

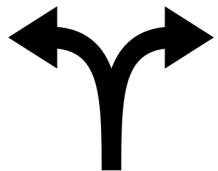


**TEZOS  
BLOCKCHAIN CAMP  
BUSAN**

# 테조스 플랫폼



**Formal verification**  
오류가 최소화된 스마트 컨트랙트

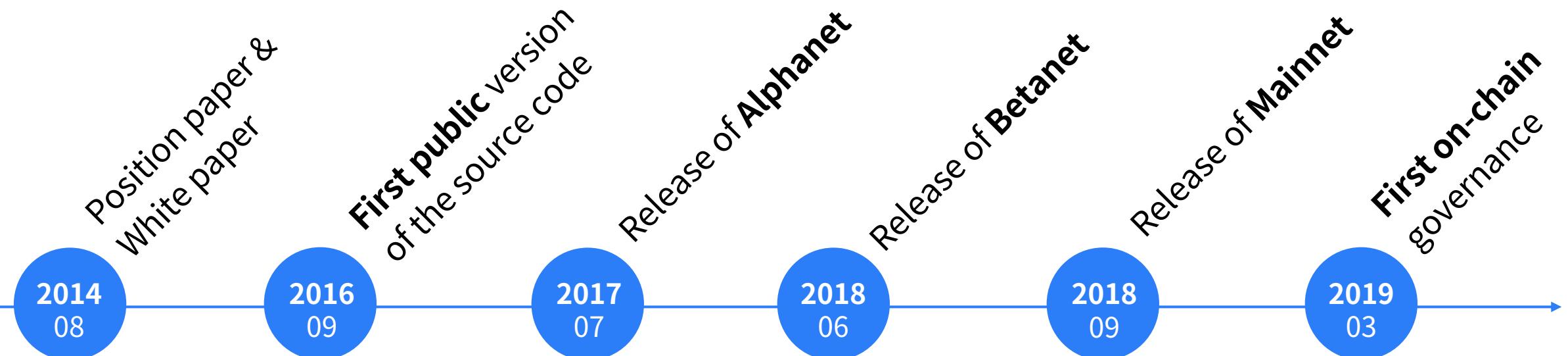


**On-chain Governance**  
하드 포크의 위험성이 최소화

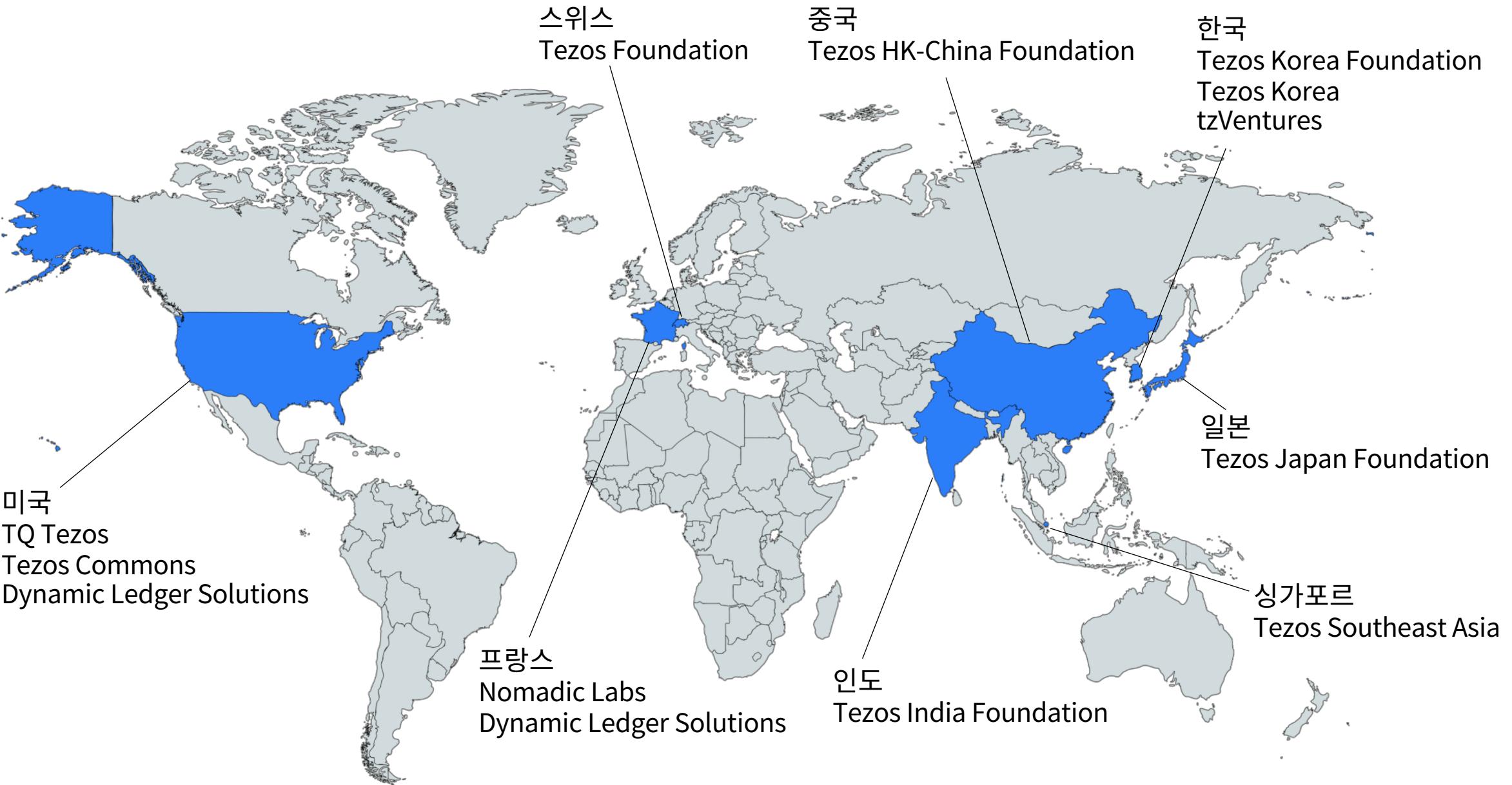


**Business Logic Automation**  
비지니스 스마트 컨트랙트 플랫폼

# 테zos 네트워크 연혁



# 테조스 글로벌 생태계



# 테조스 메인 네트워크\*

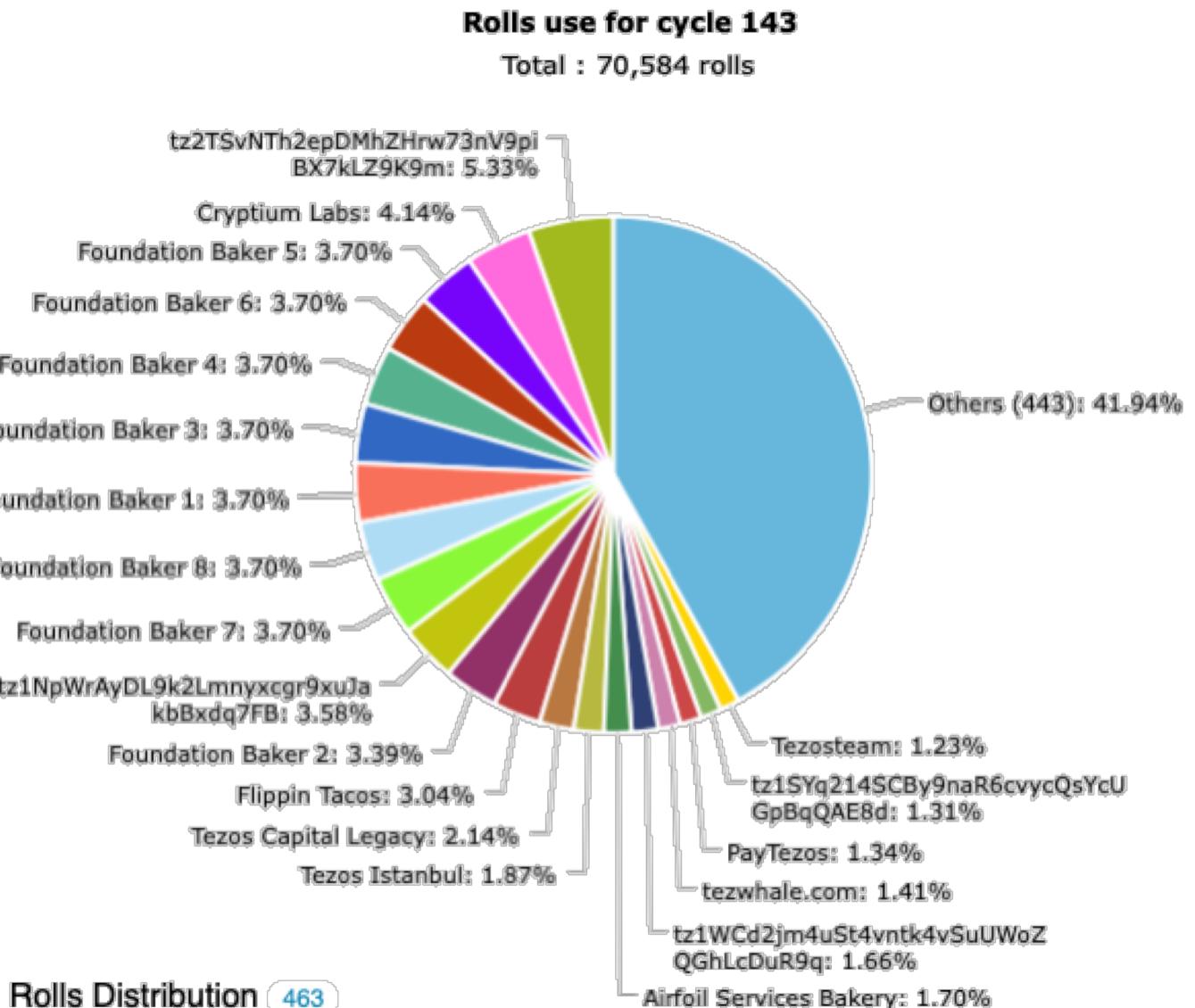
**463 Bakers**

**8,546 Nodes**

**565,519 Blocks**

**804,110,000 티저 in total**

**564,672,000 티저 in staking**



\* As of 19.08.15

# 한국 내 테조스 조직 소개

## 테조스 한국 재단 (Tezos Korea Foundation)

한국 내 테조스 생태계 조성

교육 프로그램, 학계 지원

밋업, 컨퍼런스 기획 및 주최

## (주)테조스코리아 (Tezos Korea Inc.)

테조스 한국 재단 산하 영리 기업

블록체인 컨설팅 및 기술 지원

## **tzVentures**

글로벌 테조스 인큐베이터

스타트업 발굴, 지원, 데모 데이

# Advisors & MOUs

**김기천 교수, 박진하 교수** 건국대학교 정보통신대학원 블록체인전공

- 정보통신 산업 발전 및 블록체인전공 교육과정 개발 협약

**이광근 교수** 서울대학교 컴퓨터공학, 소프트웨어 무결점 연구소

- Tezos 스마트 컨트랙트 Formal Verification 및 Audit

**이재원 교수** KAIST/세종대 경영학과

- 블록체인 비즈니스 모델 및 인큐베이팅

**정호진 교수** 홍익대학교 경영학과

- 토큰 이코노미 및 Boundary / Allocation

**박선주 교수** 연세대학교 경영학과, 디지털 사회 연구센터

- 블록체인 인재양성 교육

# (주)테조스코리아

## 서비스

- 스마트 컨트랙트 작성 및 검증(Formal verification)
- 비즈니스 모델 및 토큰 이코노미 설계
- 거래소 상장 및 스테이킹 기술 지원
- 블록체인 특강(기업, 공공기관, 학교)
- 테조스 장외(OTC) 거래
- STO 플랫폼

## 파트너



# 테조스코리아 교육 프로그램



## 멀티캠퍼스 정규 과정 개설

10월 말 개설

블록체인 이론 및 실습

Tezos 스마트 컨트랙트

## 2019년 2학기 건국대학교

정보통신대학원 블록체인전공

석사과정 정규 강의 (2학점)

합의 알고리즘 개론 및 응용



## 캠퍼스 CEO 과정

서울산업진흥원 주관

19년 2학기, 20년 1학기

블록체인 기술 및 창업 과정 (3학점)

테zos코리아 교육 프로그램

참고 자료

<https://bit.ly/2KXcrKm>

커뮤니티

<https://tezoskoreacommunity.org>

공모전

<https://tezoskorea.co.kr/innovators>

# 목차

## TEZOS BLOCKCHAIN CAMP BUSAN

**CHAPTER1 블록체인 기본**

**CHAPTER2 합의 알고리즘과 거버넌스**

**CHAPTER3 스마트 컨트랙트 이론**

**CHAPTER4 스마트 컨트랙트 실습**

# **CHAPTER1 블록체인 기본**

# **블록체인이란 무엇인가?**

Blockchain is an **open distributed ledger**

# Open Distributed Ledger

**Open** 누구에게나 공개되어 있고

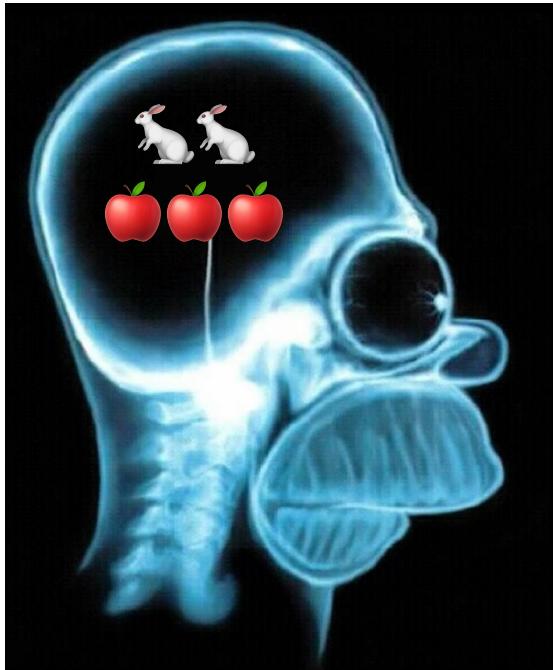
**Distributed** 분산되어 있는

**Ledger** (거래의 기록을 담는) 장부

= Open Shared Database

# 기록 방식의 발전은 효율성을 위한 노력의 결과

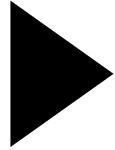
생산력의 발전  
기존 방식의 한계



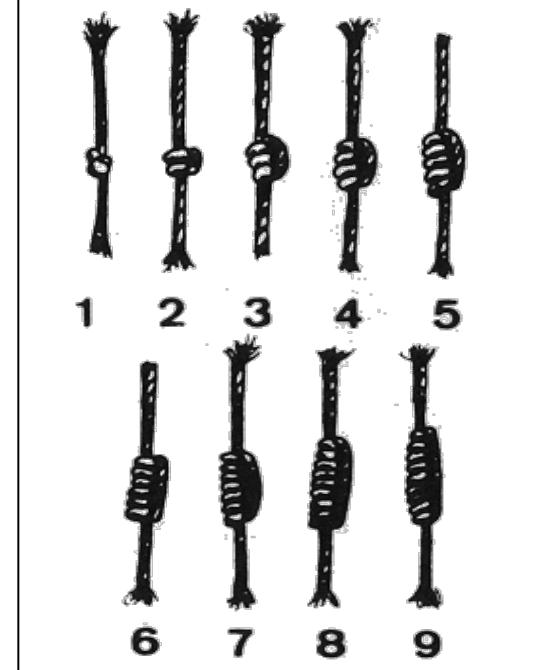
Memory



Cuneiform



새로운 기록 방식



Knots

# 기록 방식의 발전은 효율성을 위한 노력의 결과

기업의 등장  
소유/경영 분리

기록 관리인

Spring		
Negroes bought in 1848		
April 11 <sup>th</sup>	Mary	591
do 11 <sup>th</sup>	Emily	250
do 11 <sup>th</sup>	Maria & child John	630
do 11 <sup>th</sup>	Robert	100
do 10 <sup>th</sup>	Rachel & her child	575
do 10 <sup>th</sup>	James	000
do 11 <sup>th</sup>	Amanda	375
do 11 <sup>th</sup>	Charlott	375
do 11 <sup>th</sup>	Margret	400
Amt. brought over		28 294 00
Expensis on the trip		28 4,900 00
<hr/>		11 63 00
<hr/>		29 6,53 00
Cost & expensis		
neat. money		
Harriett money recovered by law		
Great money on the trip		

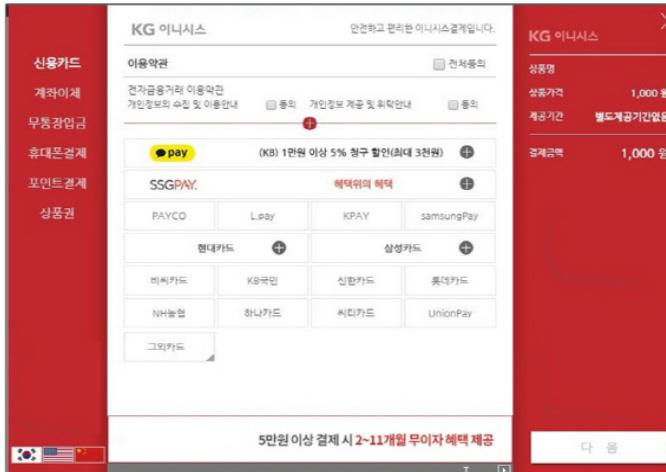
Spring		
Negroes sold in 1848		
June 25 <sup>th</sup>	Mary	591
May 1 <sup>st</sup>	Emily	300
May 1 <sup>st</sup>	Maria & child John	810
May 1 <sup>st</sup>	Robert	100
May 1 <sup>st</sup>	Rachel & her child	775
May 1 <sup>st</sup>	James	000
May 25 <sup>th</sup>	Amanda	415
May 17 <sup>th</sup>	Charlott	460
May 25 <sup>th</sup>	Margret	450
Amt. brought over		29 675 00
Cost & expensis		3 2,06 00
<hr/>		29 6,53 00
neat. money		
Harriett money recovered by law		
Great money on the trip		

Single entry bookkeeping

Folio 12			Dr.		Merchandise		Cr.	
190-			190-		190-		190-	
Jan 1	To Sundries	1	1631 00	Jan 1	By Cash	1	2960	
.	2	W. Young	3972 8	.	3	Jones & Co.	14480	
.	3	Jones & Co.	7240	.	3	A. Daniels	10160	
.	5	H. Henderson	2400 00	.	5	Bowers & Co.	14864	
June 20	Sundries	2	11945 0	.	5	H. Henderson	16460	
				.	5	Cash	1320	
				May 1	Bills Recd	3	62488	
				June 30	Balance		230816	
							353545	
				July 1	To Balanced	230816		353545

Double entry bookkeeping

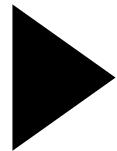
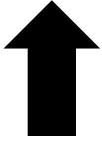
# 거래: 현대적, 효율적 삶을 위한 수단



# 거래의 기본은 신뢰

신뢰 비용의 증가

빈도  
규모  
지역



신탁

Middle man

TTP

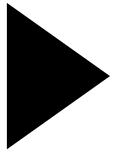
Central authority



# 중앙화된 시스템은 많은 문제를 해결

금융시스템

신뢰가 없는 사람들  
사이에 신뢰를 만들어  
거래가 가능하게 함



믿을 수 있고  
효율적이고  
편리하고  
문제가 최소화

거래 비용 감소

국가와 사회가 안정되고 제도와 규제가 정비됨에 따라 거래의 많은 문제가 해결되었다.

오늘날 우리는 인터넷을 통해 지구 반대편의 상품을 편하게 구매할 수 있다.

간편 결제, 간편 송금 서비스 등의 등장으로 거래 방식이 더욱 편리해졌다.

가끔 해킹, 피싱 등으로 피해가 발생하지만 전체 거래 규모에 비하면 매우 사소한 부분이며, 보안은 점점 발전 중이다.

...

# **블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화**

# 비트코인의 등장

# 비트코인, P2P 전자화폐 시스템의 출현 (ref #1)

## Bitcoin: A Peer-to-Peer Electronic Cash System

2008년 10월 31일

Cryptography Mailing List

2009년 1월 SW 배포 및 네트워크 가동

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.

# 거인의 어깨 위에 서 있는 비트코인

**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.

거래가 전자 서명으로 보호되고, 타임스탬프와 함께 체인 형태로 기록된다.

체인은 작업의 결과인 해시 값으로 연결되어 있다.

대다수의 컴퓨팅 파워가 정직하게 사용된다면, 기록이 안전하게 유지된다.

누구나 노드로 참여할 수 있고, 언제든 그만둘 수 있다.

# 비트코인 탄생의 배경

1982

## Byzantine Generals Problem

임의의 장애를 견딜 수 있는 (Byzantine fault tolerant) 분산 시스템(Synchronous)을 구현하기 위한 해결책 제시

1985

## FLP Impossibility

비동기 분산 시스템에서, 하나의 프로세서라도 crash될 경우 safety와 liveness를 동시에 만족하는 알고리즘은 존재하지 않는다.

1989

## Paxos

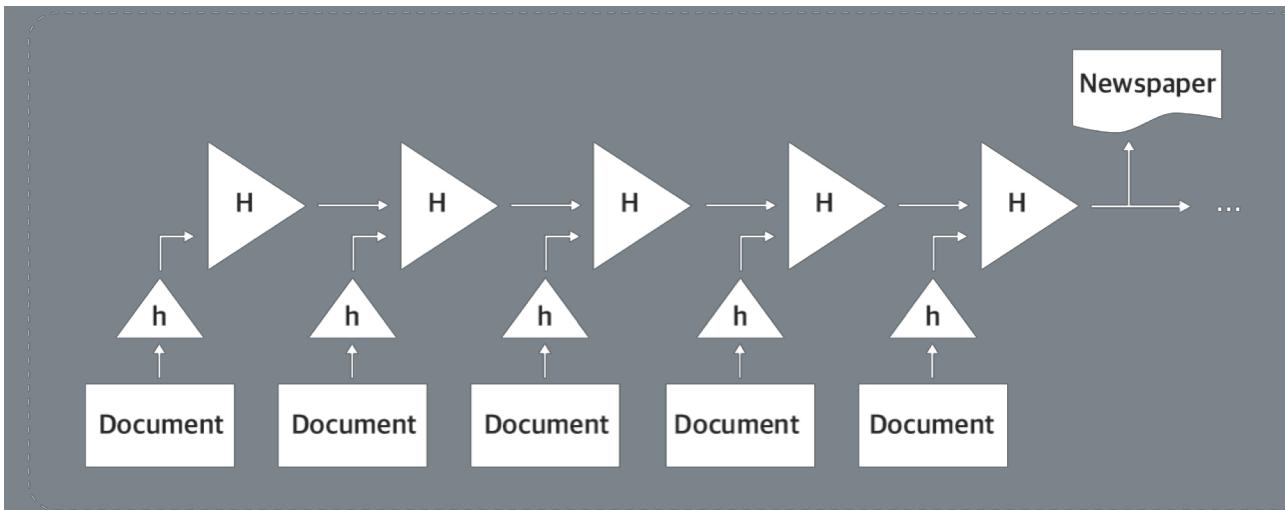
비동기(Asynchronous) 분산 시스템에서 Safety를 보장하는 합의 알고리즘 제시

1991

## A chain of cryptographically secured blocks

디지털 정보를 조작이 불가능하도록 안전하고(hash pointer) 효율적으로(grouped into batches) 저장하는 방법 제시.

1992년에 Merkel Trees(1979) 추가



# 비트코인 탄생의 배경

- 1982 **Byzantine Generals Problem**  
임의의 장애를 견딜 수 있는 (Byzantine fault tolerant) 분산 시스템(Synchronous)을 구현하기 위한 해결책 제시
- 1985 **FLP Impossibility**  
비동기 분산 시스템에서, 하나의 프로세서라도 crash될 경우 safety와 liveness를 동시에 만족하는 알고리즘은 존재하지 않는다.
- 1989 **Paxos**  
비동기(Asynchronous) 분산 시스템에서 Safety를 보장하는 합의 알고리즘 제시
- 1991 **A chain of cryptographically secured blocks**  
디지털 정보를 조작이 불가능하도록 안전하고(hash pointer) 효율적으로(grouped into batches) 저장하는 방법 제시.  
1992년에 Merkel Trees(1979) 추가
- 1992 **A concept of proof of work**  
디지털 서명을 이용해 서비스 요청자의 자원을 소모하도록 요구. 네트워크 자원 남용(Spam, Ddos)에 대한 해결책 제시
- 1997 **Hashcash**  
효율적인 해시 함수 기반의 작업 증명 시스템. 여러 암호 화폐 채굴(PoW) 알고리즘의 일부
- 1999 **Proof of Work**  
'작업 증명' 용어 등장 및 개념 정리(formalization)
- 2004 **A CHAIN OF BLOCKS 특허 만료**

# Papers

**Byzantine Generals Problem** | Byzantine Generals Problem (L. Lamport)

**FLP Impossibility** | Impossibility of Distributed Consensus with One Faulty Process (M.J. Fischer)

**Paxos** | The Part-Time Parliament (L. Lamport)

**A chain of cryptographically secured blocks** | How to time-stamp a digital document (S. Haber)

**A concept of proof of work** | Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology (C. Dwork)

**Hashcash** | A partial hash collision based postage scheme (A. Beck)

**Proof of Work** | Proof of Work and Bread Pudding Protocols (M. Jakobsson)

# Digital Cash

1983 **Ecash (David Chaum)**

Anonymous cryptographic electronic money (Blind Signatures for untraceable payments)

1997 **Hashcash (Adam Beck)**

Proof of work to aid the generation and distribution of new coins

1998 **B-Money (Wei Dai)**

Anonymous, distributed electronic cash system

1998 **Bit Gold (Nick Szabo)**

Proof of work system. Aimed to avoid centralized authorities

2004 **RPOW Token (Hal Finney)**

Reusable PoW + Hashcash

# Cryptography

1985 **ECDSA**

Elliptic Curve Digital Signature Algorithm. 256-bit ECC = 3072-bit RSA

1996 **RIPEME-160**

RIPE Message Digest of 160 bits used to make bitcoin addresses

2002 **SHA256**

Secure Hash Algorithm of 256 bits used to hash a block header and find nonce

**블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화**

**비트코인의 등장**

**거인의 어깨, Byzantine generals, Hash, Digital signature, Chain**

**WHY**

# 비트코인 개발 동기에 대한 추측: 금융 위기



# 비트코인 개발 동기에 대한 추측: 금융 위기

## RAW HEX VERSION

```
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 3B A3 ED FD 7A 7B 12 B2 7A C7 2C 3E  
67 76 8F 61 7F C8 1B C3 88 8A 51 32 3A 9F B8 AA  
4B 1E 5E 4A 29 AB 5F 49 FF FF 00 1D 1D AC 2B 7C  
01 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 FF FF FF FF 4D 04 FF FF 00 1D  
01 04 45 54 68 65 20 54 69 6D 65 73 20 30 33 2F  
4A 61 6E 2F 32 30 30 39 20 43 68 61 6E 63 65 6C  
6C 6F 72 20 6F 6E 20 62 72 69 6E 6B 20 6F 66 20  
73 65 63 6F 6E 64 20 62 61 69 6C 6F 75 74 20 66  
6F 72 20 62 61 6E 6B 73 FF FF FF FF 01 00 F2 05  
2A 01 00 00 00 43 41 04 67 8A FD B0 FE 55 48 27  
19 67 F1 A6 71 30 B7 10 5C D6 A8 28 E0 39 09 A6  
79 62 E0 EA 1F 61 DE B6 49 F6 BC 3F 4C EF 38 C4  
F3 55 04 E5 1E C1 12 DE 5C 38 4D F7 BA 0B 8D 57  
8A 4C 70 2B 6B F1 1D 5F AC 00 00 00 00
```

.....  
.....  
....;fiýz{.^zG,>  
gv.a.È.À^SQ2:Ù,a  
K.^J)\*\_Iyy...~+|  
.....  
.....  
.....ÿÿÿM.ÿy..  
..The Times 03/  
Jan/2009 Chancellor on brink of  
second bailout f  
or banksÿÿÿ...ò.  
\*....CA.gÙyºþUH'  
.gñ;q0..\Ö"(à9.|  
ybæ.eþIÖk?L18Ä  
ðU.ä.À.þ\8M+ø..W  
Šlp+kñ.\_~....



