시스 블록에는 채널 멤버쉽 관련 인증서가 담긴다



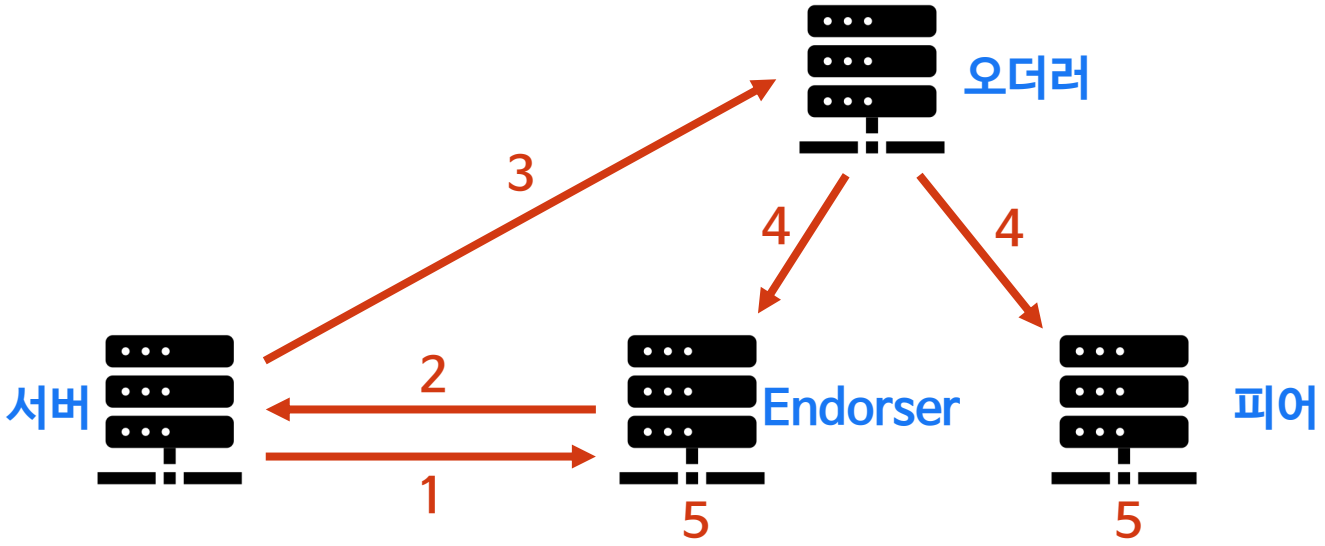
4. 트랜잭션 플로우

- **합의**
  - RAFT 알고리즘
  - Crash Tolerance
  - 최소 ½ 이상 노드가 생존해 있으면 네트워크 작동을 보장함
  - 최소 3개의 오더러 노드가 필요하며, 홀수개로 배포하는 것을 추천함



4. 트랜잭션 플로우

- **Endorsement(승인) 기반 트랜잭션 전송 플로우**
  1. 트랜잭션 생성 요청 (체인코드 실행 요청과 동일)
  2. 트랜잭션 실행 결과 반환 (아직 DB에는 반영 안 됨)
  3. 오더러에게 트랜잭션 실행 결과와 함께 블록 생성 요청
  4. 오더러는 블록을 생성. 피어들에게 전달
  5. 피어들은 블록에 오류가 없는지 검증 후 블록체인 및 State DB에 적용



5. 데이터베이스

- **노드들의 데이터베이스**
  - 모든 노드들은 자신만의 블록체인 저장소와 State DB를 갖고 있음

  - 블록체인 저장소
    - 바이트 형태로 그냥 파일에 직접 씀
    - 개발자가 직접 접근할 일은 거의 없음

  - State DB
    - Key-Value DB
      - LevelDB(기본, 권장)
      - CouchDB(복잡한 쿼리 지원, 느리고 보안 취약)
    - 체인코드가 CRUD 하는 대상은 바로 이 DB
      - LevelDB에 익숙하면 좋음

6. 구동 환경

- **Docker container**
  - Orderer, Peer, Fabric CA 모두 Docker Image로 배포됨
  - Binary도 있기 때문에 그냥 실행도 가능
- **Unix environment**
  - Linux 또는 macOS 환경이 필요
  - Windows도 가능하지만, 제공되는 기본 스크립트들이 모두 Unix shell 환경을 기준으로 작성됨
- **프로그래밍 언어**
  - Go: 하이퍼레저 패브릭, 체인코드, Fabric SDK
  - JavaScript(node.js): 체인코드, Fabric SDK
  - Java: 체인코드, Fabric SDK

6. 구동 환경

- Docker container => docker, docker-compose
  - Orderer, Peer, Fabric CA 모두 Docker Image로 배포됨
  - Binary도 있기 때문에 그냥 실행도 가능
- Unix environment => WSL 2
  - Linux 또는 macOS 환경이 필요
  - Windows도 가능하지만, 제공되는 기본 스크립트들이 모두 Unix shell 환경을 기준으로 작성됨
- 프로그래밍 언어
  - Go: 하이퍼레저 패브릭, 체인코드, Fabric SDK
  - JavaScript(node.js): 체인코드, Fabric SDK
  - Java: 체인코드, Fabric SDK