## 비트코인 제네시스(Genesis) 블록

“두 번째 구제 금융을 앞두고 있는 재무 장관”

# 비트코인 개발 동기에 대한 추측: Cyberpunk

2008년 10월 31일 **Cryptography Mailing List**에 백서 공개

“전통적인 화폐의 **근본적인 문제**는 그것을 운용하기 위해 필요한 **신뢰 그 자체**이다.”



The Foundation for Peer to Peer Alternatives

Main My Page Members Videos Forum Groups Blogs Chat

All Discussions My Discussions + Add



## Bitcoin open source implementation of P2P currency

Posted by Satoshi Nakamoto on February 11, 2009 at 22:27

[View Discussions](#)

I've developed a new open source P2P e-cash system called Bitcoin. It's completely decentralized, with no central server or trusted parties, because everything is based on crypto proof instead of trust. Give it a try, or take a look at the screenshots and design paper:

Download Bitcoin v0.1 at <http://www.bitcoin.org>

The root problem with conventional currency is all the trust that's required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust. Banks must be trusted to hold our money and transfer it electronically, but they lend it out in waves of credit bubbles with barely a fraction in reserve. We have to trust them with our privacy, trust them not to let identity thieves drain our accounts. Their massive overhead costs make micropayments impossible.

# Cyberpunk: 기술을 통해 자치, 자유를 추구

## 정보 독점 사회

1980~90 급격한 기술 발전

정부 권력의 강화 (규제)

다국적 거대 기업의 등장



## 사이버 펑크

VS

개인 사생활 보호

자치를 위한 투쟁





# Bitcoin open source implementation of P2P currency

Posted by Satoshi Nakamoto on February 11, 2009 at 22:27

(ref #2)

[View Discussions](#)

I've developed a new open source P2P e-cash system called Bitcoin. It's completely decentralized, with no central server or trusted parties, because everything is based on crypto proof instead of trust. Give it a try, or take a look at the screenshots and design paper: Download Bitcoin v0.1 at <http://www.bitcoin.org>

The root problem with conventional currency is all the trust that's required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust. Banks must be trusted to hold our money and transfer it electronically, but they lend it out in waves of credit bubbles with barely a fraction in reserve. We have to trust them with our privacy, trust them not to let identity thieves drain our accounts. Their massive overhead costs make micropayments impossible.

A generation ago, multi-user time-sharing computer systems had a similar problem. Before strong encryption, users had to rely on password protection to secure their files, placing trust in the system administrator to keep their information private. Privacy could always be overridden by the admin based on his judgment call weighing the principle of privacy against other concerns, or at the behest of his superiors. Then strong encryption became available to the masses, and trust was no longer required. Data could be secured in a way that was physically impossible for others to access, no matter for what reason, no matter how good the excuse, no matter what.

It's time we had the same thing for money. With e-currency based on cryptographic proof, without the need to trust a third party middleman, money can be secure and transactions effortless.

One of the fundamental building blocks for such a system is digital signatures. A digital coin contains the public key of its owner. To transfer it, the owner signs the coin together with the public key of the next owner. Anyone can check the signatures to verify the chain of ownership. It works well to secure ownership, but leaves one big problem unsolved: double-spending. Any owner could try to re-spend an already spent coin by signing it again to another owner. The usual solution is for a trusted company with a central database to check for double-spending, but that just gets back to the trust model. In its central position, the company can override the users, and the fees needed to support the company make micropayments impractical.

Bitcoin's solution is to use a peer-to-peer network to check for double-spending. In a nutshell, the network works like a distributed timestamp server, stamping the first transaction to spend a coin. It takes advantage of the nature of information being easy to spread but hard to stifle. For details on how it works, see the design paper at <http://www.bitcoin.org/bitcoin.pdf>

The result is a distributed system with no single point of failure. Users hold the crypto keys to their own money and transact directly with each other, with the help of the P2P network to check for double-spending.

Satoshi Nakamoto <http://www.bitcoin.org>



## Bitcoin open source implementation of P2P currency

Posted by Satoshi Nakamoto on February 11, 2009 at 22:27

[View Discussions](#)

I've developed a new open source P2P e-cash system called Bitcoin. It's completely decentralized, with no central server or trusted parties, because everything is based on crypto proof instead of trust. Give it a try, or take a look at the screenshots and design paper: Download Bitcoin v0.1 at <http://www.bitcoin.org>

The root problem with conventional currency is all the trust that's required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust. Banks must be trusted to hold our money and transfer it electronically, but they lend it out in waves of credit bubbles with barely a fraction in reserve. We have to trust them with our privacy, trust them not to let identity thieves drain our accounts. Their massive overhead costs make micropayments impossible.

A generation ago, multi-user time-sharing computer systems had a similar problem. Before strong encryption, users had to rely on password protection to secure their files, placing trust in the system administrator to keep their information private. Privacy could always be overridden by the admin based on his judgment call weighing the principle of privacy against other concerns, or at the behest of his superiors. Then strong encryption became available to the masses, and trust was no longer required. Data could be secured in a way that was physically impossible for others to access, no matter for what reason, no matter how good the excuse, no matter what.

It's time we had the same thing for money. With e-currency based on cryptographic proof, without the need to trust a third party middleman, money can be secure and transactions effortless.



## Bitcoin open source implementation of P2P currency

Posted by Satoshi Nakamoto on February 11, 2009 at 22:27

View Discussions

One of the fundamental building blocks for such a system is digital signatures. A digital coin contains the public key of its owner. To transfer it, the owner signs the coin together with the public key of the next owner. Anyone can check the signatures to verify the chain of ownership. It works well to secure ownership, but leaves one big problem unsolved: double-spending. Any owner could try to re-spend an already spent coin by signing it again to another owner. The usual solution is for a trusted company with a central database to check for double-spending, but that just gets back to the trust model. In its central position, the company can override the users, and the fees needed to support the company make micropayments impractical.

Bitcoin's solution is to use a peer-to-peer network to check for double-spending. In a nutshell, the network works like a distributed timestamp server, stamping the first transaction to spend a coin. It takes advantage of the nature of information being easy to spread but hard to stifle. For details on how it works, see the design paper at <http://www.bitcoin.org/bitcoin.pdf>

The result is a distributed system with no single point of failure. Users hold the crypto keys to their own money and transact directly with each other, with the help of the P2P network to check for double-spending.

Satoshi Nakamoto <http://www.bitcoin.org>

## 블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화

비트코인의 등장	거인의 어깨, Byzantine generals, Hash, Digital signature, Chain
개발 동기	제네시스 블록(금융위기), Cyberpunk, 중앙화, 신뢰, 프라이버시, 암호학

# 중앙화의 문제

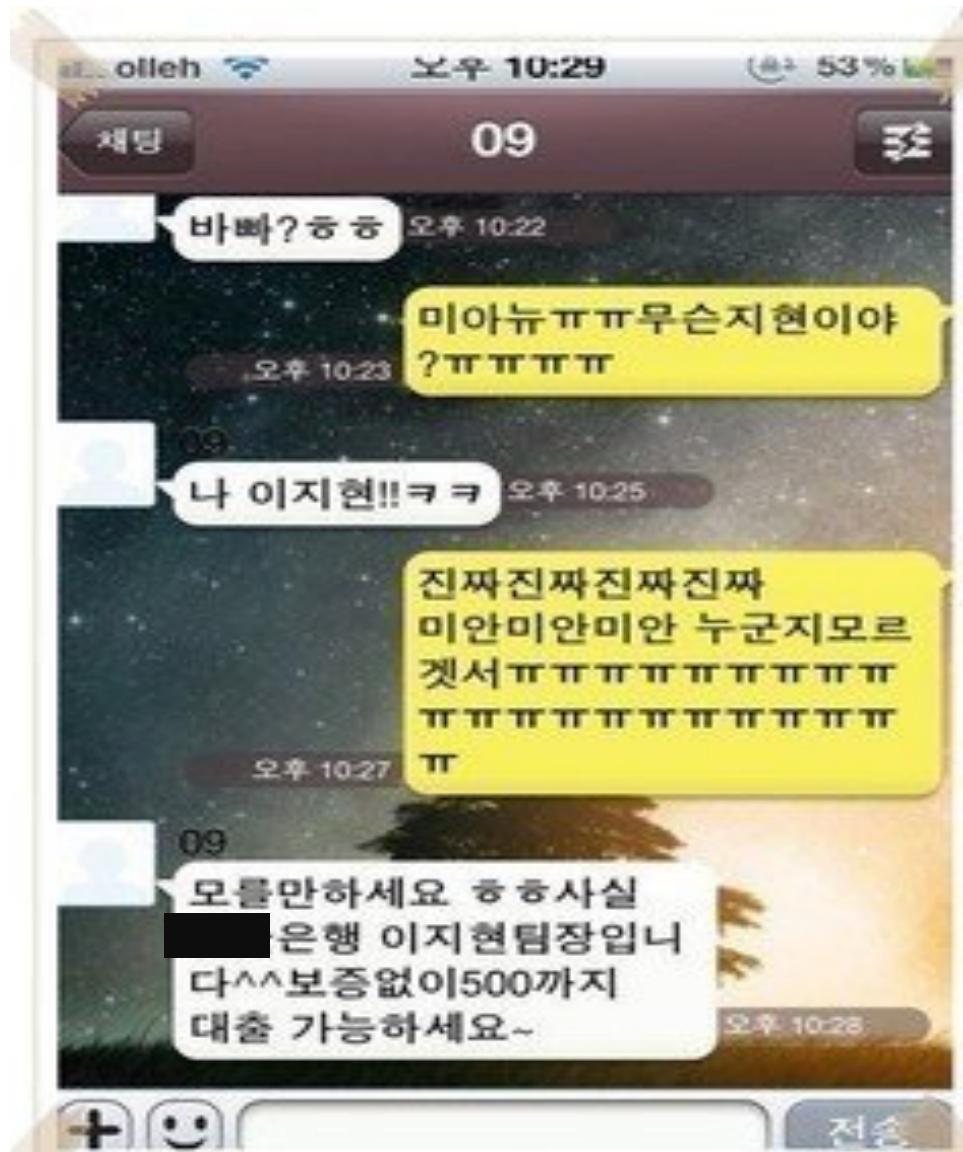
# 중앙화의 문제: Trust



# 중앙화의 문제: Trust

김미영 팀장입니다.

고객님께서는 최저 이율로 최고  
3,000만원까지 30분 이내  
통장 입금 가능합니다.



## 중앙화의 문제: Single point of failure

화재로 인한 문화재 손실



## 중앙화의 문제: Single point of failure

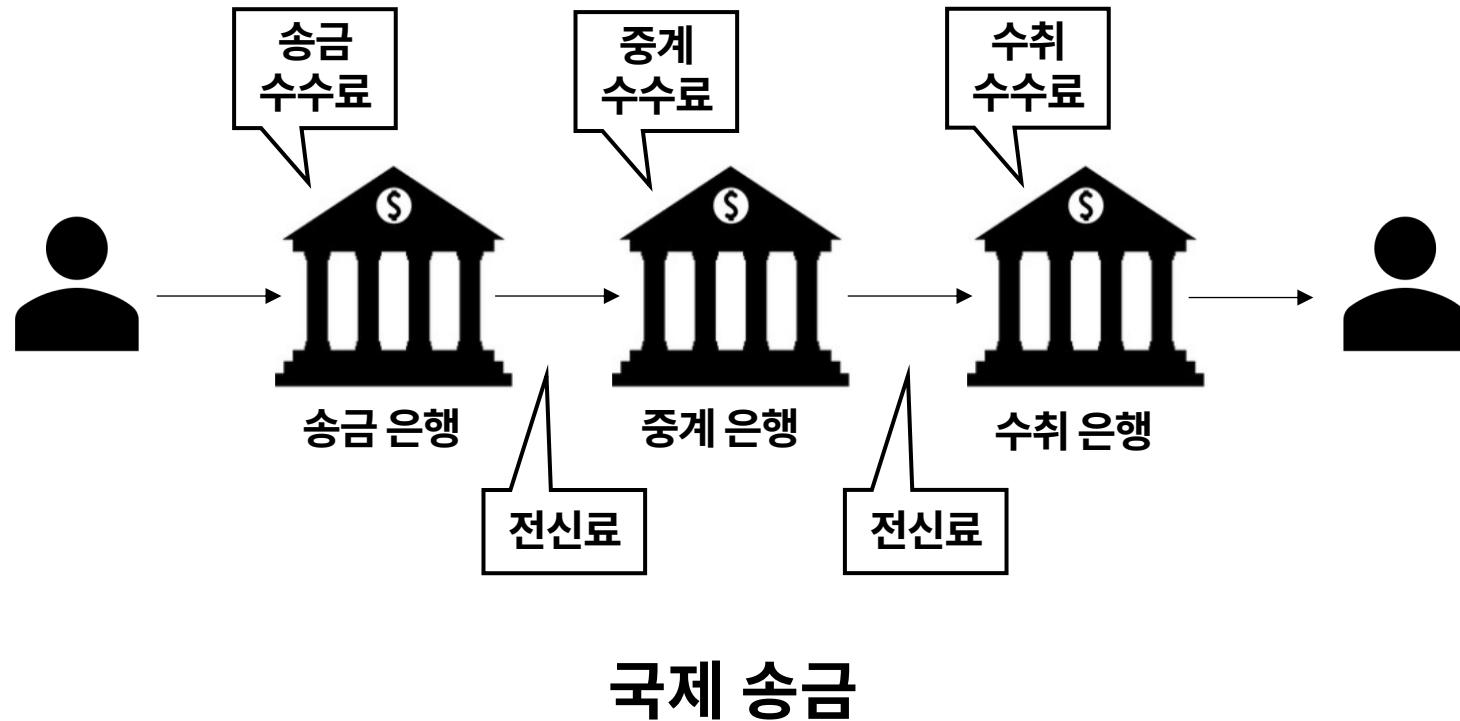
### 먹물퐁당 감자치즈



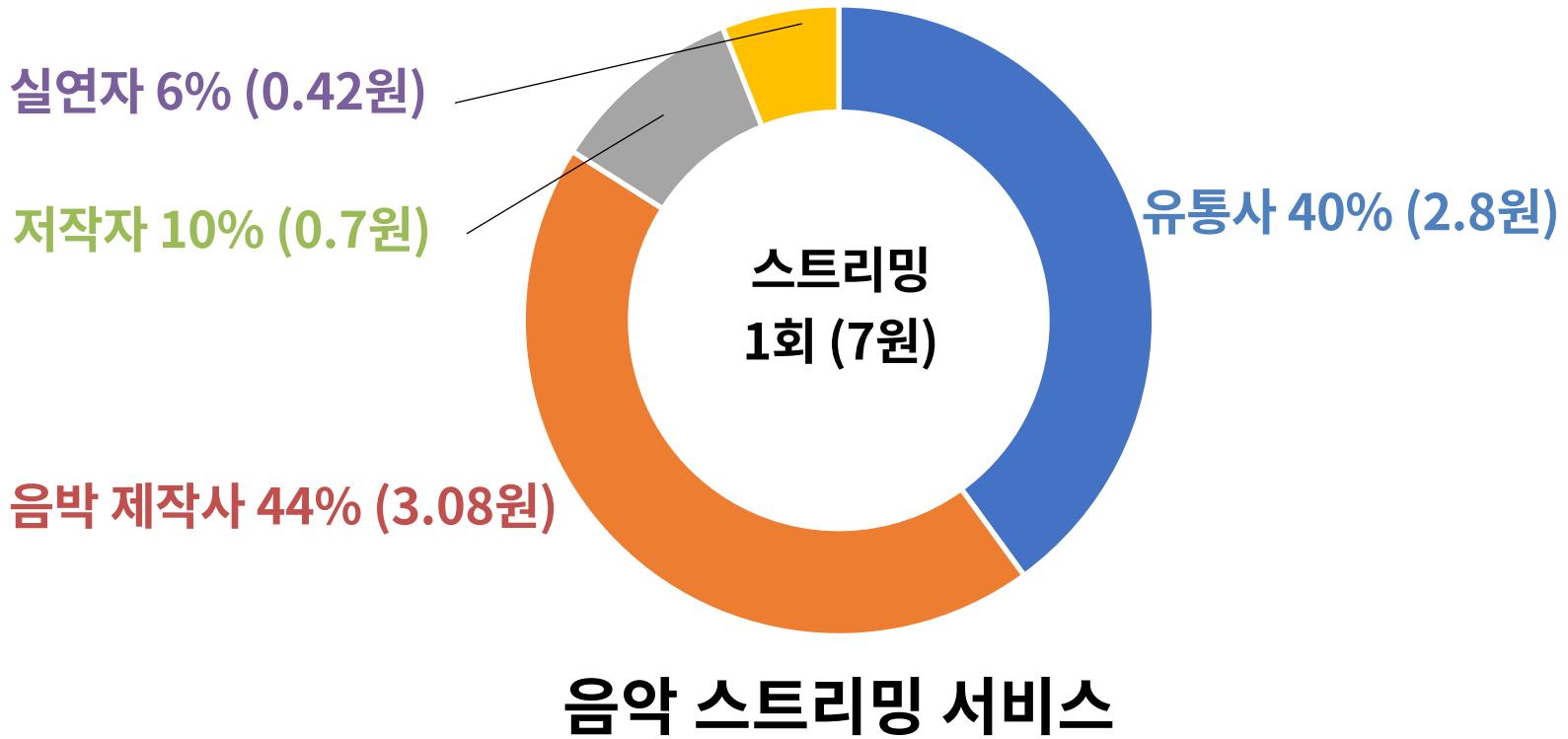
# 중앙화의 문제: Single point of failure

- 1995 **Citibank**  
러시아 해커 Vladimir Levin이 시티뱅크 전산망을 해킹하여 천만달러를 전세계로 송금함.
- 2000 **Mafia Boy**  
캐나다 10대 소년 Michel Calce에 의해 yahoo, Dell, CNN 등 다국적 기업들의 서비스가 중단(DoS)됨
- 2004 **Delta Airline**  
Sven Jaschan(당시 18세)가 델타 항공사의 운항 노선의 일부를 취소 시킴
- 2016 **Bangladesh Bank Heist**  
방글라데시아 은행이 사용하는 SWIFT 시스템(국제 은행 간 송금 표준)이 해킹당해 8,100만 달러 손실
- 2017 **WannaCry**  
'비트코인을 요구하는 랜섬 웨어. 텔레포니카, 영국 국민 건강 서비스, 페덱스 등 150개 이상 국가의 대기업, 기관이 피해를 입음
- 2018 **Facebook**  
페이스북 서버 해킹으로 5천만명의 개인정보가 유출

# 중앙화의 문제: Tyranny



## 중앙화의 문제: Tyranny



# 중앙화의 문제

## Trust

Efficiency, Authority, License

## Single point of failure

Ownership of data

## Tyranny

Network effect, Platform power

## Overhead cost

Availability, Security, Regulation

## 블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화

비트코인의 등장	거인의 어깨, Byzantine generals, Hash, Digital signature, Chain
개발 동기	제네시스 블록(금융위기), Cyberpunk, 중앙화, 신뢰, 프라이버시, 암호학
중앙화의 문제	Trust, Single point of failure, Tyranny, Overhead cost

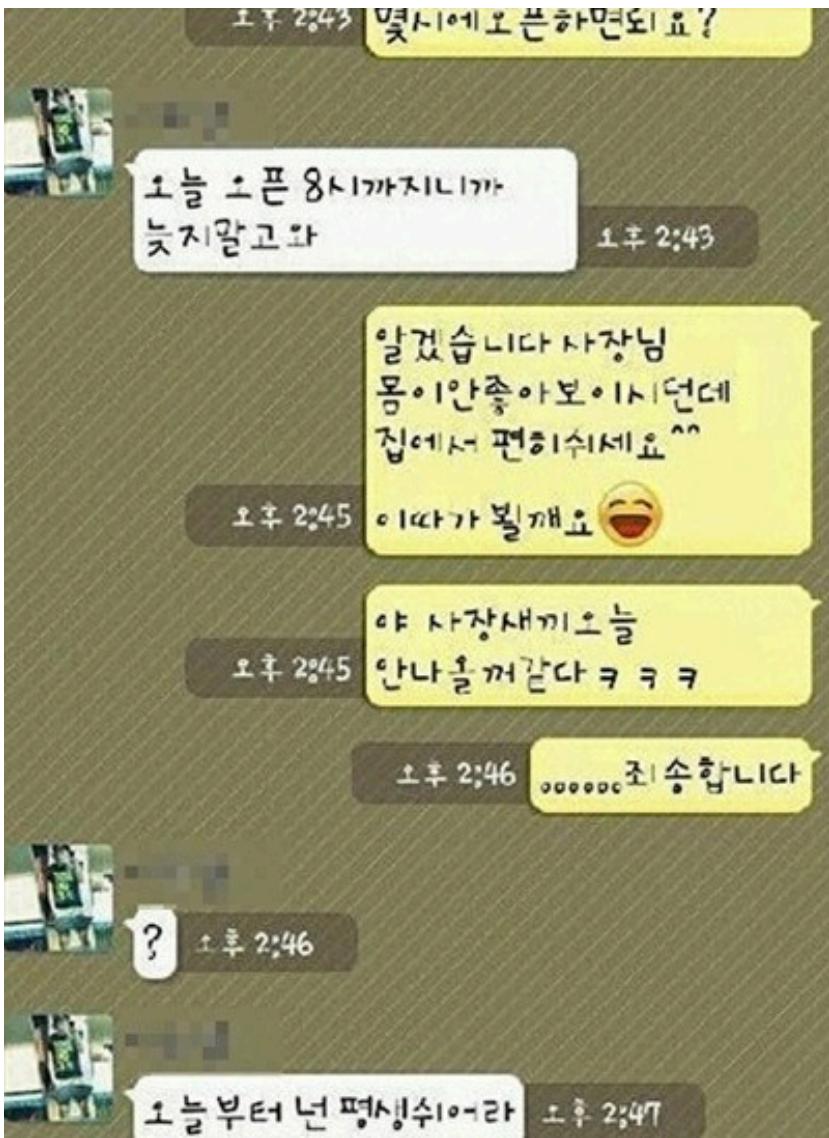
# 중개자 없는 P2P 거래

## 중개자 없는 P2P 거래

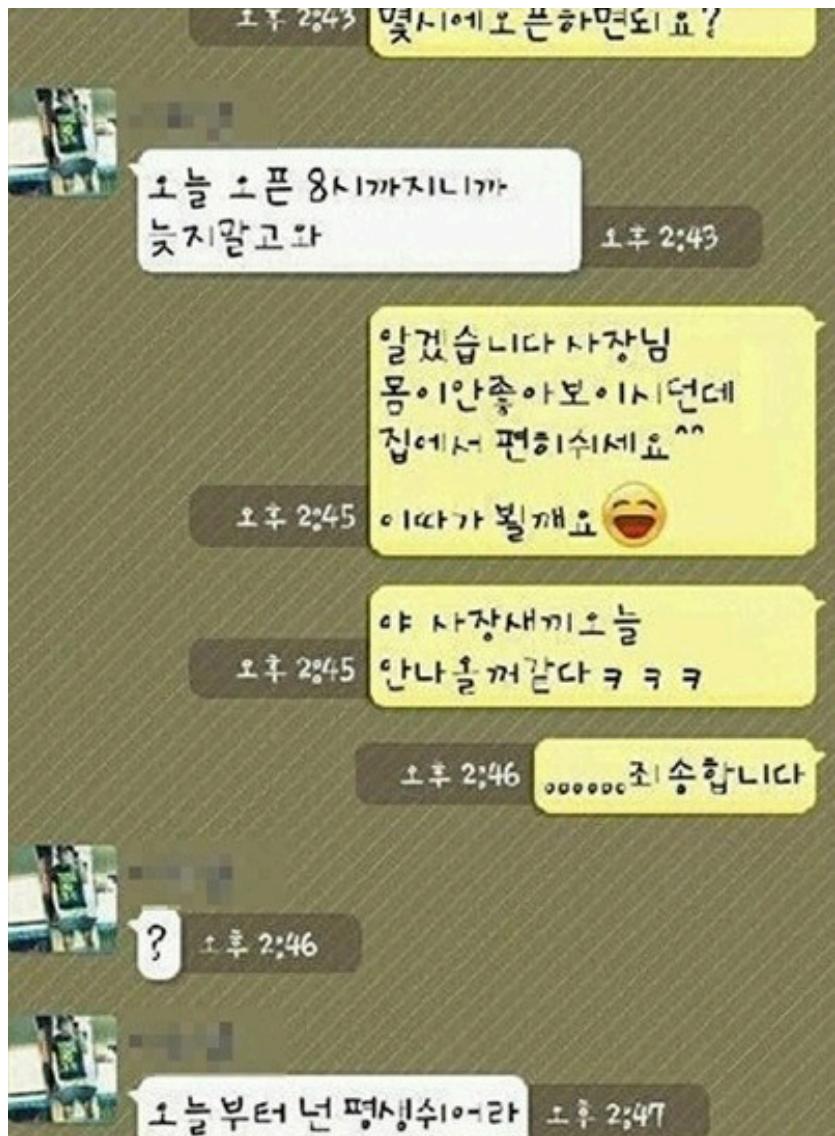
I've developed a new open source **P2P** e-cash system called Bitcoin. It's completely **decentralized, with no central server or trusted parties.**

Users hold the crypto keys to their own money and **transact directly with each other,** with the help of the P2P network to check for double-spending.

# Server-client vs. Peer-to-Peer



# Server-client



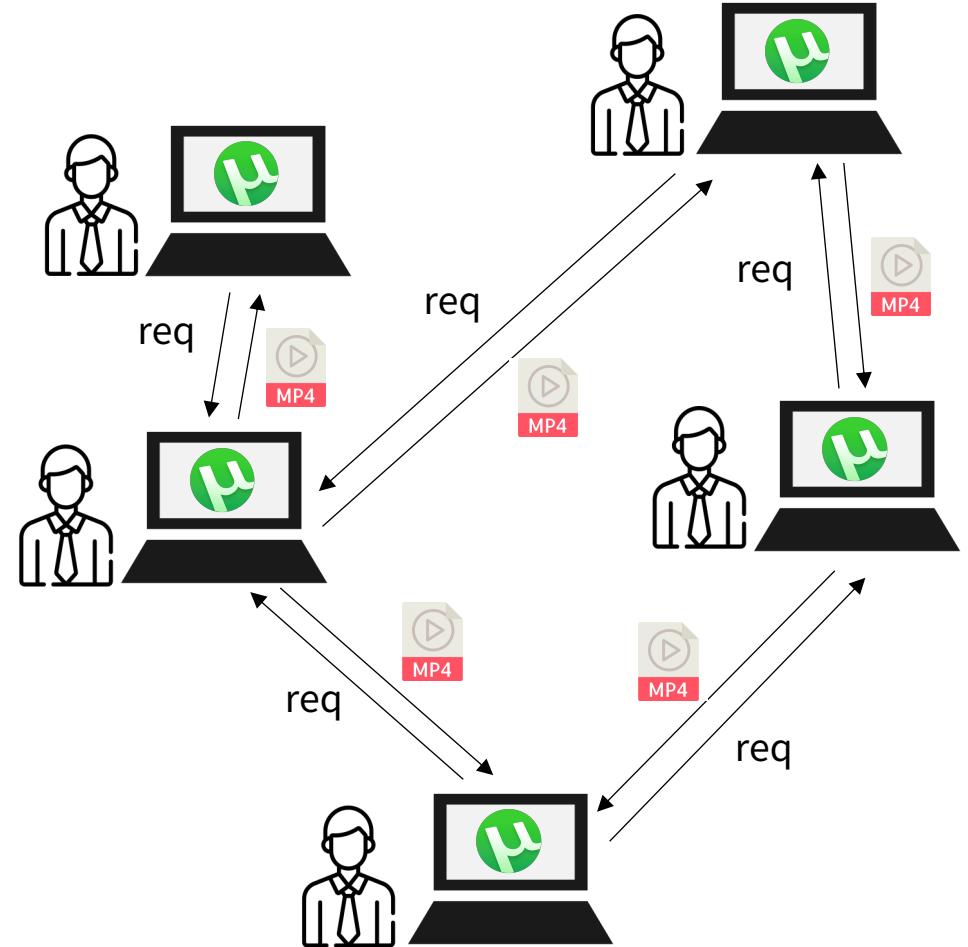
# Peer-to-Peer (ref #3)

Screenshot of a BitTorent client interface showing multiple download tasks:

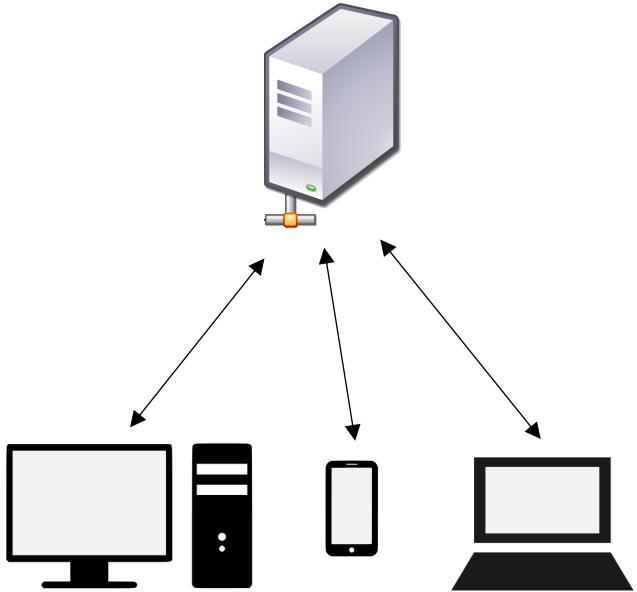
#	Name	Status	Done	Size	Seeds	Peers	Down Speed	Up Speed	ETA
1	* archlinux-2019.04.01-x86_64.iso	[F] Seed...	100%	604,0 MiB	0 (358)	0 (107)	0 B/s	0 B/s	∞
2	* minix R3.3.0-588a35b.iso.bz2	[F] Comple...	100%	287,6 MiB	0 (48)	0 (24)	0 B/s	0 B/s	∞
3	5 ↓ ubuntu-18.10-desktop-amd64.iso	Downlo...	81,3%	1,86 GiB	79 (10...)	0 (235)	711,8 KiB/s	0 B/s	29m
4	12 ↓ Solus-3-Budgie.iso	[F] Dow...	23,1%	1,14 GiB	39 (85)	0 (1)	528,5 KiB/s	0 B/s	29m
5	3 ↓ linuxmint-18-cinnamon-64bit.iso	[F] Dow...	22,9%	1,58 GiB	30 (55)	1 (11)	91,6 KiB/s	0 B/s	3h 37m
6	7 ↓ slackware64-14.2-iso	Downlo...	14,6%	2,58 GiB	75 (150)	0 (48)	368,4 KiB/s	0 B/s	1h 44m
7	4 ↓ kali-linux-2019-1a-amd64.iso	[F] Dow...	13,5%	3,24 GiB	85 (19...)	1 (132)	366,3 KiB/s	5,5 KiB/s	2h 5m
8	2 ↓ FreeBSD-12.0-STABLE-amd64-disc1.iso	Downlo...	13,2%	898,6 MiB	6 (9)	0 (1)	15,8 KiB/s	0 B/s	9h 6m
9	9 ↓ tails-amd64-3.13.1.img	[F] Dow...	11,8%	1,15 GiB	44 (133)	0 (84)	260,3 KiB/s	0 B/s	1h 1m
10	13 ↓ MX-18.2_x64.iso	[F] Dow...	10,6%	1,34 GiB	35 (126)	0 (69)	354,7 KiB/s	0 B/s	52m
11	16 ↓ enwiki-20190101-pages-articles-multistream.x...	[F] Dow...	3,5%	15,54 GiB	17 (68)	0 (3)	592,3 KiB/s	0 B/s	6h 49m
12	1 ↓ manjaro-xfce-18.0-stable-x86_64.iso	[F] Dow...	3,1%	1,86 GiB	2 (6)	1 (2)	0 B/s	0 B/s	∞
13	15 ↓ debian-9.8.0-amd64-DVD-1.iso	[F] Dow...	2,3%	3,37 GiB	32 (172)	0 (128)	127,7 KiB/s	0 B/s	6h 24m
14	6 ↓ openSUSE-Leap-42.3-DVD-x86_64.iso	[F] Dow...	2,0%	4,32 GiB	42 (67)	0 (53)	179,0 KiB/s	0 B/s	7h 3m
15	8 ↓ hannah_montana_linux_x86_basic_edition.iso	[F] Dow...	0,5%	691,5 MiB	1 (1)	0 (8)	6,4 KiB/s	0 B/s	11h 11m
16	14 ↓ elementaryos-5.0-stable.20181016.iso	[F] Dow...	0,0%	1,37 GiB	0 (0)	0 (0)	0 B/s	0 B/s	∞
17	11 ↓ haiku-r1beta1-x86_gcc2_hybrid-anyboot.zip	Paused	0,0%	932,8 MiB	0 (0)	0 (0)	0 B/s	0 B/s	∞
18	10 ↓ Gentoo-Linux-livedvd-amd64-multilib-20160704	Paused	0,0%	2,12 GiB	0 (6)	0 (1)	0 B/s	0 B/s	∞

Below the table, there is a summary section with the following information:

- Progress: [progress bar]
- Availability: [availability bar]
- Transfer:
  - Time Active: 30m
  - Downloaded: 618,2 MiB (617,5 MiB this session)
  - Download Speed: 662,2 KiB/s (351,3 KiB/s avg.)
  - Download Limit: ∞
  - Share Ratio: 0,00
  - ETA: 29m
  - Uploaded: 0 B (0 B this session)
  - Upload Speed: 0 B/s (0 B/s avg.)
  - Upload Limit: ∞
  - Reannounce In: 0
  - Connections: 80
  - Seeds: 7
  - Peers: 0
  - Wasted: 1
  - Last Seen Complete: 10m ago
  - DHT: 0 nodes

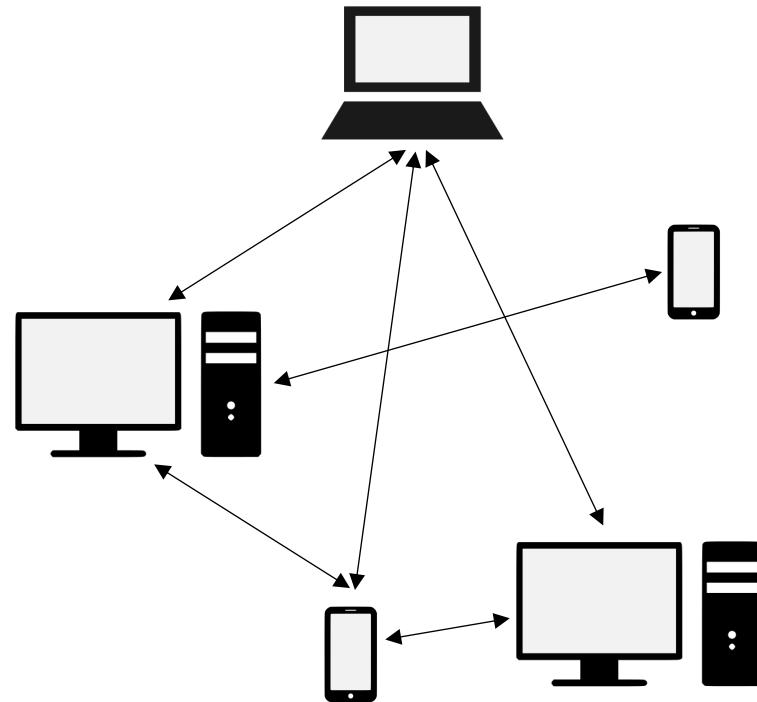


# Server-client vs. Peer-to-Peer



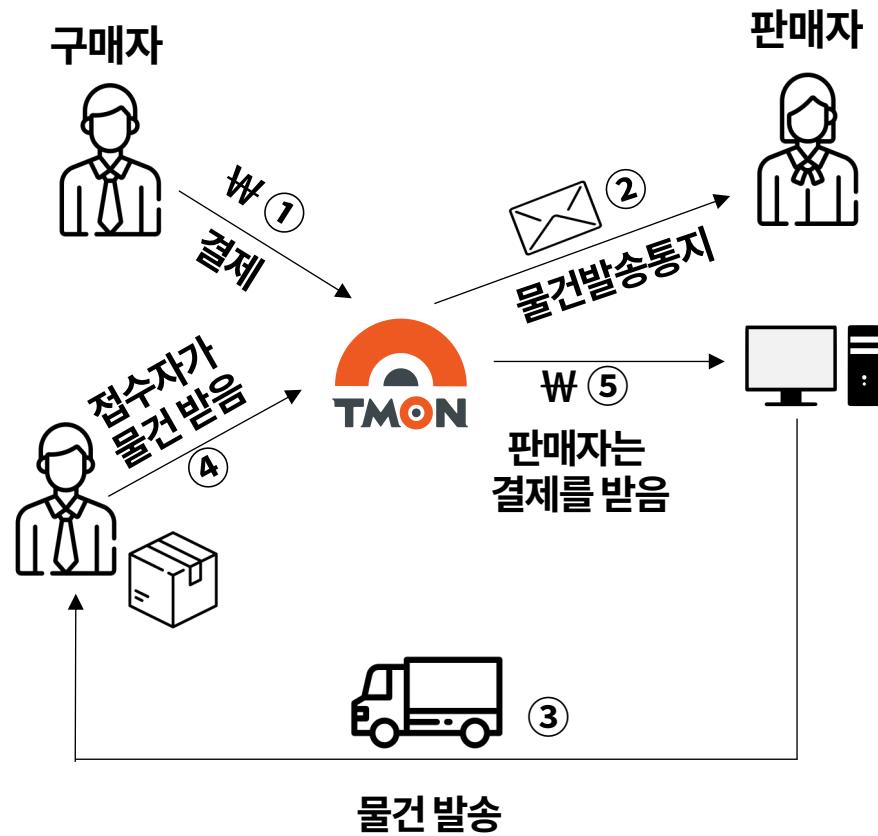
**Clients** request  
**A server** responses

**VS**

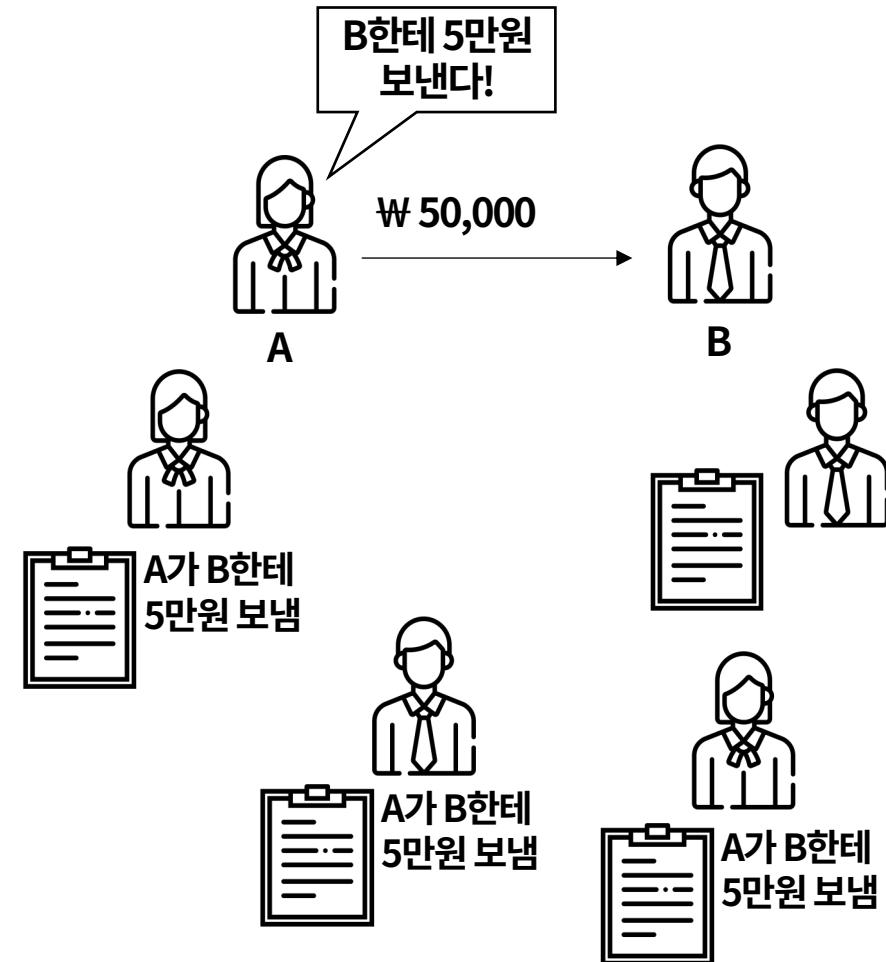


**Each peer** request as a client  
**Each peer** responses as a server

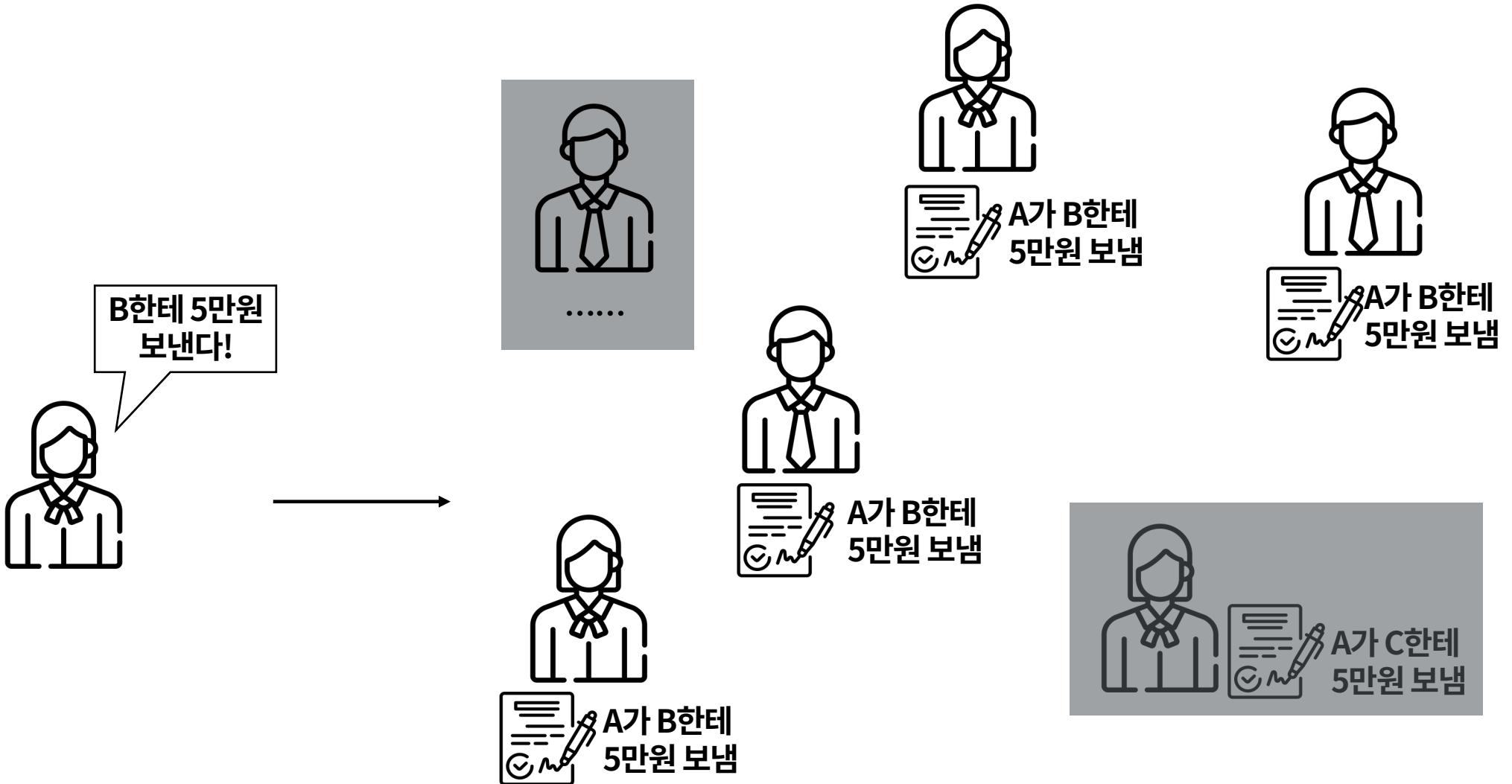
# Server-client vs. Peer-to-Peer



VS



## 분산화의 장점: 대다수가 정직한 경우



분산 장부의 목표: 동일한 장부의 유지

언제나 대다수의 노드가 동일한 장부에 합의  
합의를 위해서는 투표와 같은 다수결 규칙이 필요

# 합의 알고리즘

# 폐쇄 네트워크 vs. 공개 네트워크



소문과 평판



???

나카모토의 해결책

컴퓨팅 파워로 투표를 대체  
보상 인센티브

# Proof of Work

# Bitcoin: 1st generation of blockchain

사이버 평크 운동

2008년 금융위기

기술의 발전

컴퓨터 과학 + 암호학 + 경제학(게임이론)

블록체인을 활용한 애플리케이션

최초의 성공한 암호 화폐



# Ethereum: Beyond Bitcoin

## Vitalik Buterin

초기 비트코인 개발 기여

Bitcoin Magazine 공동 설립자 (11년 9월)

비트코인의 한계 극복 시도

13년 이더리움 백서 발표

14년 11월 월드 테크놀로지 어워드 IT/SW 수상

15년 7월 30일 이더리움 메인넷 런칭



# Ethereum: 2nd generation of blockchain

비트코인의 한계 : Scripts, DDoS(서비스 거부 공격)

## 스마트 컨트랙트

닉 자보에 의해 처음 제안 (1994)

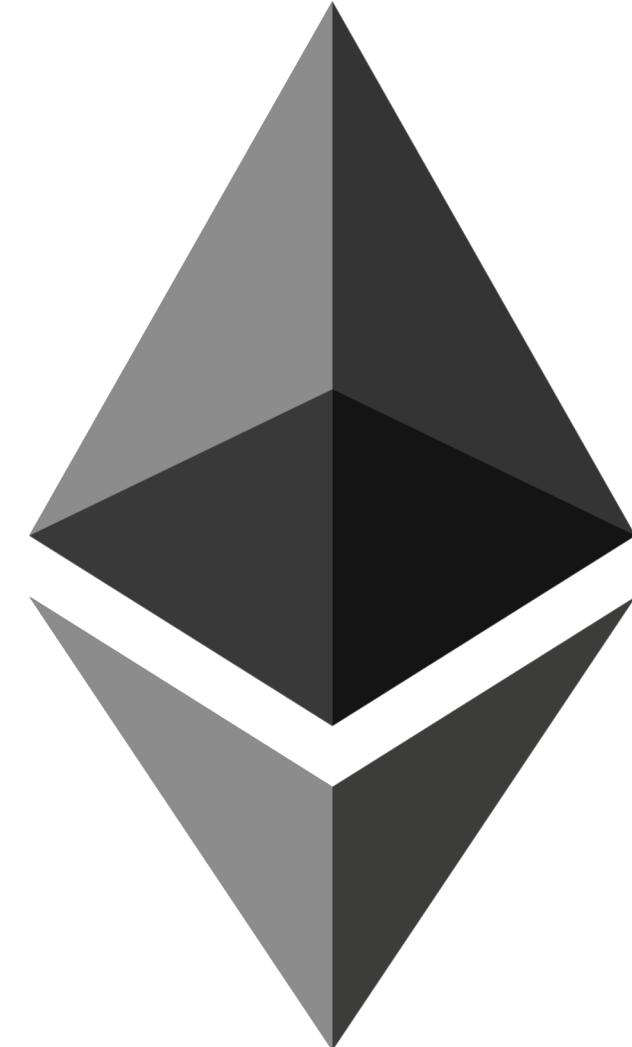
계약을 프로그래밍 언어로 작성

Not by court, but code

블록체인에 저장된 프로그램

가스 모델로 DDoS 방지

디지털 암호 화폐 -> 블록체인 플랫폼



## 블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화

비트코인의 등장	거인의 어깨, Byzantine generals, Hash, Digital signature, Chain
개발 동기	제네시스 블록(금융위기), Cyberpunk, 중앙화, 신뢰, 프라이버시, 암호학
중앙화의 문제	Trust, Single point of failure, Tyranny, Overhead cost
중개자 없는 P2P 거래	직접 거래, 대다수가 정직, 다수결, 합의 알고리즘, 컴퓨팅 파워, 보상 인센티브

**SO WHAT**

## Applications of blockchain

# Key features

Decentralization → Disintermediation

Immutability → Verifiability

Transparency → Auditability

Programmability → Automation

## Applications of blockchain

# Key advantages

**Replacement** of existing trust system

Reducing transaction cost by **standardization**

Cost cutting by **automation**

New business model of a form of **direct participation**

# Considerations

Not (**yet**) for high transaction frequency & strict finality

Not only for fin-tech → New **transaction** models

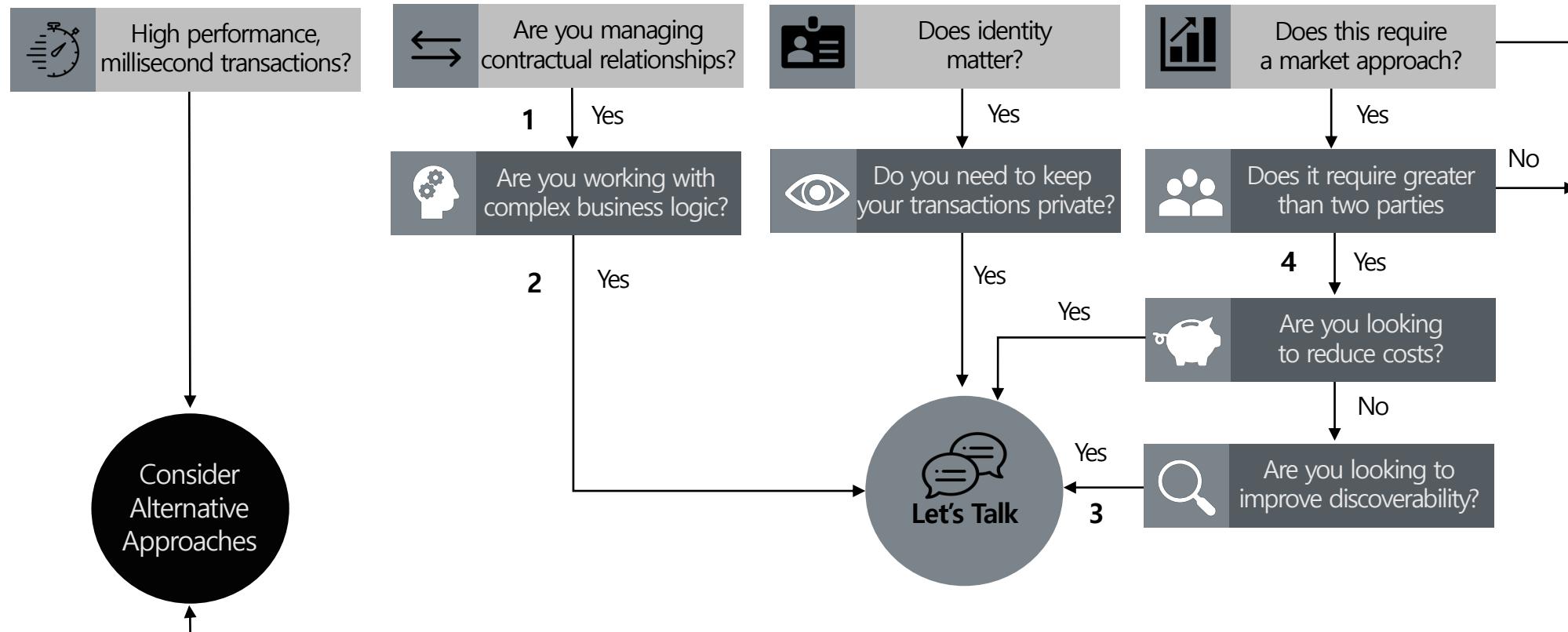
# DLT-based applications in financial sector (WBG, ref #4)

Money & Payments	<ul style="list-style-type: none"><li>- Digital currencies</li><li>- Payment authorization, clearance &amp; settlement</li><li>- International remittances and cross-border payments</li><li>- Foreign exchange</li><li>- Micropayments</li></ul>
Financial Services & Infrastructure	<ul style="list-style-type: none"><li>- Capital markets: digital issuance, trading &amp; settlements of securities</li><li>- Commodities trading</li><li>- Notarization services (e.g. for mortgages)</li><li>- Collateral / movable asset registries</li><li>- Syndicated loans</li><li>- Crowdfunding (as initial coin offerings)</li><li>- Insurance for automating payouts and validation of occurrence of insured event</li></ul>
Collateral registries & ownership registers	<ul style="list-style-type: none"><li>- Land registries, property titles &amp; other collateral registries</li></ul>
Internal systems of financial service provider	<ul style="list-style-type: none"><li>- Replacing internal ledgers maintained by large, multinational financial service providers that record information across different departments, subsidiaries, or geographies</li></ul>

# DLT-based applications in other sectors (WBG)

Identity	<ul style="list-style-type: none"><li>- Digital identity platforms</li><li>- Storing personal records: birth, marriage &amp; death certificates</li></ul>
Trade & Commerce	<ul style="list-style-type: none"><li>- Supply chain management (management of inventory and disputes)</li><li>- Product provenance &amp; authenticity (e.g. artworks, pharmaceuticals, diamonds)</li><li>- Trade finance and Post-trade processing</li><li>- Rewards &amp; loyalty programs</li><li>- Invoice management</li><li>- Intellectual property registration</li><li>- Internet of Things</li></ul>
Agriculture	<ul style="list-style-type: none"><li>- Financial services in the agricultural sector like insurance, crop finance and warehouse receipt s</li><li>- Provenance of cash crops</li><li>- Safety net programs related to delivery of seeds, fertilizers and other inputs</li></ul>
Governance	<ul style="list-style-type: none"><li>- E-voting systems and E-Residence</li><li>- Government record-keeping, e.g. criminal records</li><li>- Reducing fraud and error in government payments and tax fraud</li><li>- Protection of critical infrastructure against cyberattacks</li></ul>
Healthcare	<ul style="list-style-type: none"><li>- Electronic medical records</li></ul>
Humanitarian & Aid	<ul style="list-style-type: none"><li>- Tracking delivery &amp; distribution of food, vaccinations, medications, etc.</li><li>- Tracking distribution and expenditure of aid money</li></ul>

# How to decide whether to use it? (IBM, ref #5)



- 1 By design, no one party can modify, delete, or even append any record without consensus, making the system useful for ensuring the immutability of contracts and other legal documents.
- 2 Smart contracts aim to provide security superior to traditional contract and reduce other Tx costs associated with contracting.
- 3 When everyone on an exchange can view the same ledger, it is easy to broadcast an intention (or offer) by appending it. For example, in a trading network, all ask and bids would be visible to every network participant.
- 4 Blockchain networks allow each participant to create customized solutions using their own proprietary business logic while running on the same common ledger.

## 블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화

비트코인의 등장	거인의 어깨, Byzantine generals, Hash, Digital signature, Chain
개발 동기	제네시스 블록(금융위기), Cyberpunk, 중앙화, 신뢰, 프라이버시, 암호학
중앙화의 문제	Trust, Single point of failure, Tyranny, Overhead cost
중개자 없는 P2P 거래	직접 거래, 대다수가 정직, 다수결, 합의 알고리즘, 컴퓨팅 파워, 보상 인센티브
특징	Decentralization, Transparency, Programmability, Immutability

# **다양한 관점들**

Blockchain is a particular type of **Distributed Ledger Technology**

**Distributed Ledger Technology**

**The Internet of Value**

**Decentralized Autonomous Organization**

**Trustless**

**Bootstrapped**

**Protocol is fat**

**Anti-fragile**

Blockchain is a particular type of **Distributed Ledger Technology**

Blockchain is a particular type of **Distributed Ledger Technology**

Blockchain gave rise to **The Internet of Value**

Blockchain is a **Decentralized Autonomous Organization**

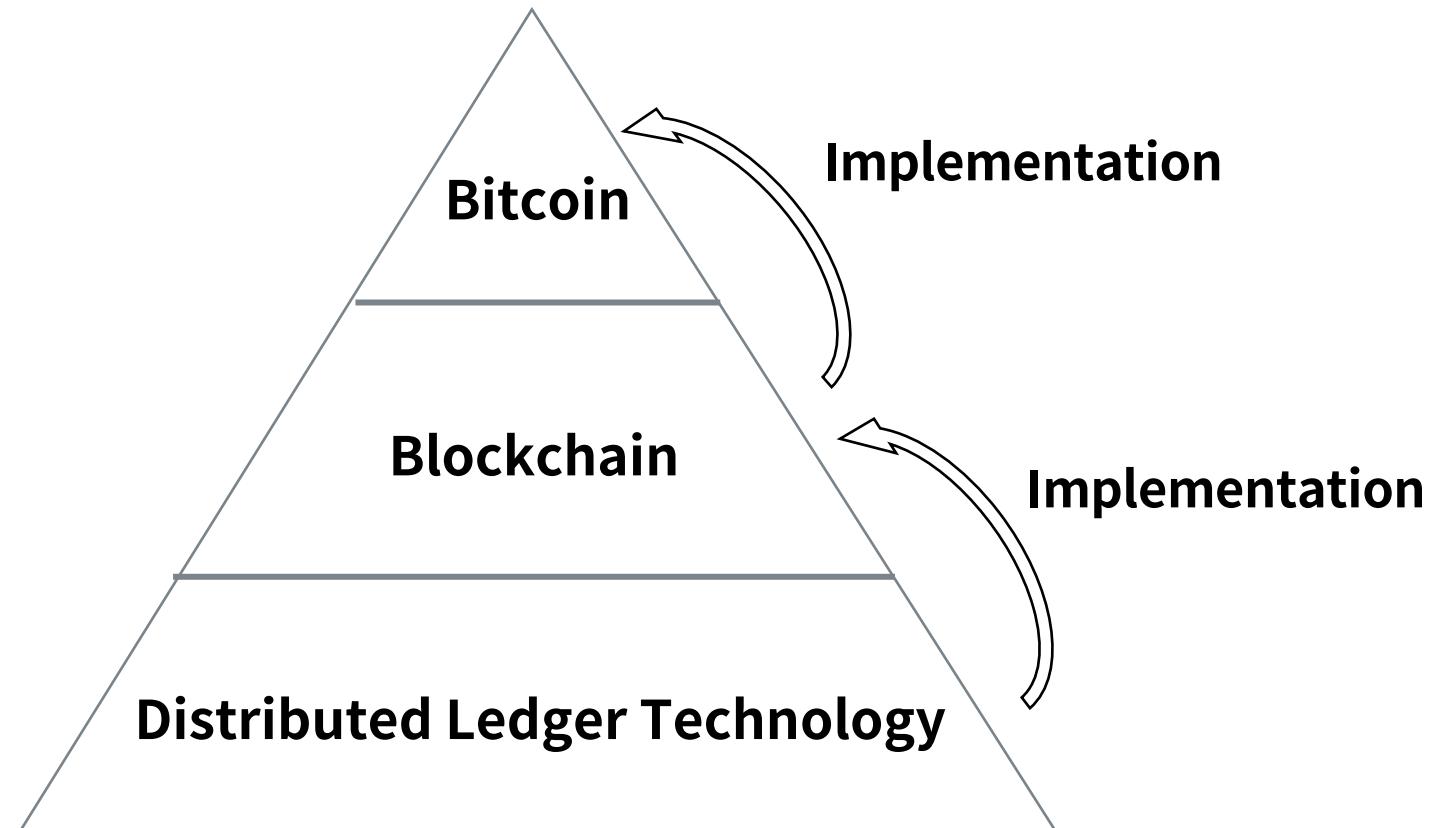
Blockchain is **Trustless**

Bitcoin is **bootstrapped**

Blockchain **protocol is fat**

Bitcoin is **anti-fragile**

Blockchain is a particular type of **Distributed Ledger Technology**



Blockchain is a particular type of **Distributed Ledger Technology**

Bitcoin ≠ Blockchain

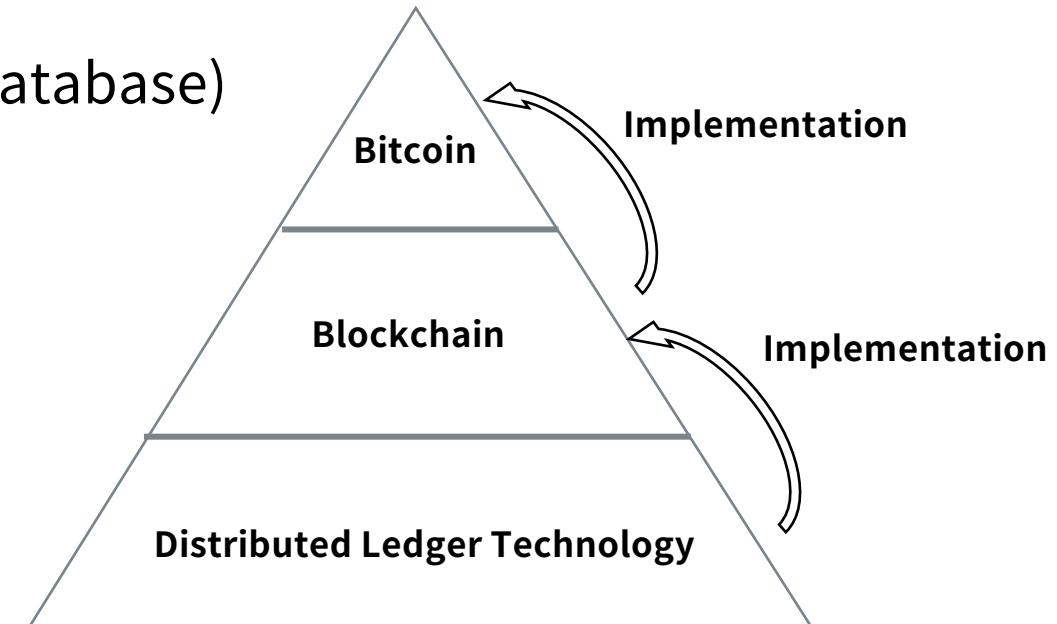
Bitcoin is a **cryptocurrency** implemented on top of blockchain

Blockchain is an **append-only data structure**

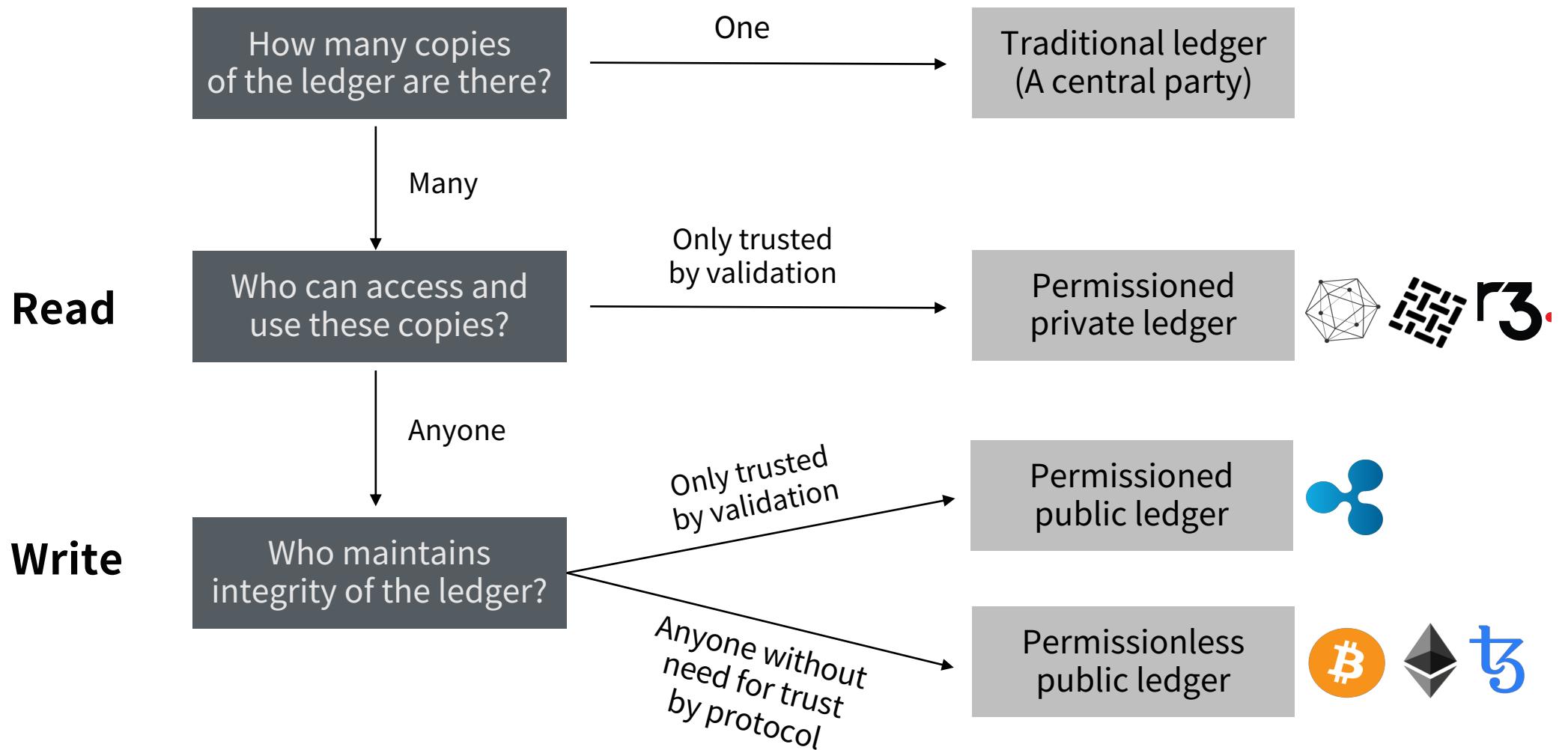
Some blockchains are open distributed ledgers (database)

Distributed ledger → Blockchain X

Blockchain → Distributed X



# Types of blockchain (WBG)

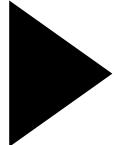


Blockchain gave rise to **The Internet of Value**

정보는 순식간에 전 세계로 이동 가능

가치의 이전과 교환은 여전히 신뢰 시스템에 의존

**The Internet  
of Information**



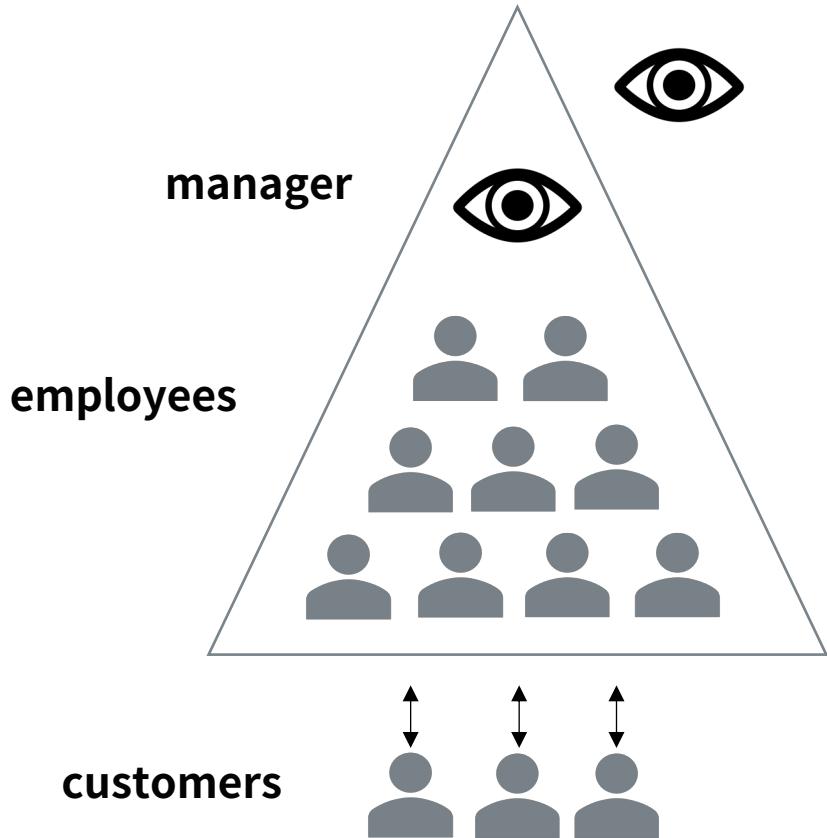
**The Internet  
of Value**

Bitcoin = Internet-like money

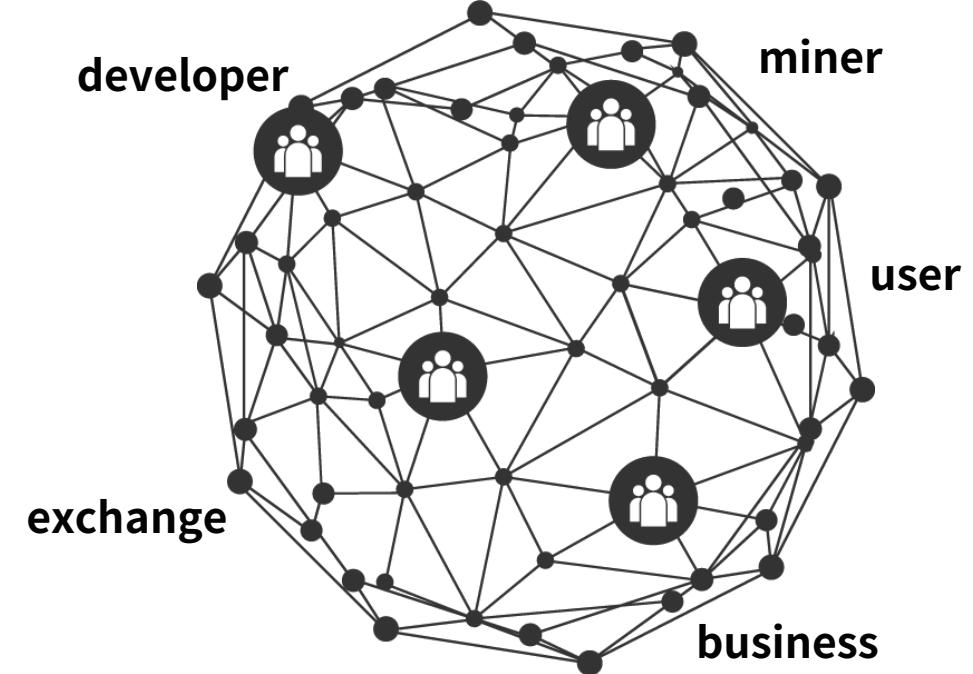
인터넷을 통해 모든 자산을 자유롭게 교환 가능

# Blockchain is a **Decentralized Autonomous Organization**

**Management by people  
Automation by machine**

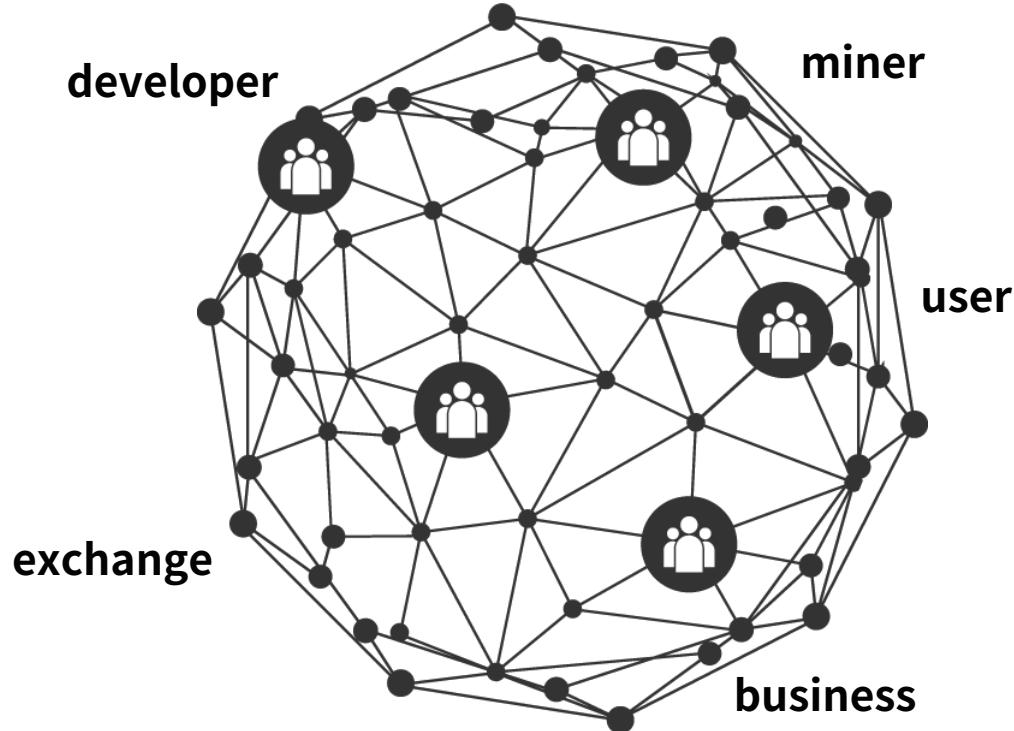


**Watching each other  
Governed by protocol**



# Blockchain is a **Decentralized Autonomous Organization**

**Bitcoin network as a world biggest auto payment service**



CEO / Managers:

Employees:

Salary:

Customers:

Stockholders:

Stocks:

Stock listing:

Capital gain:

# Blockchain is **Trustless**

## Trust

제삼자에 대한 신용을 전제  
리스크, 신용 증명 비용  
규제, 감독, 처벌  
제삼자에 의한 강제 집행  
측정 가능한 신뢰의 범위  
= 사회 발전의 범위

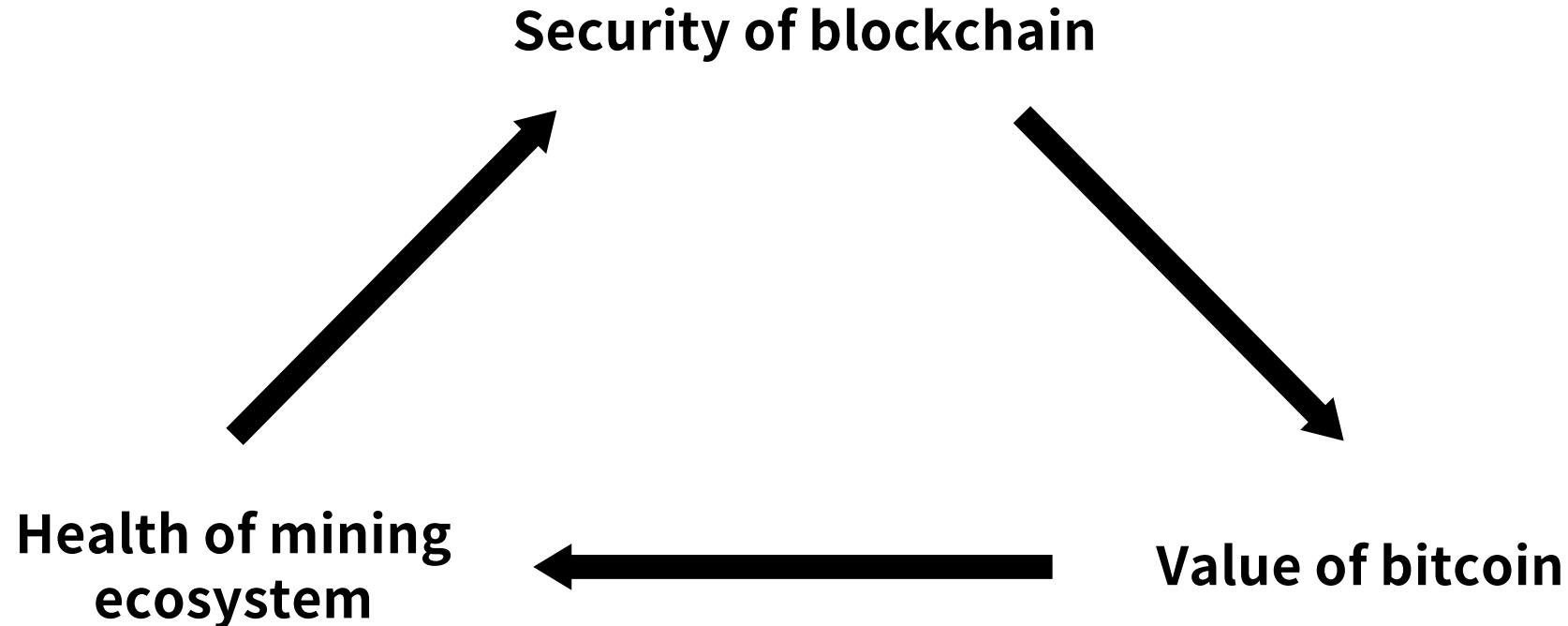
VS

## Trustless

신용이 필요 없음  
신용 자체가 존재 X  
부정행위 자체가 불가능  
프로토콜에 의한 자동 집행  
시스템의 분산  
자율, 자치

# Bitcoin is **bootstrapped** : from **NOTHING** to **ALL**

## Interlocking interdependencies in Bitcoin



Blockchain **protocol** is fat

# Protocol

통신이 가능한 네트워크 장비 사이에서 데이터, 메시지를 주고 받는 규칙  
신호 체계, 인증, 오류 감지 및 수정  
표준으로서 역할 (HTTP, TCP/IP, SMTP 등)

Blockchain **protocol** is fat

# Blockchain protocol

네트워크에 참여한 **노드**들이 **연결**되는 **방식**

기록(트랜잭션, 블록)에 대한 노드들의 **검증과 합의**를 통해 블록 추가

예. 블록 사이즈, 트랜잭션 조건, 이중 지불 금지

Blockchain **protocol is fat**

# Fat protocol\*

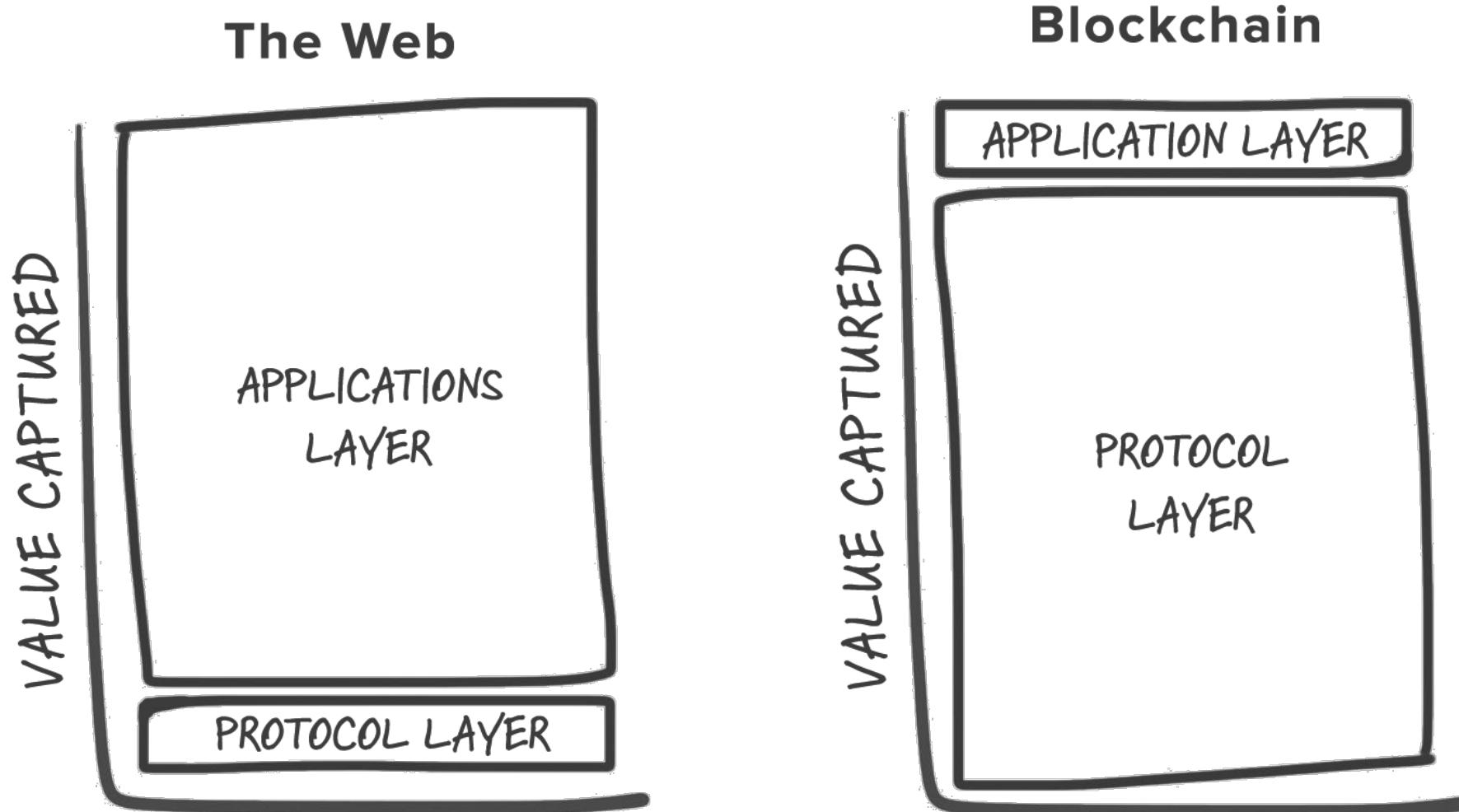
웹: 애플리케이션(아마존, 구글, 페이스북)에 가치가 집중

블록체인: dApp을 위한 **인프라\*\***로서 프로토콜에 가치가 집중

\* Joel Monégro

\*\* Shared data, Cryptographic access token

# Blockchain protocol is fat

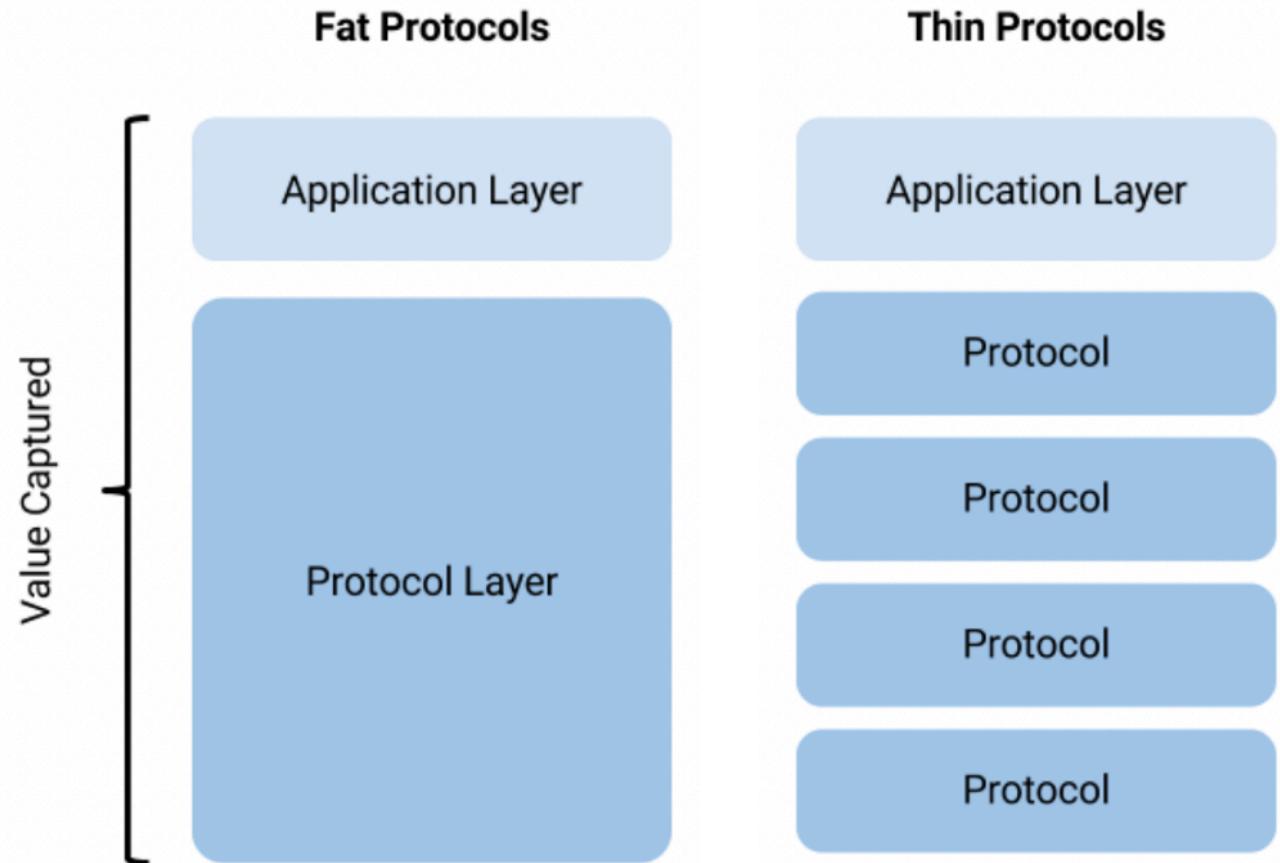


# Thin protocol (Teemu Paivinen)

Protocols in aggregate is fat

Divided by **multiple protocols**

**Forking** competitive market

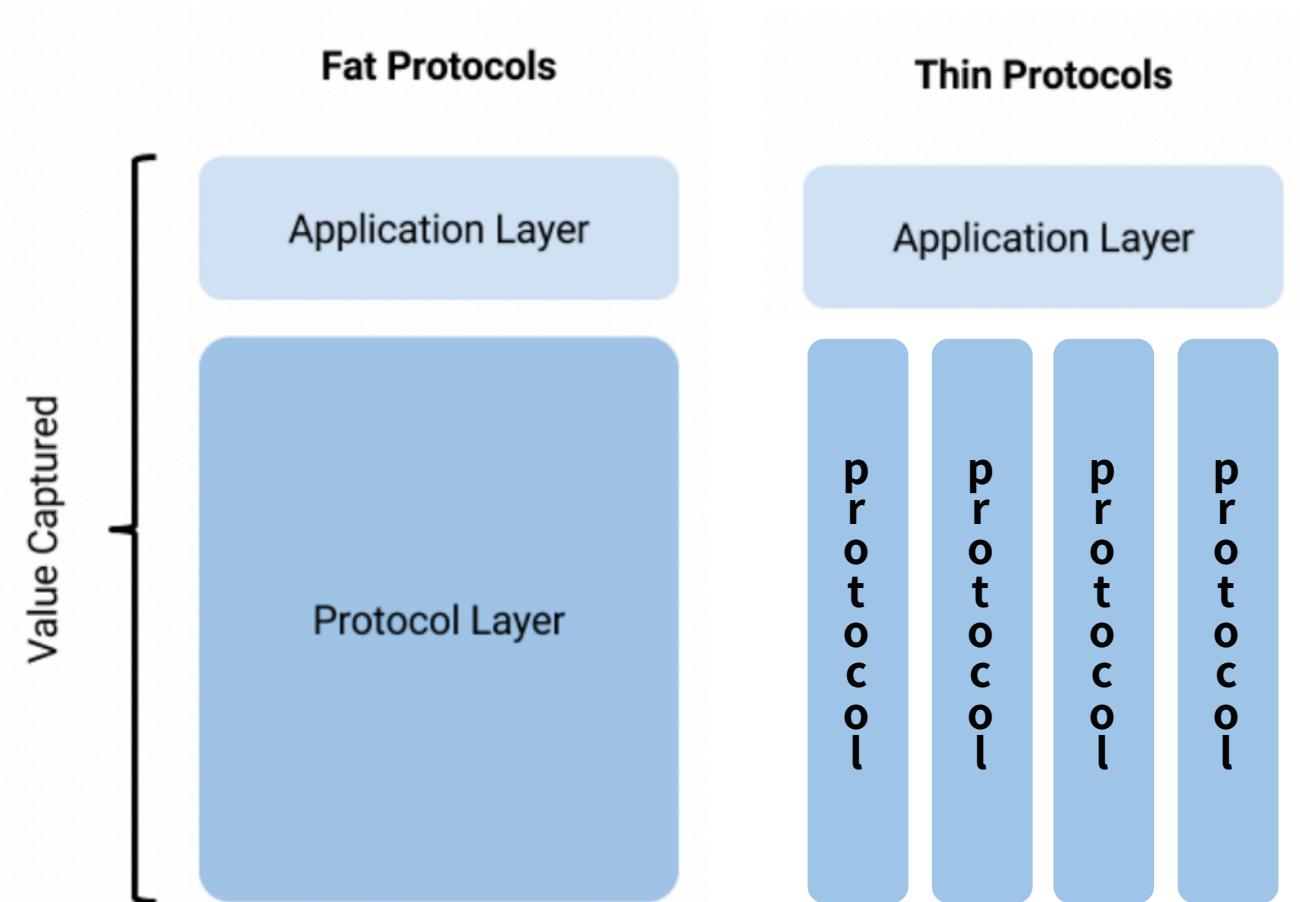


# Thin protocol

Protocols in aggregate is fat

Divided by **multiple protocols**

**Forking** competitive market



# Bitcoin is **anti-fragile**

마운트곡스 파산

14년 2월 85만개의 비트코인 도난  
(4억7400만 달러, 5천억)

비트코인 네트워크 자체에 영향 X

일반 투자자는 기피

VC들은 투자

(14년 3.6억 달러, 15년 6.5억 달러)

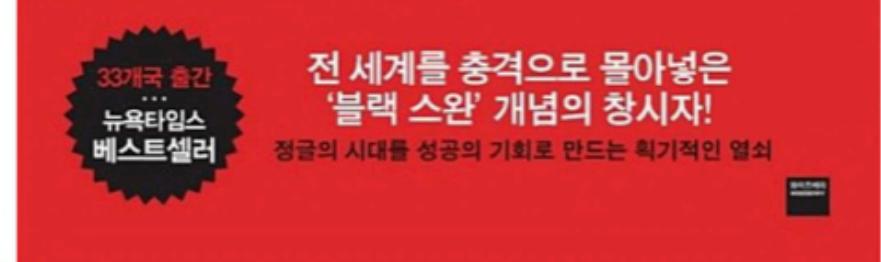
불확실성과 충격을 성장으로 이끄는 힘

# 안티프래질

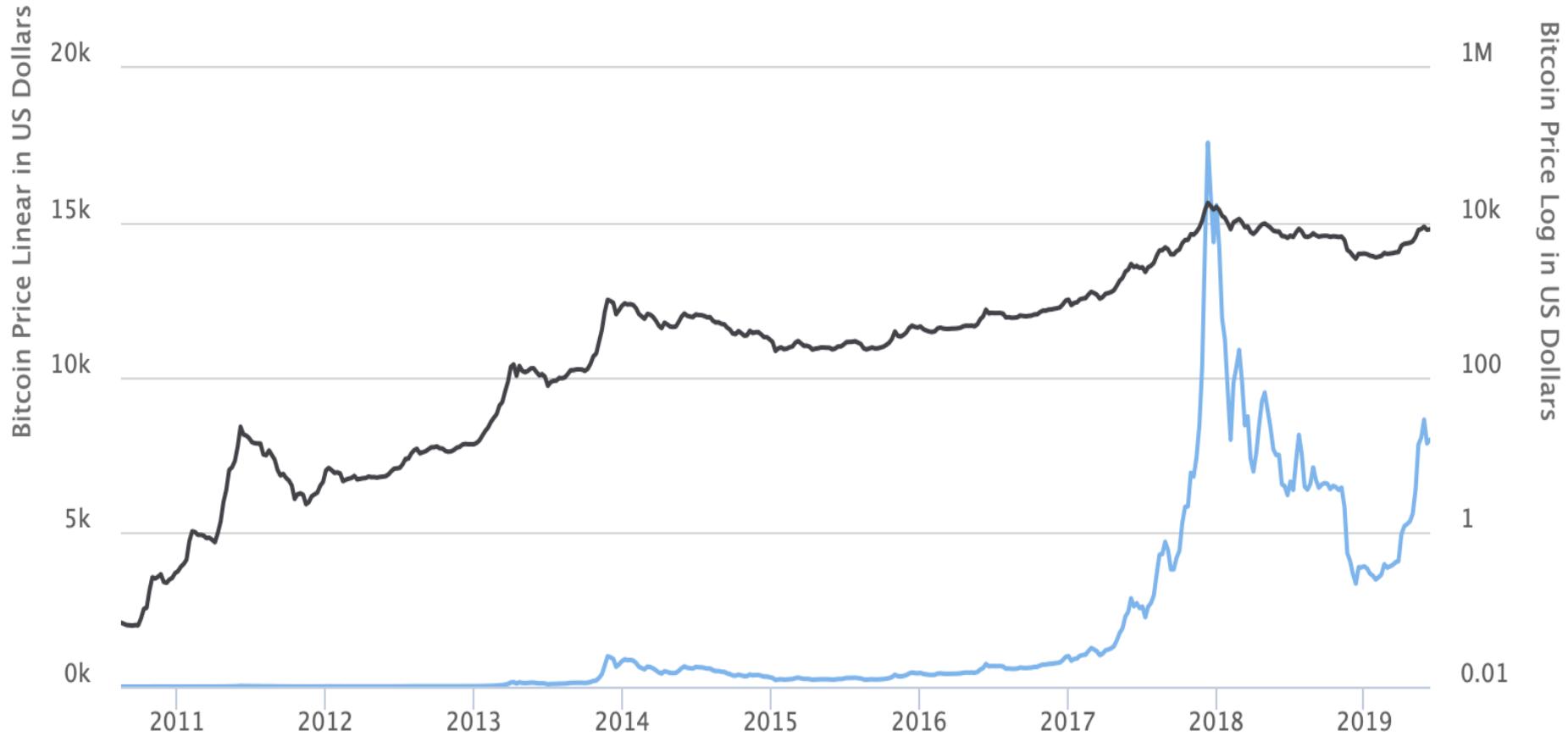
Antifragile



나심 니콜라스 탈레브  
안세민 옮김



# Bitcoin is **anti-fragile**



## 블록체인이란 무엇인가? Open Distributed Ledger, 거래, 효율성, 신뢰, 중앙화

비트코인의 등장	거인의 어깨, Byzantine generals, Hash, Digital signature, Chain
개발 동기	제네시스 블록(금융위기), Cyberpunk, 중앙화, 신뢰, 프라이버시, 암호학
중앙화의 문제	Trust, Single point of failure, Tyranny, Overhead cost
중개자 없는 P2P 거래	직접 거래, 대다수가 정직, 다수결, 합의 알고리즘, 컴퓨팅 파워, 보상 인센티브
특징	Decentralization, Immutability , Transparency, Programmability
의의	DLT, IoV, DAO, Trustless, Bootstrapped, Fat protocol, Anti-fragile

**HOW**

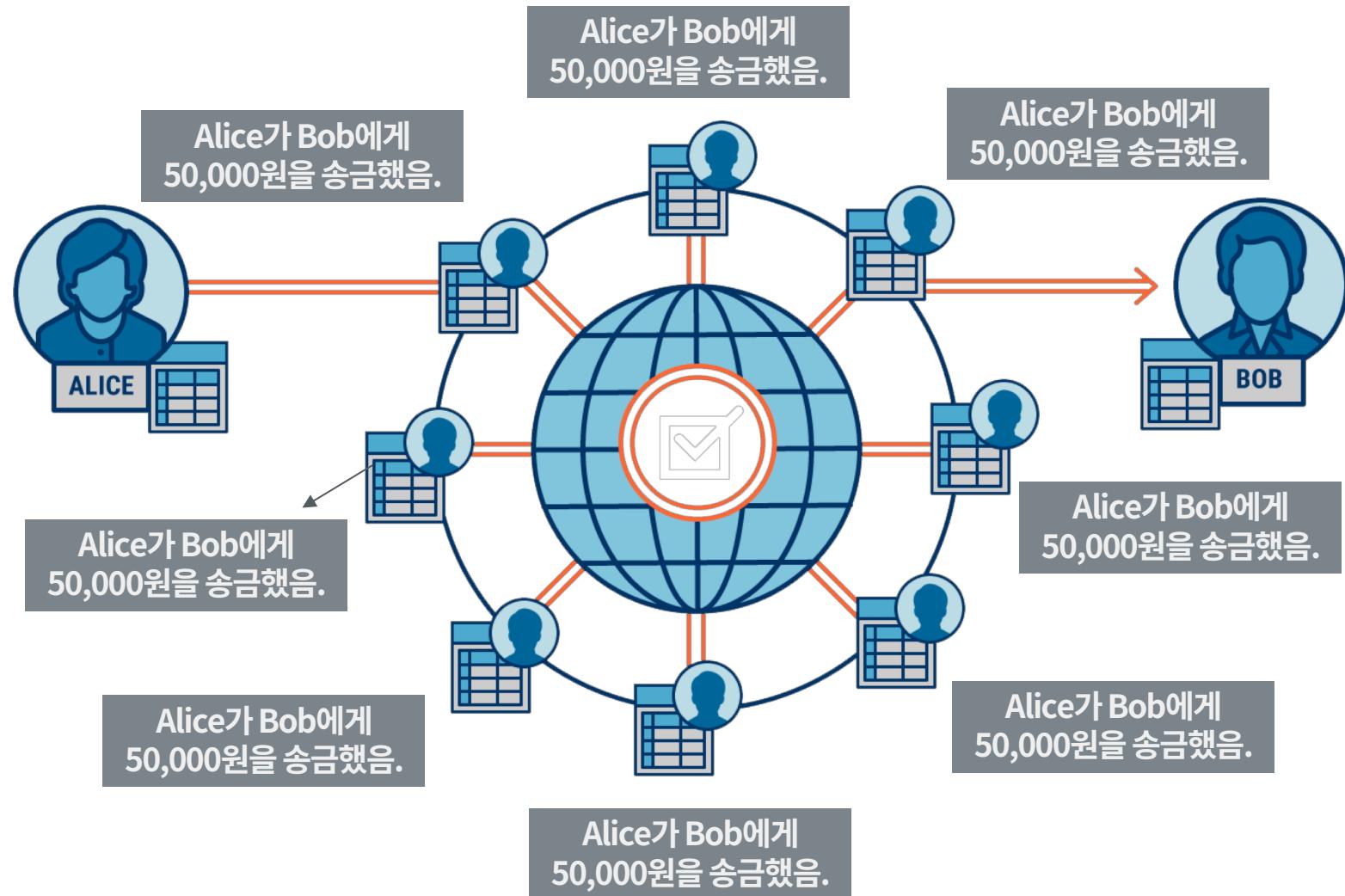
# **CHAPTER2 합의 알고리즘과 거버넌스**

## 중개자 없는 P2P 거래: Trustless

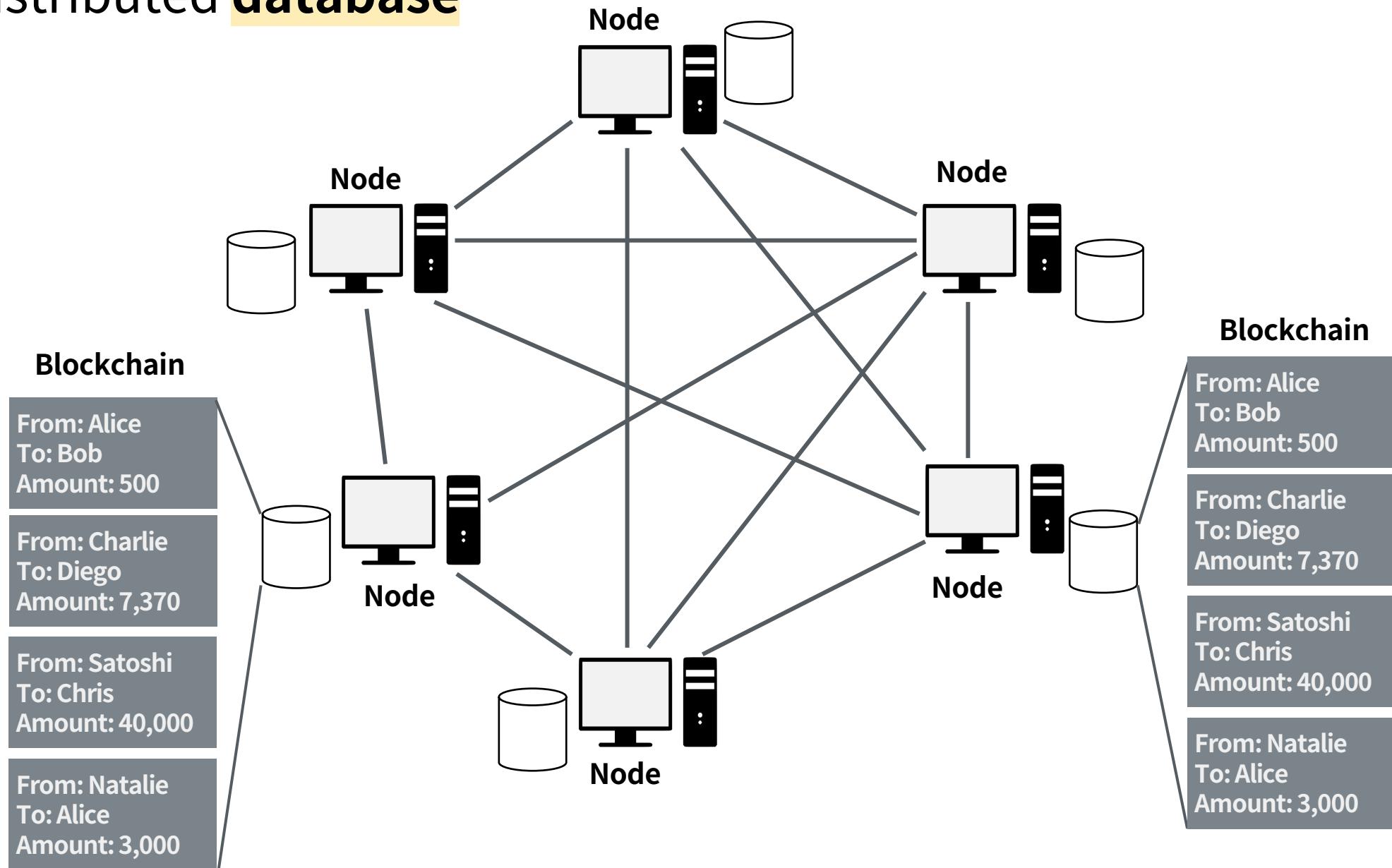
I've developed a new open source **P2P** e-cash system called Bitcoin. It's completely decentralized, with no central server or trusted parties.

Users hold the crypto keys to their own money and transact directly with each other, with the help of the **P2P network to check for double-spending**.

# Open distributed ledgers



# Open distributed database



## 분산 장부의 목표: 동일한 장부의 유지

언제나 대다수의 노드가 동일한 장부에 합의  
합의를 위해서는 투표와 같은 다수결 규칙이 필요  
제한된 (폐쇄, 소규모) 네트워크에서 해결

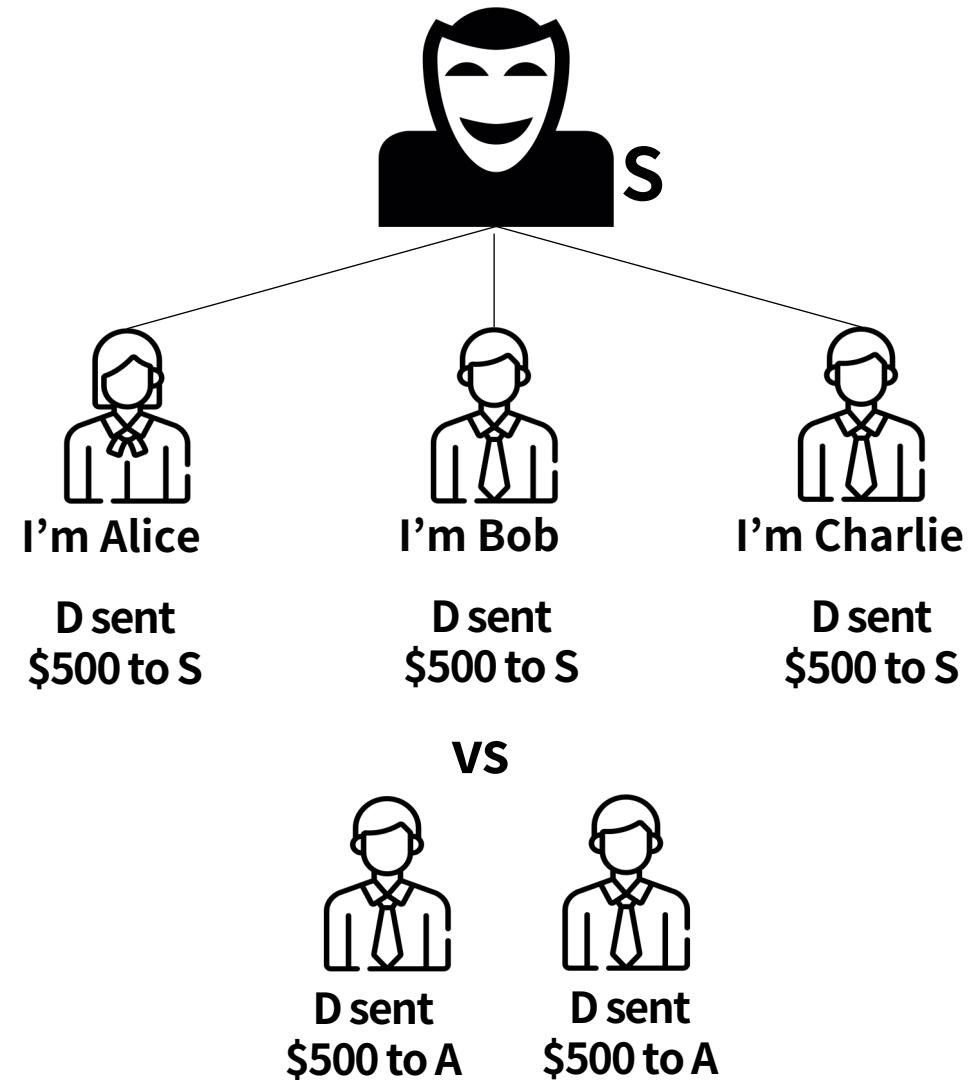
# 합의 알고리즘



# 분산 장부의 목표: 동일한 장부의 유지

## Sybil attack

공개 네트워크의 한계



**규칙이 필요**

# Blockchain protocol

누군가가 내 돈을  
함부로 쓰지는 않을까

기록하려는 사람이  
아무도 없으면?

진짜 동일한 기록일까  
아니라고 발뺌하면?

## RULES

---

- 1.
- 2.
- 3.

# 해결책: 암호학 + 게임이론 + 컴퓨터 과학

누군가가 내 돈을  
함부로 쓰지는 않을까

디지털 서명

기록하려는 사람이  
아무도 없으면?

기록에 대한 보상

진짜 동일한 기록일까  
아니라고 발뺌하면?

해시 함수 + PoW

# 해결책: 암호학 + 게임이론 + 컴퓨터 과학

누군가가 내 돈을  
함부로 쓰지는 않을까

디지털 서명

기록하려는 사람이  
아무도 없으면?

기록에 대한 보상

진짜 동일한 기록일까  
아니라고 발뺌하면?

해시 함수 + PoW

# Encryption and Key Scheme (ref #6)

Symmetric-Key Scheme



Asymmetric-Key Scheme



# Encryption and Key Scheme

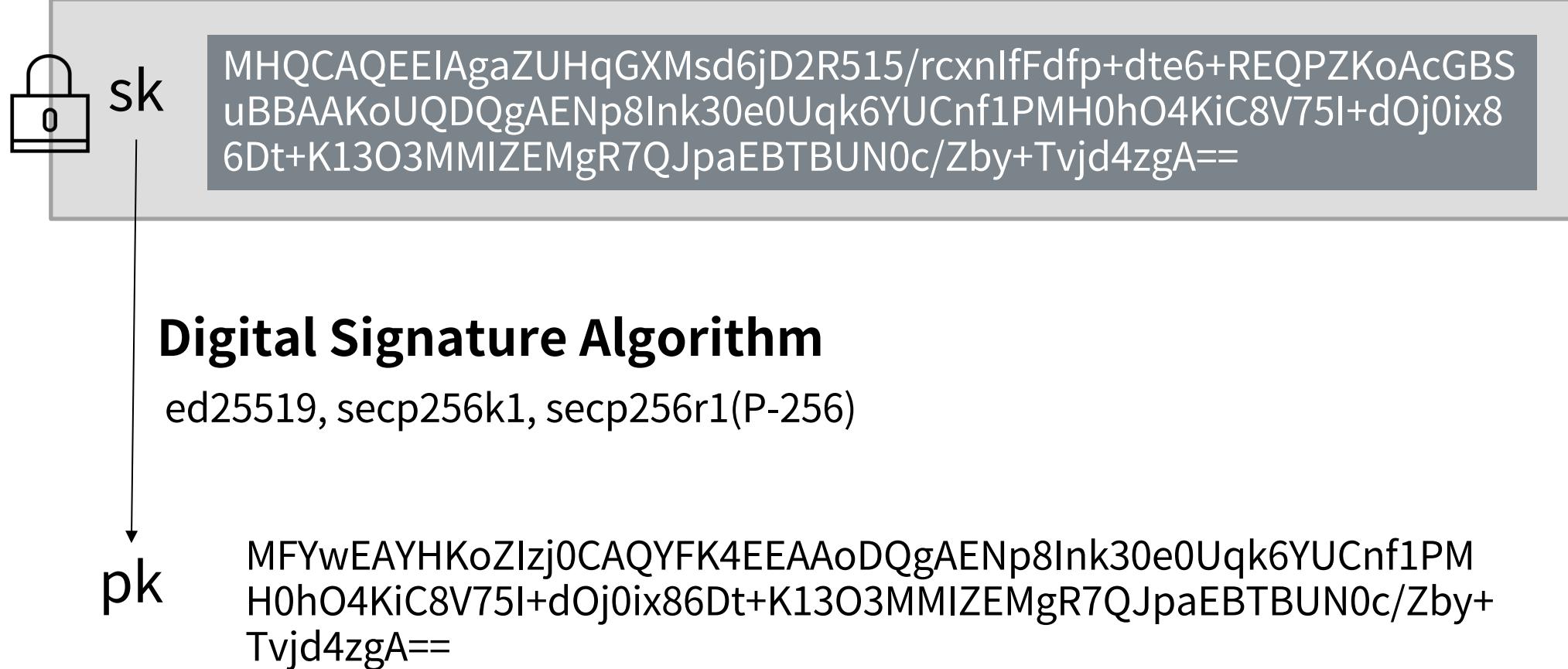
## Symmetric-Key Scheme



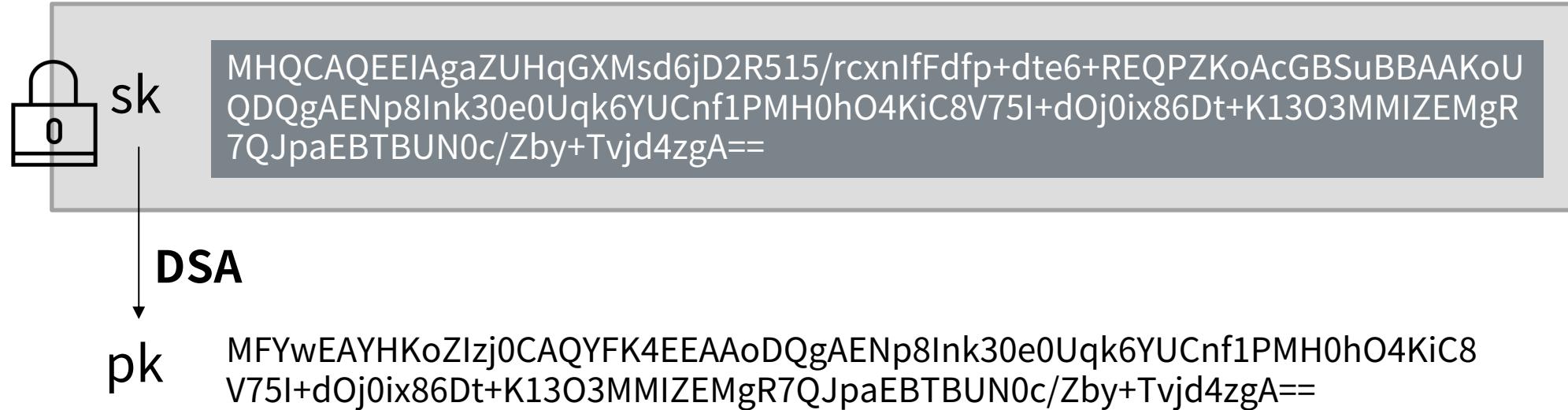
## Asymmetric-Key Scheme



# Digital signature: Pk는 Sk로부터 (거의) 유일하게 계산된다 (ref #7)



# Digital signature: Sk로 서명하고, 이 서명과 Pk로 검증한다 (ref #7)



$\text{Sign}(\text{Message}, \text{sk}) = \text{Signature}$

$\text{Verify}(\text{Message}, \text{Signature}, \text{pk}) = \text{T/F}$

Digital signature: 서명은 메시지마다 (거의) 유일하게 결정된다 (ref #7)

$\text{Sign}(\text{Message}, \text{sk}) = \text{Signature}$

I will give you  
**500** dollars



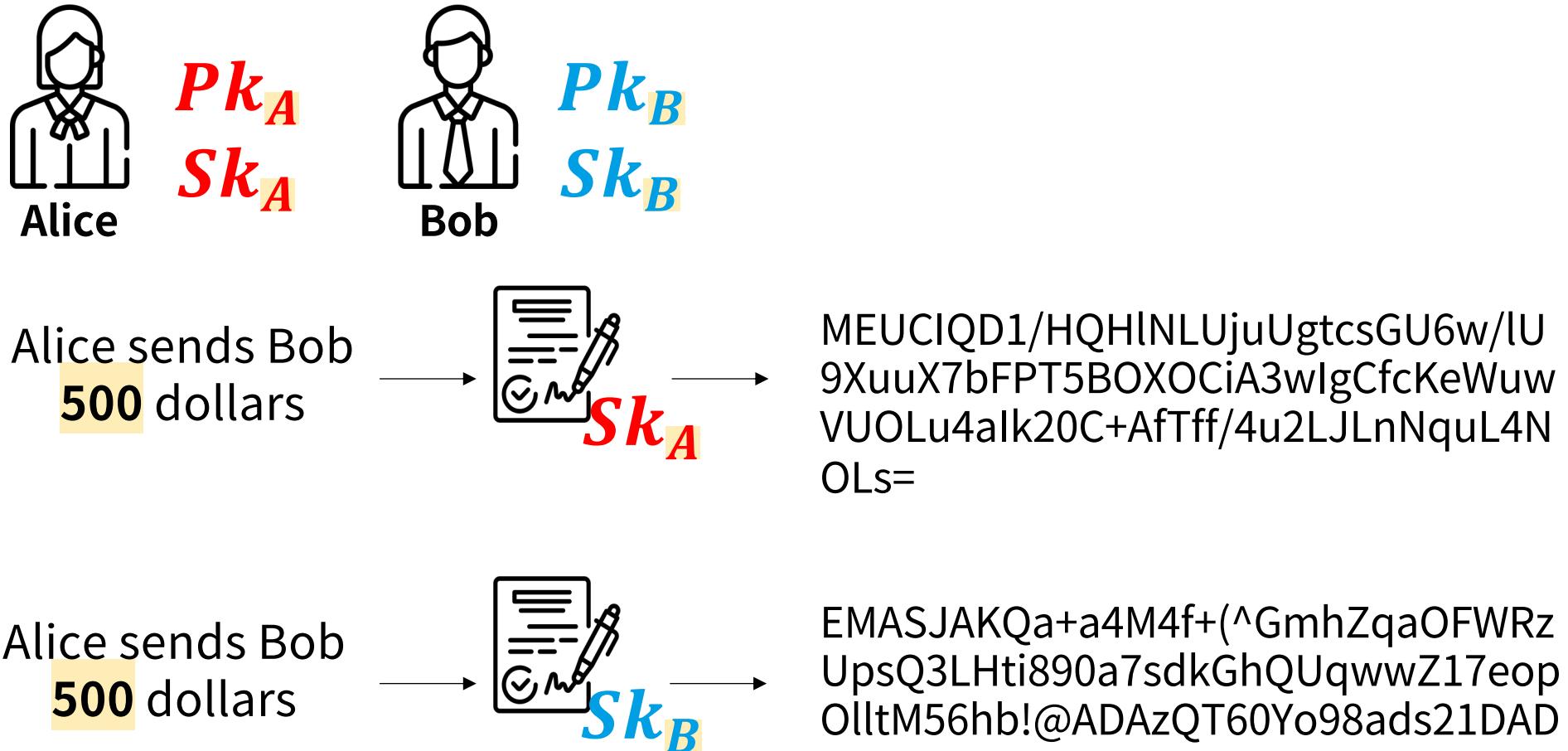
MEUCIQD1/HQHlNLUjuUgtcsGU6w/lU9X  
uuX7bFPT5BOXOCiA3wlgCfcKeWuwVUOL  
u4alk20C+AfTff/4u2LJLnNquL4NOLs=

I will give you  
**5000** dollars



MEQCIHGa+a4M4f+BuKmhZqaOFWRzUps  
Q3LHti83lSpDHkhQUAiBcZ17eopOlltMuW  
ZzFXnxo9AzQT60Y2mQpP39WbDW2Vg==

# Digital signature: 동일한 메시지를 다른 Sk로 서명하면 결과가 다르다 (ref #7)



Digital signature: 서명에 사용된 Sk에 대응하는 Pk만이 True를 리턴한다 (ref #7)



Sign(Message,  $Sk_A$ ) = *Signature*

Verify(Message, *Signature*,  $Pk_A$ ) = **True**

Verify(Message, *Signature*,  $Pk_B$ ) = **False**

# Digital signature

서명은 메시지마다 (거의) 유일하게 결정된다.

동일한 메시지를 다른 Sk로 서명하면 결과가 (거의) 다르다.

서명에 사용된 Sk에 대응하는 Pk만이 True를 리턴한다.

**나만이 내 행동(돈)에 대해 서명할 수 있다.  
이미 서명한 이상 반박할 수는 없다.**

거의 유일하게 = 그렇지 않은 경우가 현실적으로 불가능 (**Infeasible**)

$$2^{256} = (2^{32})^8 \quad 1600\text{경}$$

$$\doteq (40\text{억}) (40\text{억}) (40\text{억}) (40\text{억}) (40\text{억}) (40\text{억}) (40\text{억}) (40\text{억})$$

최고급 GPU  $\doteq$  초당 약 10억 번

GPU 4대(**40억 번**) 장착한 PC **40억 개**

**40억 명**

**40억 개**의 지구

**40억 초**  $\doteq$  126.8년

126.8년 \* **40억**  $\doteq$  5,070억 년

거의 유일하게 = 그렇지 않은 경우가 현실적으로 불가능 (**Infeasible**)

$$2^{256} = (2^{23})^{11.\text{xx}} = (2^{18})^{14.\text{xx}}$$

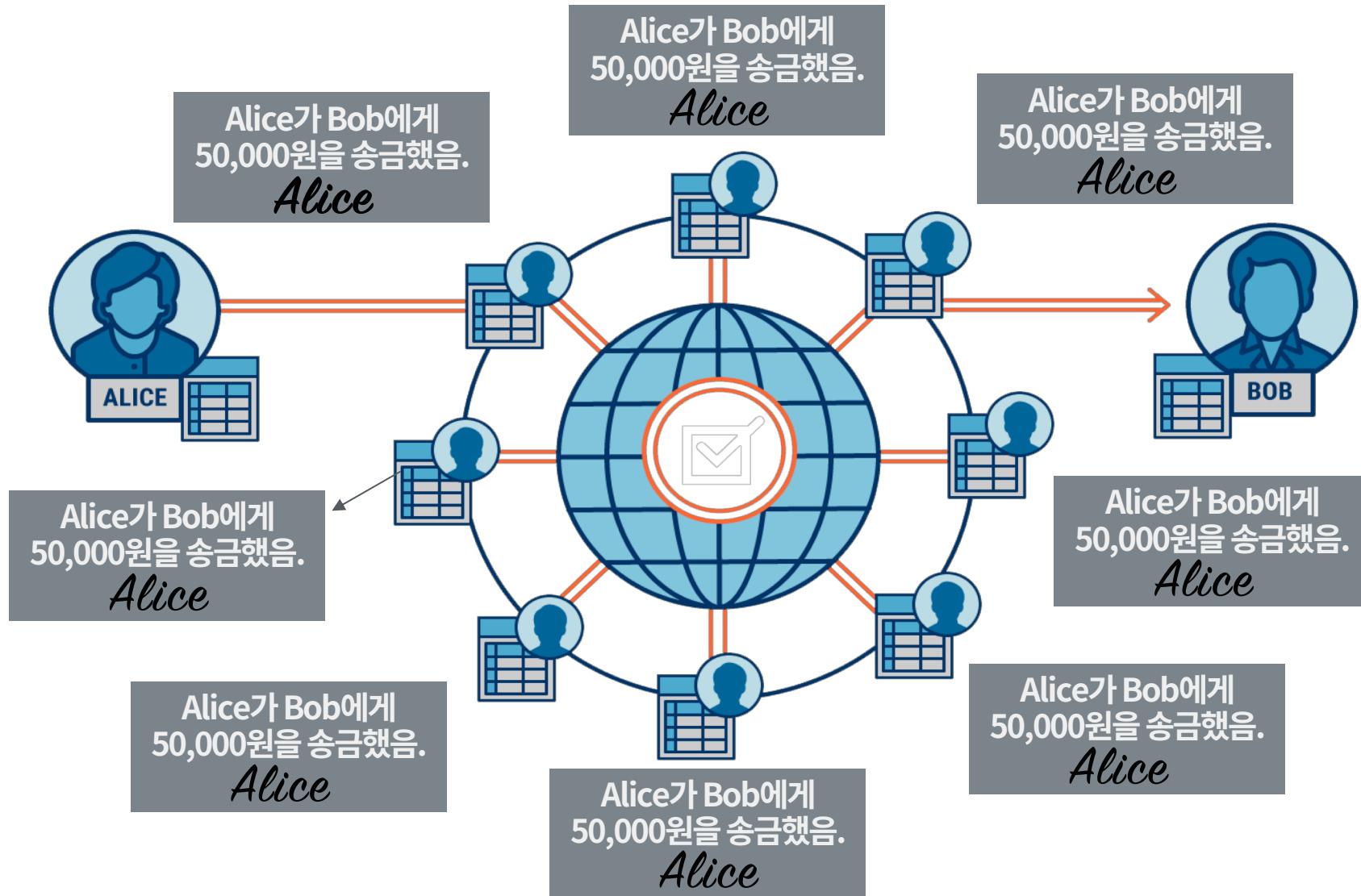
로또 1등 맞을 확률 =  $1 / 8,145,060 \doteq 1 / 2^{23}$

로또 1등을 연속으로 **11번** 당첨될 확률

벼락 맞을 확률  $\doteq 1 / 280,000 \doteq 1 / 2^{18}$

벼락을 연속으로 **14번** 맞을 확률

# Distributed ledgers with digital signature



# RULES

---

1. 비트코인을 송금하는 사람은 거래에 자신의 디지털 서명을 포함시켜야 한다.



$Pk_A$   
 $Sk_A$

Sign('Alice sends 10 btc to Bob',  $Sk_A$ ) = *Alice's Signature*

# 해결책: 암호학 + 게임이론 + 컴퓨터 과학

누군가가 내 돈을  
함부로 쓰지는 않을까

디지털 서명

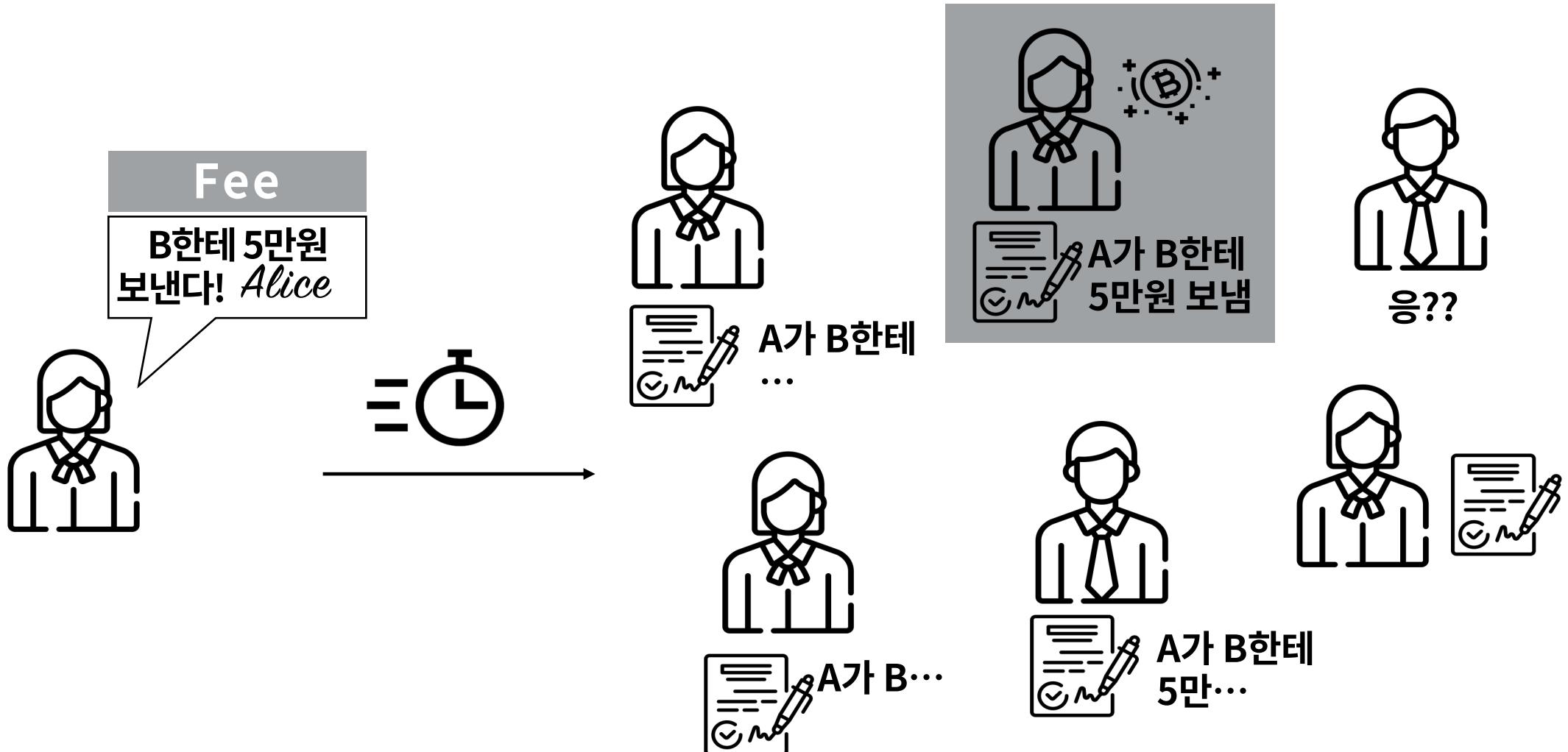
기록하려는 사람이  
아무도 없으면?

기록에 대한 보상

진짜 동일한 기록일까  
아니라고 발뺌하면?

해시 함수 + PoW

# 제일 먼저 기록한 사람이 보상을 가져간다

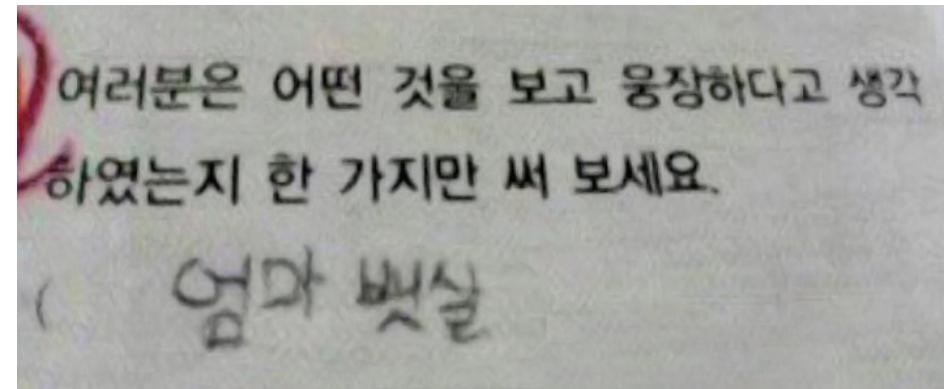
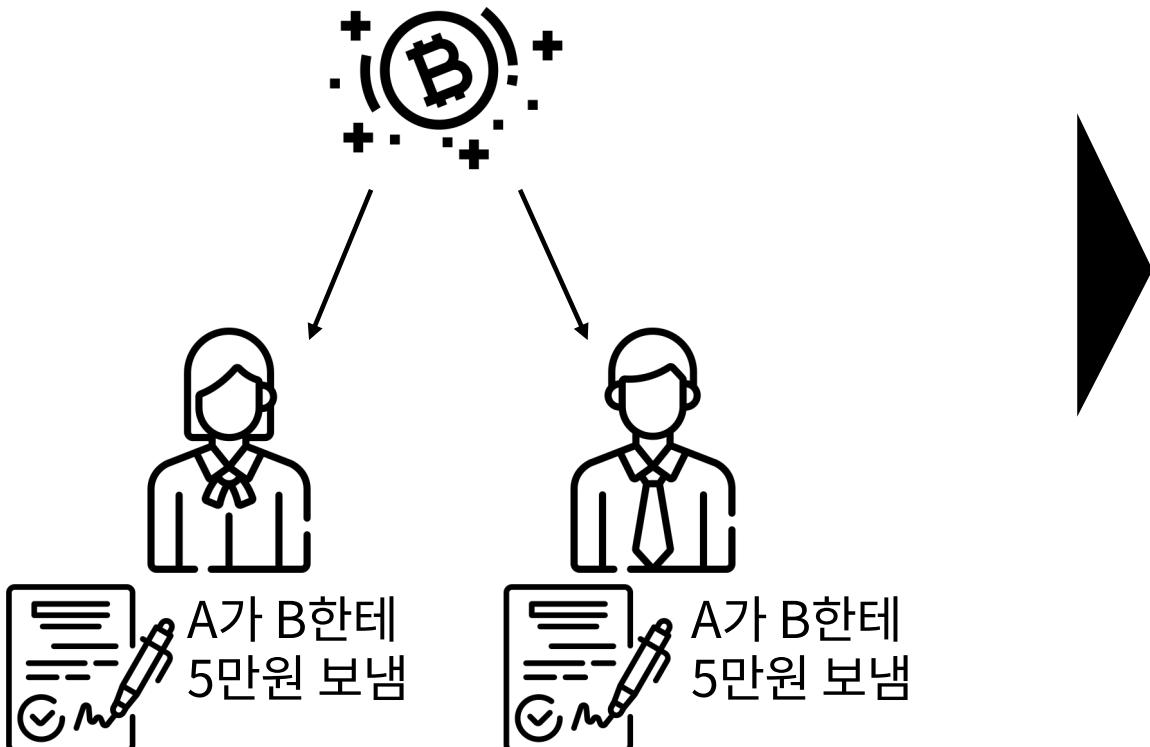


# 제일 먼저 기록한 사람이 보상을 가져간다

제일 먼저???

경쟁

문제를 **가장 빨리**  
**맞춘** 사람이 승리



# RULES

---

1. 비트코인을 송금하는 사람은 거래 기록에 자신의 디지털 서명을 포함시켜야 한다.
2. 문제를 가장 빨리 맞춘 사람이 기록이 공식 기록으로 인정되고,  
기록의 대가로 보상을 받는다.

# RULES

---

1. 비트코인을 송금하는 사람은 거래에 자신의 디지털 서명을 포함시켜야 한다.
2. 문제를 가장 빨리 맞춘 사람이 기록이 공식으로 인정되고, 그 대가로 보상을 받는다.

# 해결책: 암호학 + 게임이론 + 컴퓨터 과학

누군가가 내 돈을  
함부로 쓰지는 않을까

디지털 서명

기록하려는 사람이  
아무도 없으면?

기록에 대한 보상

진짜 동일한 기록일까  
아니라고 발뺌하면?

**해시 함수 + PoW**

함수의 결과값을 보고 입력값 맞추기

?



$$f(x) = x + 2$$



10

## 함수의 결과값을 보고 입력값 맞추기 (Hash Puzzle)



# (Cryptographic) Hash Function (ref #8)

임의의 길이의 데이터를 고정된 길이의 데이터로 맵핑하는 함수

**Deterministic**

$$x = y \Rightarrow h(x) = h(y)$$

Cryptographic  
Hash Function

**Collision resistance**

같은 결과값을 갖는 서로 다른 입력값을 찾기가 실질적으로 불가능

$$x \neq y \Rightarrow h(x) \neq h(y)$$

**Hiding**

결과값을 가지고 입력값을 찾기가 실질적으로 불가능

**Puzzle friendliness**

무작위로 찾는게 현재로선 최선

# Secure Hash Algorithm with 256 bits (ref #8)

sha256 generator

전체 이미지 동영상 뉴스 지도 더보기 설정 도구

검색결과 약 571,000개 (0.21초)

## SHA256 Online

<https://emn178.github.io/online-tools/sha256.html> ▾ 이 페이지 번역하기

SHA256 online hash function. ... SHA256 online hash function. Auto Update. Hash. CRC-16 · CRC-32 · MD2 · MD4 · MD5 · SHA1 · SHA224 · SHA256 · SHA384 ...

# Secure Hash Algorithm with 256 bits (ref #8)

SHA256 online hash function

1

Hash  Auto Update

6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b

SHA256 online hash function

2

Hash  Auto Update

d4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35

# Applications of SHA: Digital fingerprint, Message digest (ref #8)

내 전 재산 중 **5%**를 ○○○에 기부한다.



6BC1820A9BC0EB207F86741A22DCE5460  
C959655A73FB3BDF2581B62FEF631CF

내 전 재산 중 **95%**를 ○○○에 기부한다.



7A32C64C3AE19CEC3809E6DAC5B10A59F  
58AEFF127907446F59A8375830D573A

고대 근처 맛집 리스트

대성집: 해장국 핵존맛, 수육 캬

고른햇살: 김혜자 쓰앵님 이상의 가성비 분식

형제집: 아재감성 술집쓰. 오돌뼈, 닭도리탕...

용초수: 꿔바로우 하... 토마토볶음 오...

유자유: 김치떡볶이 + 비빔밥

오샬: 인도커리 냠냠

춘자: 핵 저렴한 술집

회기역 근처 이자카야 고우 꼭 가세요. JMT!

회기역 근처 이자카야 고우 두 번 가세요!!!

...



9DC44C08F26189C2DB2ECB5B0711348D  
CDE76DAC6227F39ACB33CB0882BD3A6E

# Hash puzzle in Bitcoin

블록의 해시 값이 특정 조건을 만족해야 적법한(**valid**) 블록으로 인정



특정 조건을 만족하도록 블록을 만들자!

# Hash puzzle in Bitcoin

블록의 해시 값이 특정 조건을 만족해야 적법한(**valid**) 블록으로 인정

Block $\mathcal{B}$
Operations
Alice → Bob: 1btc Kim → Lee: 2.3btc

$\text{Hash}(\mathcal{B}) = 0347C308D64E2DF...D0C23B6250319800891C3E6D2$

$\text{Hash}(\mathcal{B}) = 0000\ 0011\ 0100\ 0111\ ...$

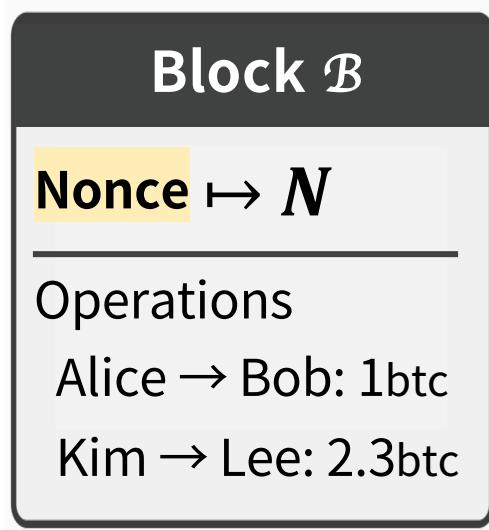
$\text{Hash}(\mathcal{B})$ 가 홀수인가?

$\text{Hash}(\mathcal{B})$ 를 7로 나눈 나머지가 2인가?

...

# Hash puzzle in Bitcoin

블록의 해시 값이 특정 조건을 만족해야 적법한(**valid**) 블록으로 인정



$\mathcal{B}_i = \text{Block } \mathcal{B} \text{ with } i \text{ as a nonce:}$

$$\mathcal{H}\text{ash}(\mathcal{B}_0) = 0110\ 1011\ 0110\ 1101\ \dots$$

$$\mathcal{H}\text{ash}(\mathcal{B}_1) = 0011\ 0011\ 0100\ 0111\ \dots$$

...

$$\mathcal{H}\text{ash}(\mathcal{B}_{98}) = 0000\ 0110\ 0110\ 0101\ \dots$$

...

$\mathcal{H}\text{ash}(\mathcal{B}_i)$ 이 5개의 0으로 시작하는  $i$ 를 찾아라

# Hash puzzle in Bitcoin

$\text{Hash}(\mathcal{B}_i)$ 이  $n$ 개의 0으로 시작하는  $i$ 를 찾아라

Block $\mathcal{B}$
Nonce $\mapsto N$
Operations
Alice $\rightarrow$ Bob: 1btc
Kim $\rightarrow$ Lee: 2.3btc

$\mathcal{B}_i$  = Block  $\mathcal{B}$  with  $i$  as a nonce

256 0s and 1s

$\text{Hash}(\mathcal{B}_i) = \boxed{0110\ 1011\ 0110\ 1101\ \dots\ 0110\ 1101\ 1111}$

$n$ 이 충분히 커지면, 확률적으로  $i$  역시 커진다

# RULES

---

1. 비트코인을 송금하는 사람은 거래에 자신의 디지털 서명을 포함시켜야 한다.
2. 블록의 해시가 n개의 0으로 시작하도록 만드는 논스를 가장 먼저 찾은 사람이 기록이 공식으로 인정되고, 그 대가로 보상을 받는다.
3. 각 블록은 오퍼레이션과 논스로 구성된다.

# Mining competition & Coinbase Tx



Miner A



SHA256

001010100111100010011101101010010100100011  
010011010100010101100011010111011000011110  
1011100001110011100000110001000001110011101  
110010110110111010000100010110110001111110  
1010110011000110111100100100110011111010100  
0010101111010000000100111111001011101101



Miner B



SHA256

0000000001110010100111100010011101101001  
0100100011010011010100001010110001101011101  
1000000001110011100000110001000001110011101  
110010110110111010000100010110110001111110  
1010110011000110111100100100110011111010100  
0010101111010000000100111111001011101101

## Mining difficulty

$n$ 개의 bit로 나타낼 수 있는 경우의 수

$$\boxed{11111\dots11111} = 2^n$$

## Mining difficulty

$n$ 개의 bit 중 앞  $i$ 개가 0일 때,  
나타낼 수 있는 경우의 수

$$\frac{00001\dots11111}{\begin{matrix} n \\ i \qquad n-i \end{matrix}} = 2^{n-i}$$

## Mining difficulty

256개의 bit 중 앞 96개가 0일 때

Output space of SHA256

000000000011111111...11111111

target space

$2^{96}$

probability

$2^{160}$

2의 160승도 여전히 매우 큰 수

# 2<sup>160</sup> 비트코인 지갑의 개수 (RIPE-MD160)

= 1,460,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000

2,045년, 지구 인구 90억명

모든 사람이 비트코인 주소 천만개 씩 소유

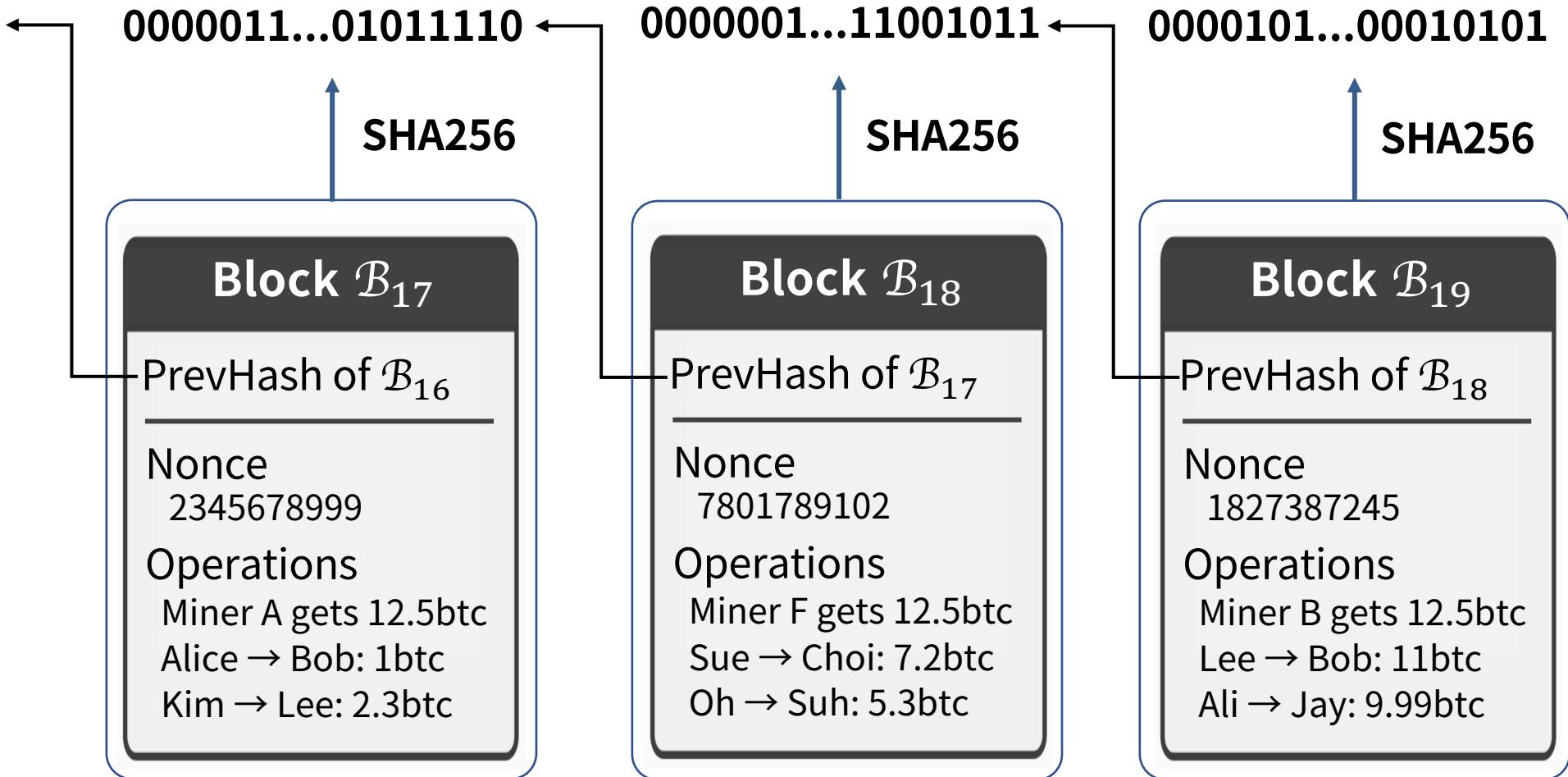
총 90,000,000,000,000,000개의 지갑

$90,000,000,000,000 / 2^{160}$

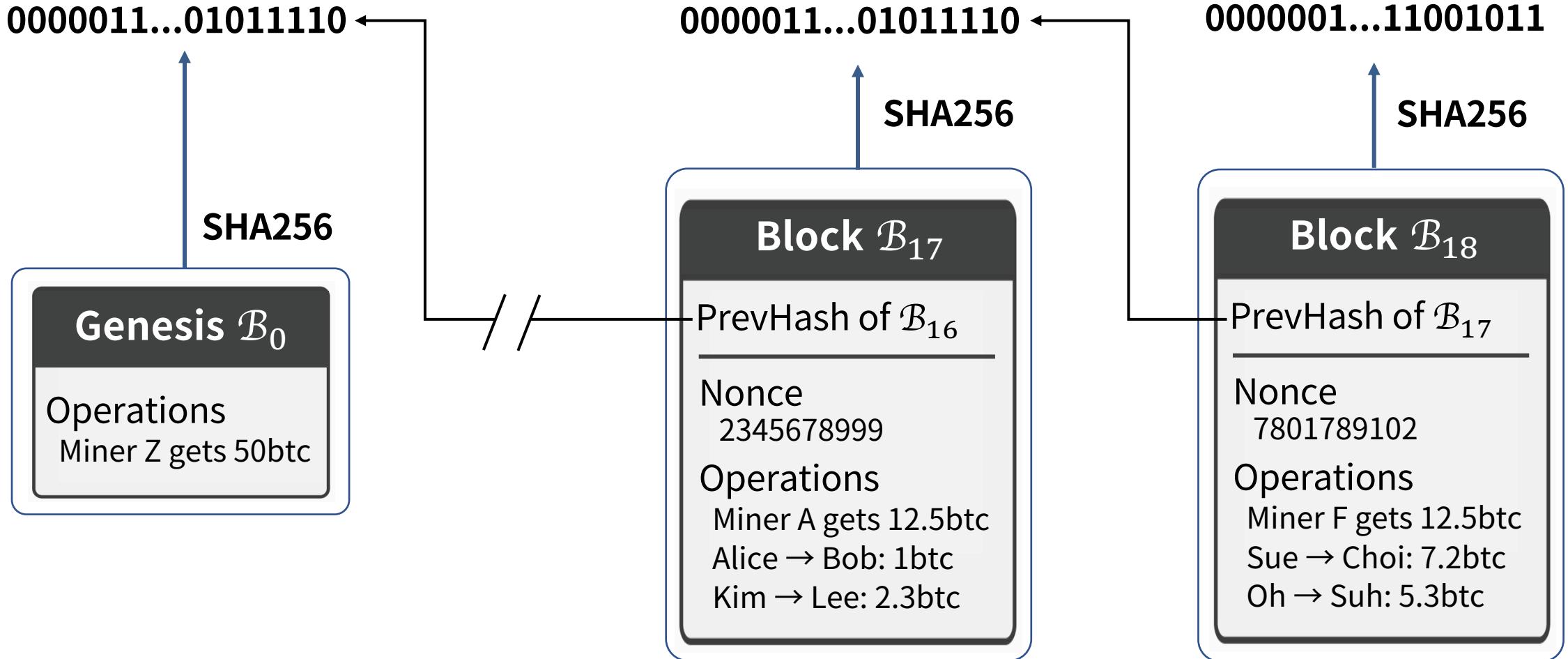
= 0.000000000000000000000000000000615%

= 전기세, 실제 비트코인이 들어있는지, 얼마나 들어있는지 등을 고려 안 됨

# Blockchain = A chain of blocks



# Blockchain = A chain of blocks



# bitcoin/src/chainparams.cpp

```
42 * Build the genesis block. Note that the output of its generation
43 * transaction cannot be spent since it did not originally exist in the
44 * database.
45 *
46 * CBlock(hash=000000000019d6, ver=1, hashPrevBlock=0000000000000000, hashMerkleRoot=4a5e1e, nTime=1231006505, nBits=1d00ffff, nNonce=
47 *   CTransaction(hash=4a5e1e, ver=1, vin.size=1, vout.size=1, nLockTime=0)
48 *     CTxIn(COutPoint(000000, -1), coinbase 04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e63656c6c6f72206f6
49 *     CTxOut(nValue=50.00000000, scriptPubKey=0x5F1DF16B2B704C8A578D0B)
50 *   vMerkleTree: 4a5e1e
51 */
52 static CBlock CreateGenesisBlock(uint32_t nTime, uint32_t nNonce, uint32_t nBits, int32_t nVersion, const CAmount& genesisReward)
53 {
54     const char* pszTimestamp = "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks";
55     const CScript genesisOutputScript = CScript() << ParseHex("04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb64
56     return CreateGenesisBlock(pszTimestamp, genesisOutputScript, nTime, nNonce, nBits, nVersion, genesisReward);
57 }
```

# bitcoin/src/chainparams.cpp

```
45 *
46 * CBlock(hash=000000000019d6, ver=1, hashPrevBlock=0000000000000000, hashMerkleRoot=4a5e1e, nTime=1231006505, nBits
47 *     CTransaction(hash=4a5e1e, ver=1, vin.size=1, vout.size=1, nLockTime=0)
48 *         CTxIn(COutPoint(000000, -1), coinbase 04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e6
49 *             CTxOut(nValue=50.00000000, scriptPubKey=0x5F1DF16B2B704C8A578D0B)
50 *         vMerkleTree: 4a5e1e
51 */
52 static CBlock CreateGenesisBlock(
53     uint32_t nTime,
54     uint32_t nNonce,
55     uint32_t nBits,
56     int32_t nVersion,
57     const CAmount& genesisReward) {
58     const char* pszTimestamp = "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks";
59     const CScript genesisOutputScript = CScript() \
60     << ParseHex(
61         "04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384
62     << OP_CHECKSIG;
63     return CreateGenesisBlock(pszTimestamp, genesisOutputScript, nTime, nNonce, nBits, nVersion, genesisReward);
64 }
```

# Blockchain Demo: Finding nonce (ref #9)

The screenshot shows a web browser window titled "Blockchain Demo" at the URL <https://anders.com/blockchain/blockchain.html>. The browser has a dark theme. The page features a navigation bar with tabs: Hash, Block, Blockchain (which is selected), Distributed, Tokens, and Coinbase. Below the navigation bar, the word "Blockchain" is displayed in large letters.

The main content area displays three blocks of a blockchain:

- Block 1:** Number 1, Nonce 40546, Data "Hello".  
Prev: 00  
Hash: 0000b3db41cb3918560115ce7300a08e78f9ae0444!
- Block 2:** Number 2, Nonce 3303, Data "World".  
Prev: 0000b3db41cb3918560115ce7300a08e78f9ae0444!  
Hash: 0000054380cb9a1da7ce5bde1d113b5ccb32256634!
- Block 3:** Number 3, Nonce 53114, Data (empty).  
Prev: 0000054380cb9a1da7ce5bde1d113b5ccb32256634  
Hash: 00001b51a26b1b9549e4bbc66e4

Each block has a "Mine" button at the bottom.

## Blockchain

Three separate blockchain blocks are shown, each with the following fields:

- Block:** # [input field]
- Nonce:** [input field]
- Data:** [text input field]
- Prev:** [input field]
- Hash:** [input field]
- Mine:** [button]

The blocks are numbered 1, 2, and 3 from left to right. Each block contains the previous block's hash in its "Prev" field and its own unique hash in its "Hash" field.

Block	Nonce	Data	Prev	Hash
1	40546	Hello	00	0000b3db41cb3918560115ce7300a08e78f9ae0444!
2	3303	World	0000b3db41cb3918560115ce7300a08e78f9ae0444!	0000054380cb9a1da7ce5bde1d113b5ccb32256634!
3	53114	(empty)	0000054380cb9a1da7ce5bde1d113b5ccb32256634	00001b51a26b1b9549e4bbc66e4

# Blockchain Demo: Finding nonce (ref #9)

The screenshot shows a web-based blockchain demo application with a dark-themed interface. At the top, the title "Blockchain Demo" is displayed in a browser window along with the URL <https://anders.com/blockchain/blockchain.html>. Below the title, there is a navigation bar with tabs: Hash, Block, Blockchain (which is currently selected), Distributed, Tokens, and Coinbase.

The main content area is titled "Blockchain" and displays three separate blocks:

- Block 1:** Contains the number "# 1", the nonce "40546", and the data "Hello!". The previous hash is listed as all zeros ("00"). The current hash is "38878e2aac0265c09213c3639632a33f66a1225a78". A "Mine" button is at the bottom.
- Block 2:** Contains the number "# 2", the nonce "3303", and the data "World". The previous hash is "38878e2aac0265c09213c3639632a33f66a1225a78". The current hash is "8a9e8b5d8214b850d75caea9153d22f10e12c8d7a1". A "Mine" button is at the bottom.
- Block 3:** Contains the number "# 3", the nonce "53114", and the data " ". The previous hash is "8a9e8b5d8214b850d75caea9153d22f10e12c8d7a1". The current hash is "606294250a90bf0f90a0295a42c". A "Mine" button is at the bottom.

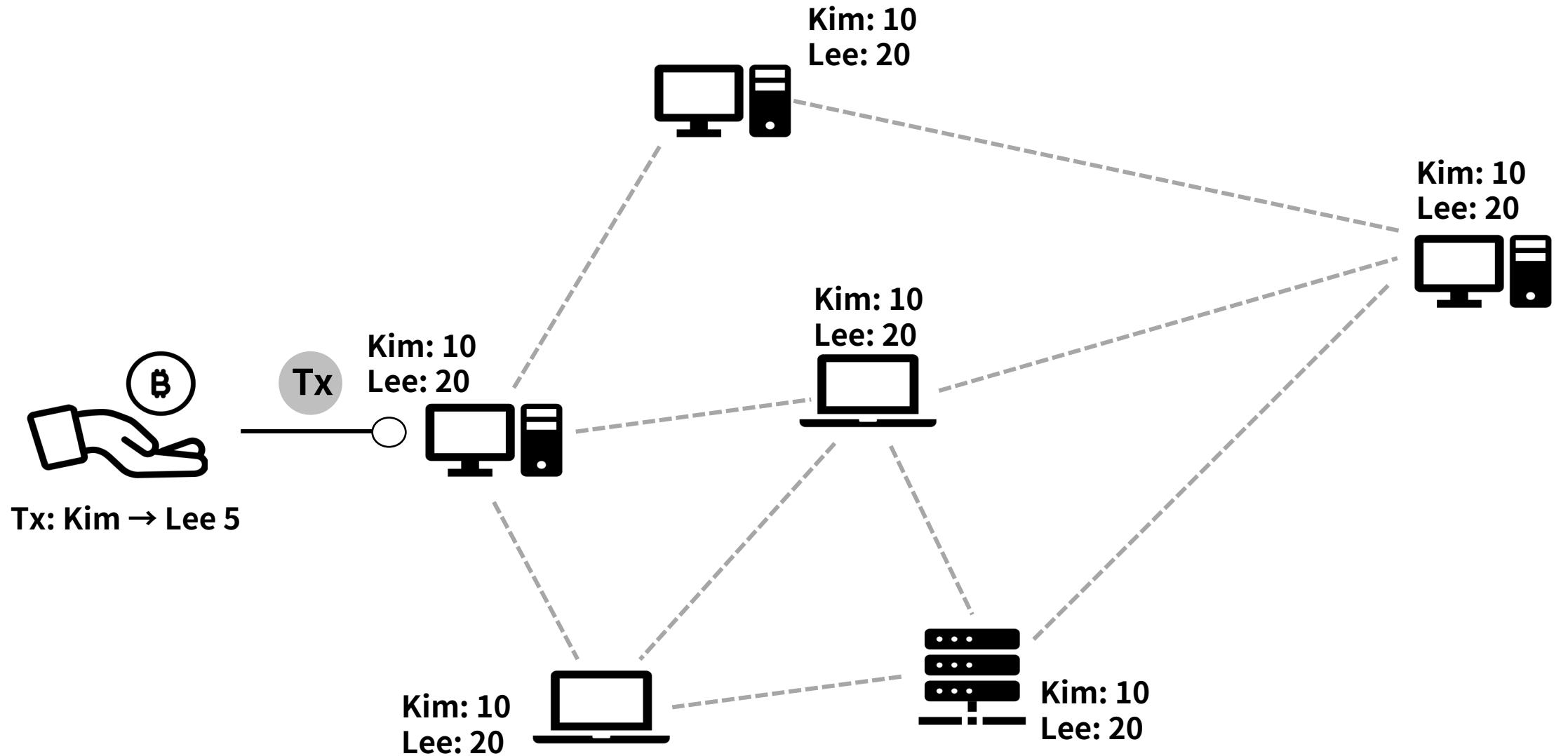
Each block also features a green circular "G" icon with a white arrow pointing right, located near the bottom right corner of each block's content area.

# RULES

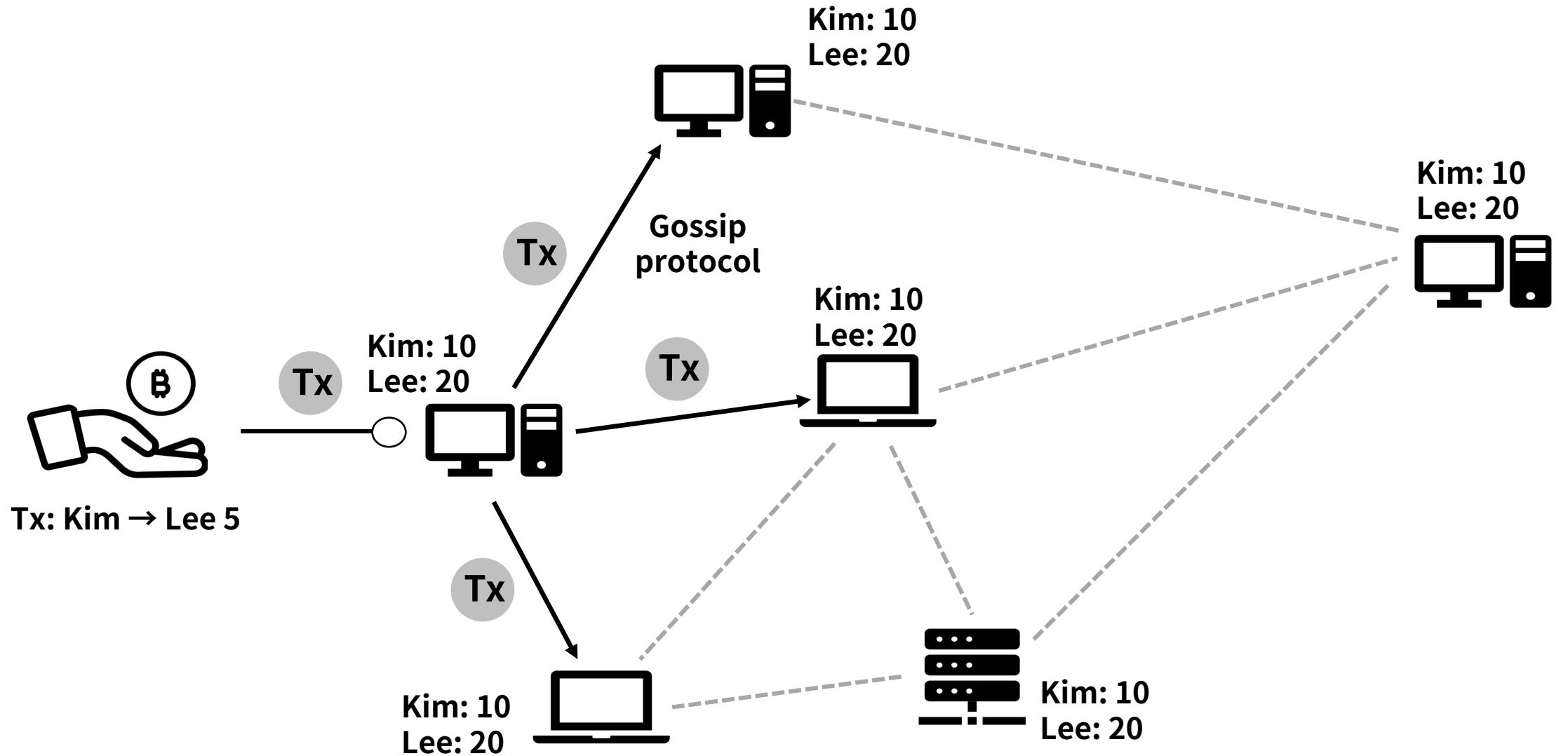
---

1. 비트코인을 송금하는 사람은 거래에 자신의 디지털 서명을 포함시켜야 한다.
2. 각 블록은 이전 블록의 해시, 오퍼레이션, 이전 블록의 해시, 논스로 구성된다.
3. 제일 먼저 블록의 해시를 난이도보다 낮게 만든 채굴자가 보상을 받는다.
4. 채굴자는 블록에 보상(코인베이스 트랜잭션)을 포함시킨다.
5. 새 블록의 해시는  $\text{Hash}(\text{이전 블록의 해시} + \text{오퍼레이션} + \text{논스})$ 이며, 논스를 바꿔가며 목표 해시를 찾는다.

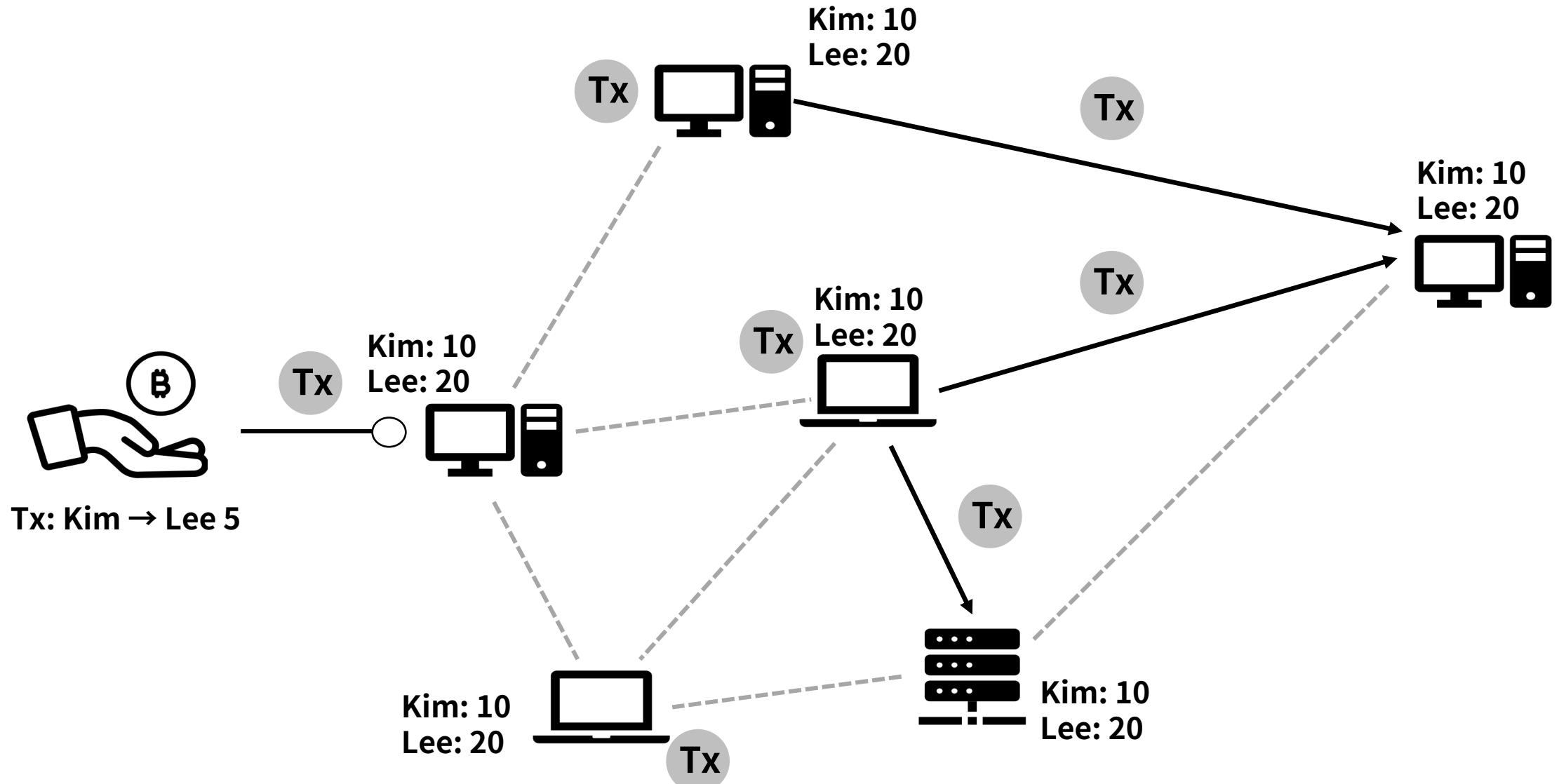
# Life cycle of transaction



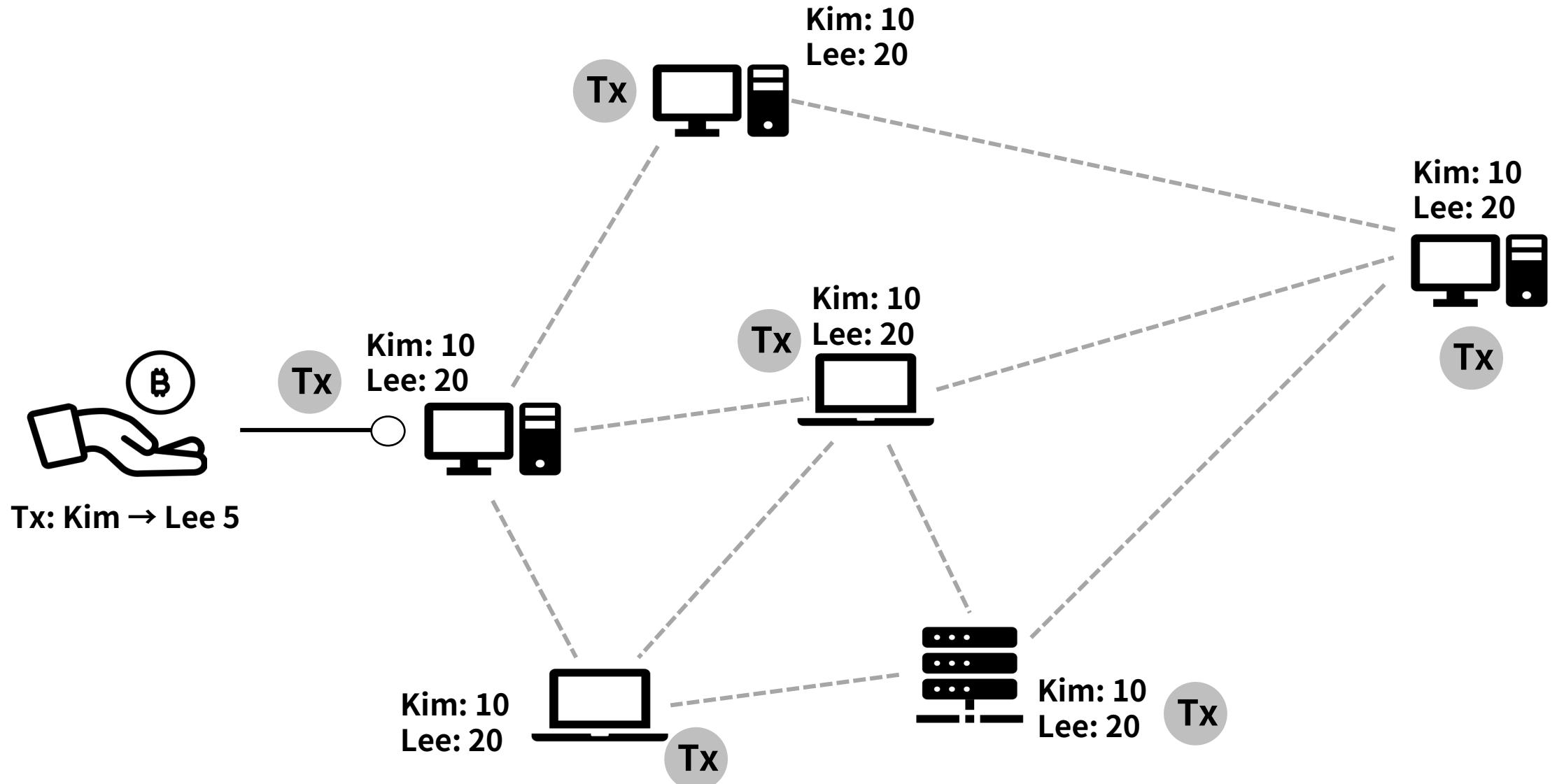
# Life cycle of transaction



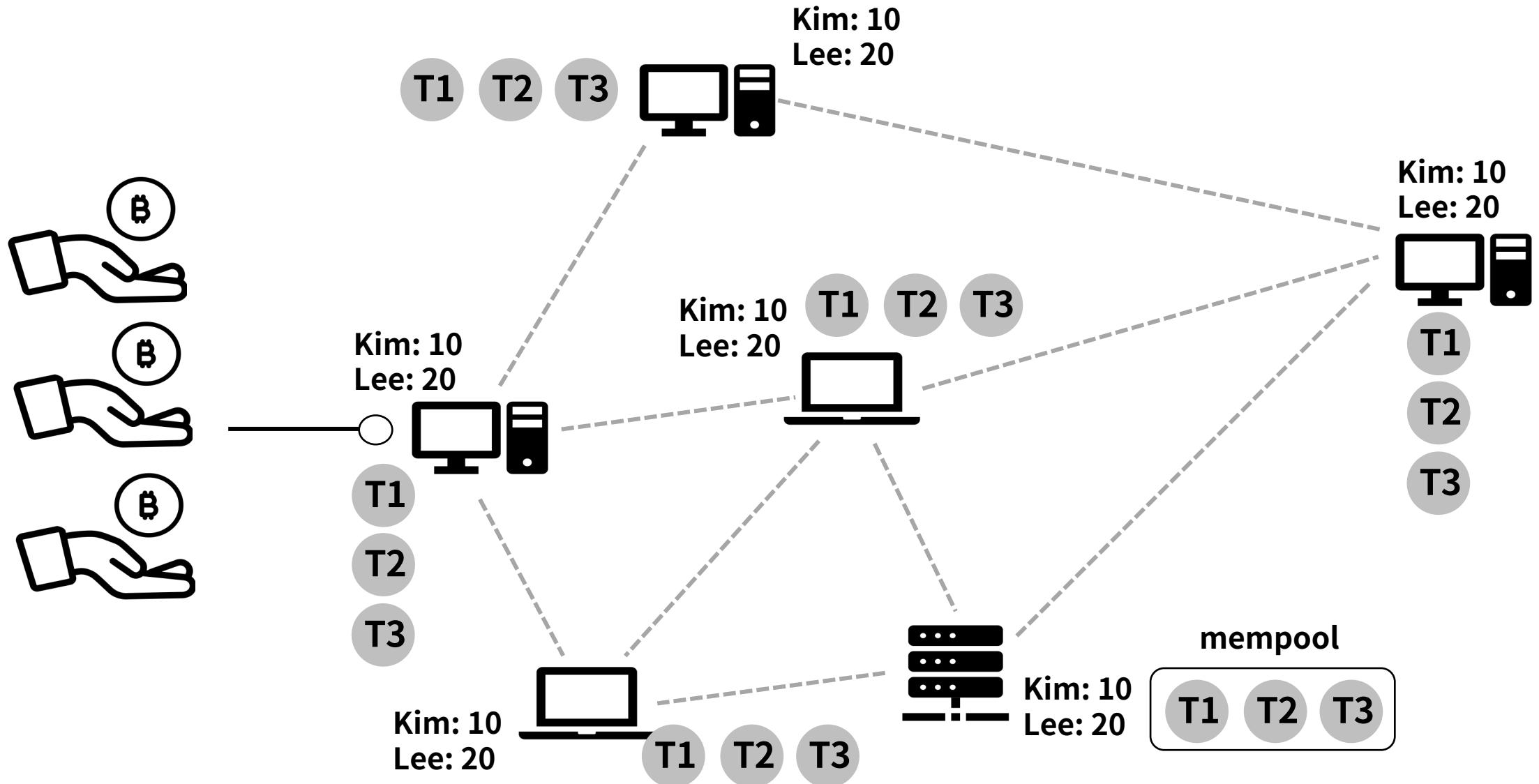
# Life cycle of transaction



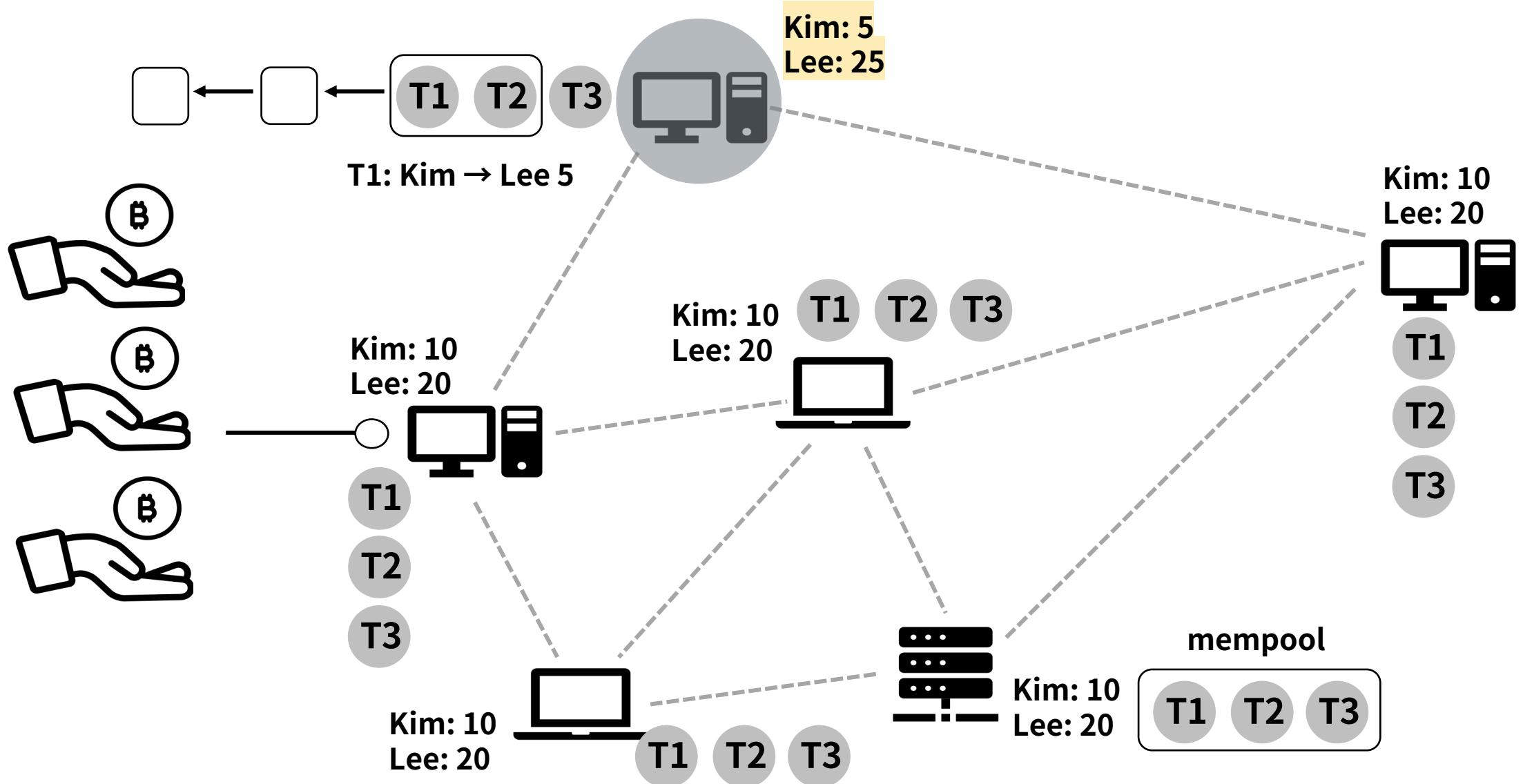
# Life cycle of transaction



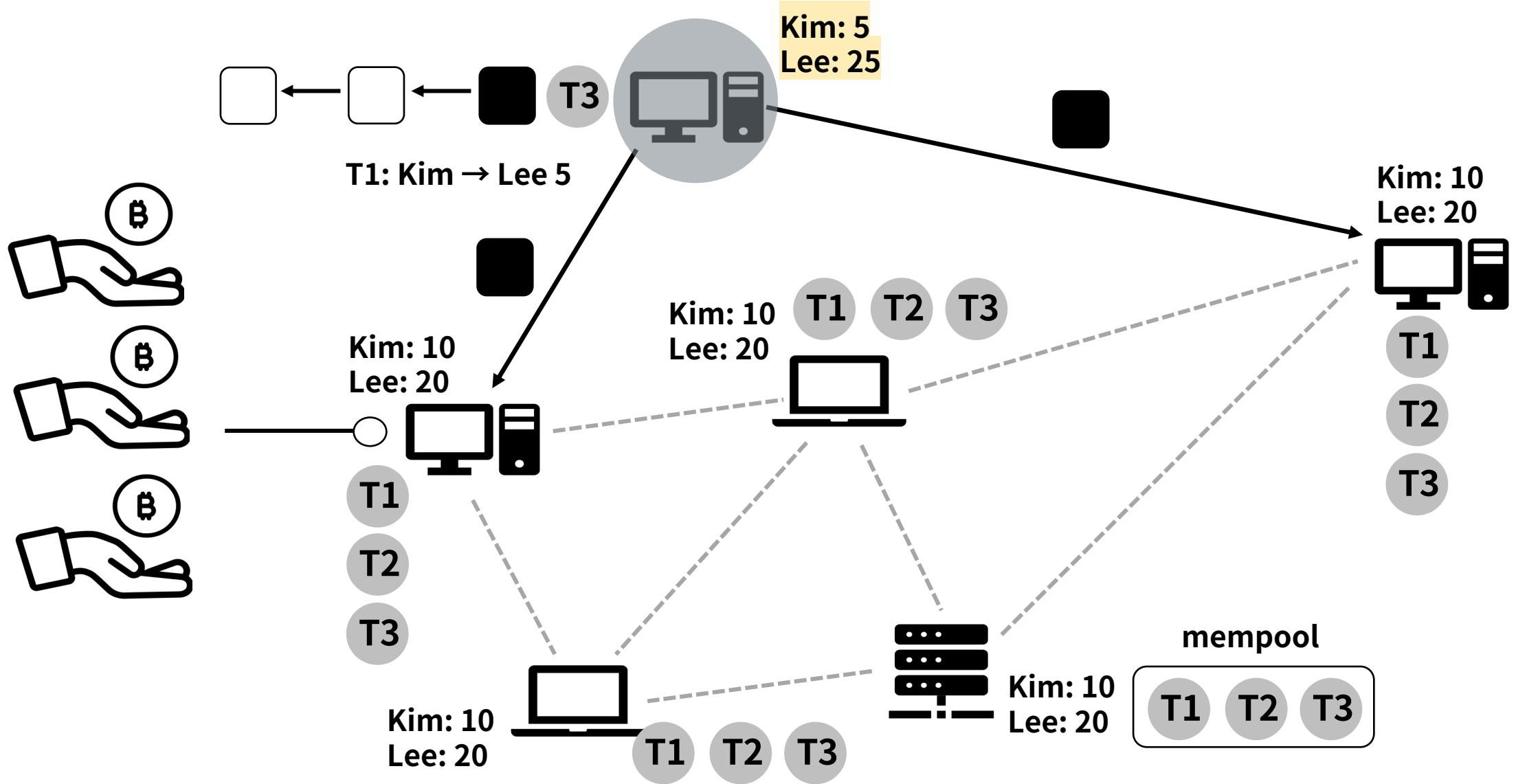
# Life cycle of transaction



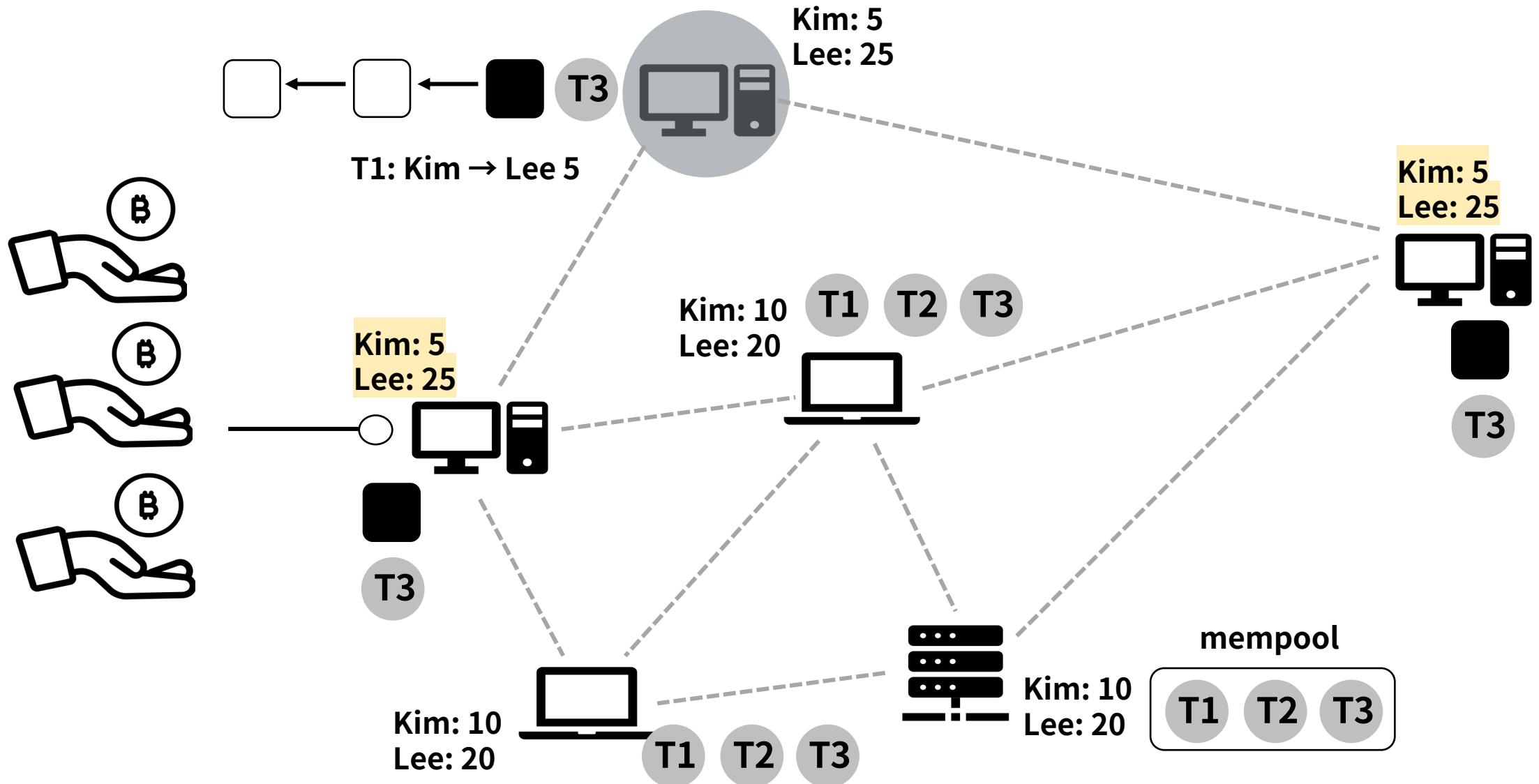
# Life cycle of transaction



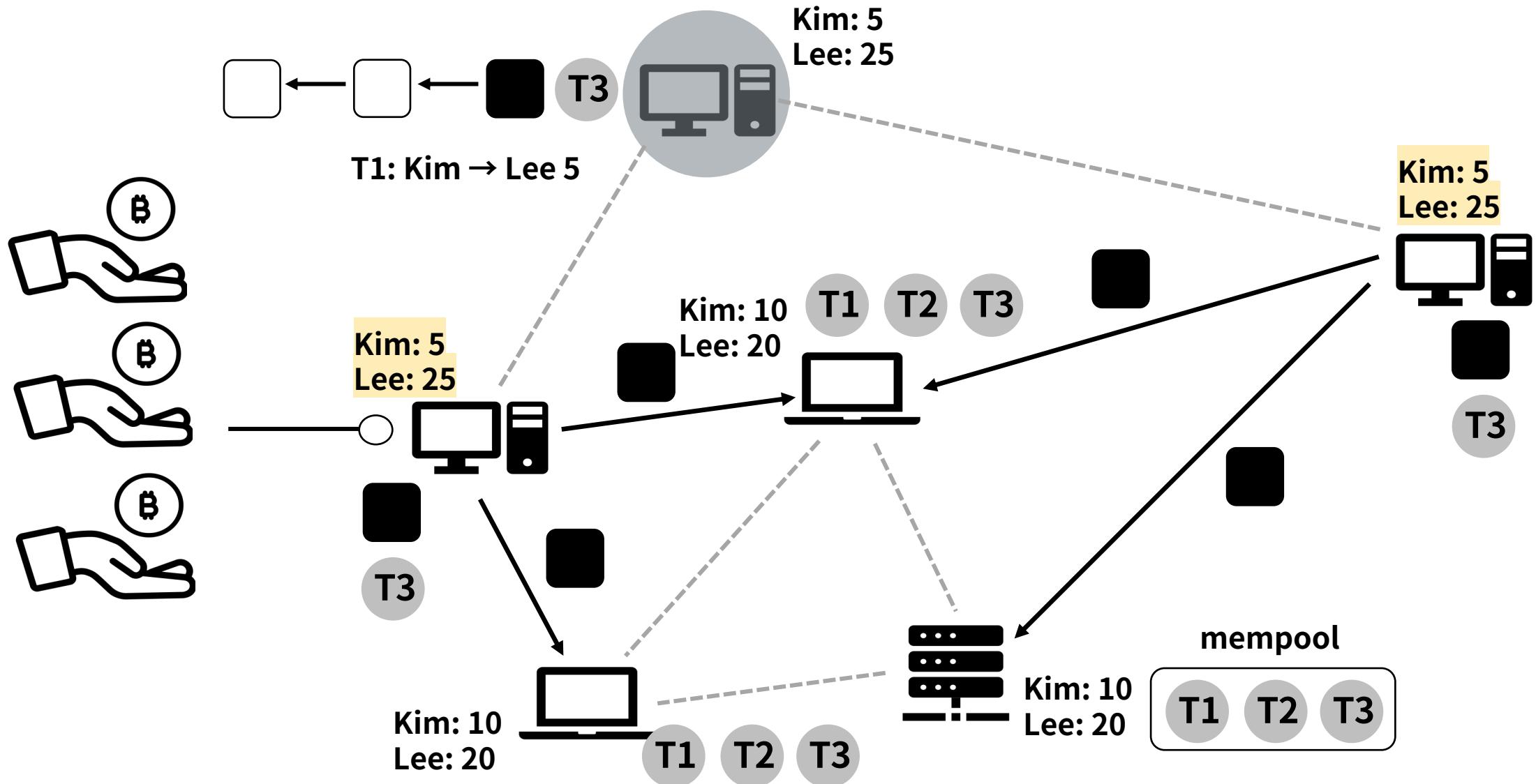
# Life cycle of transaction



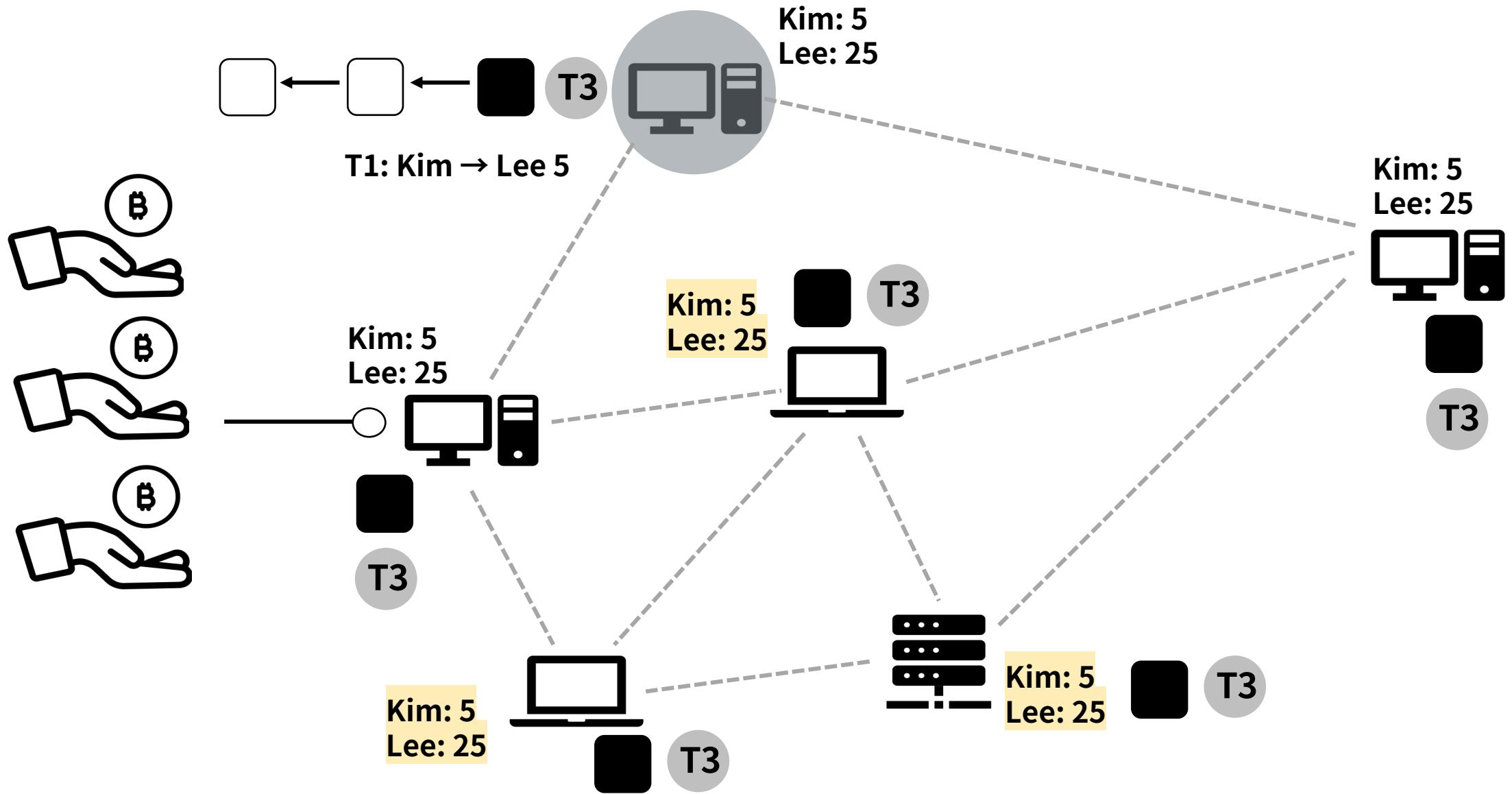
# Life cycle of transaction



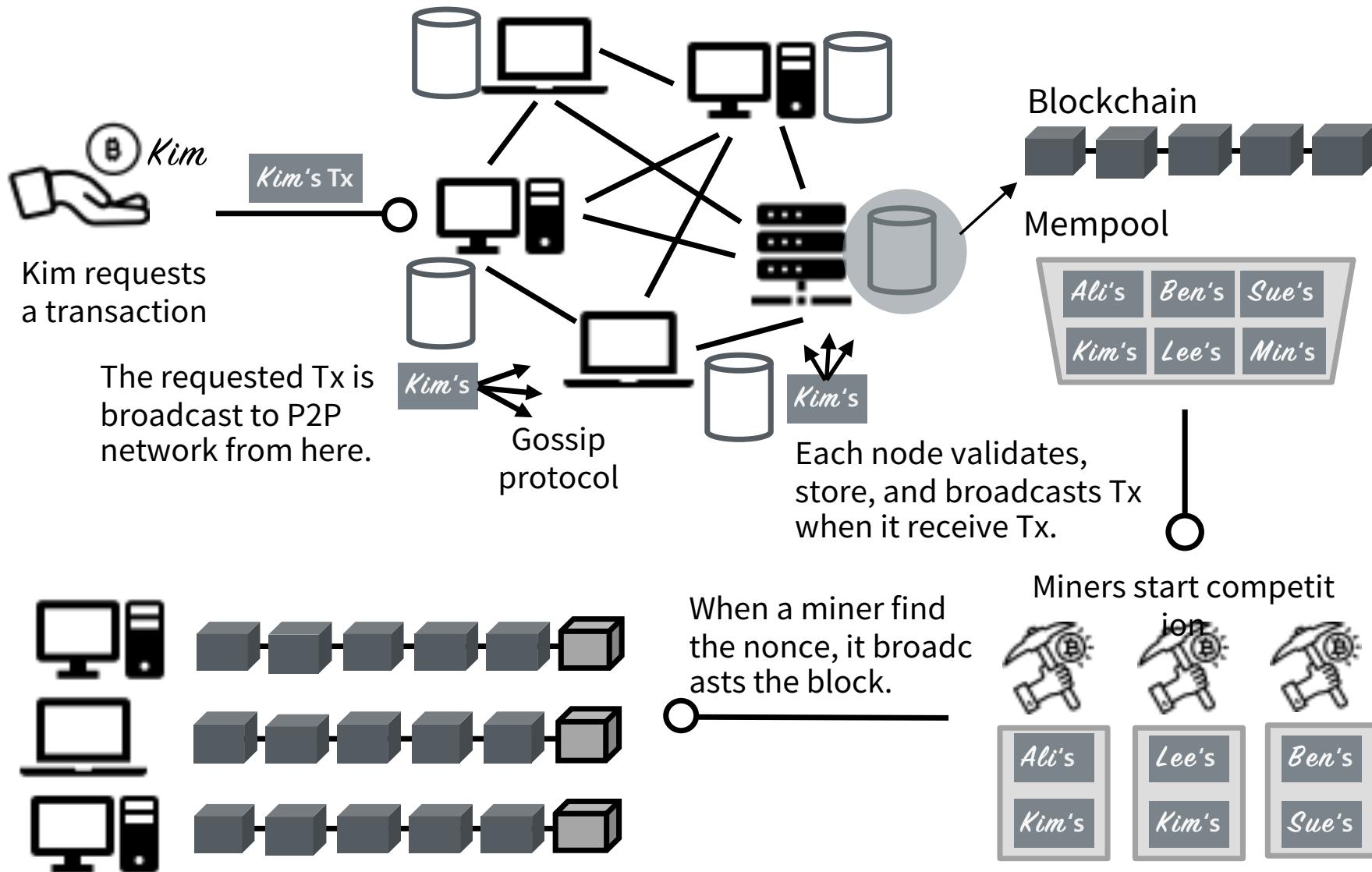
# Life cycle of transaction



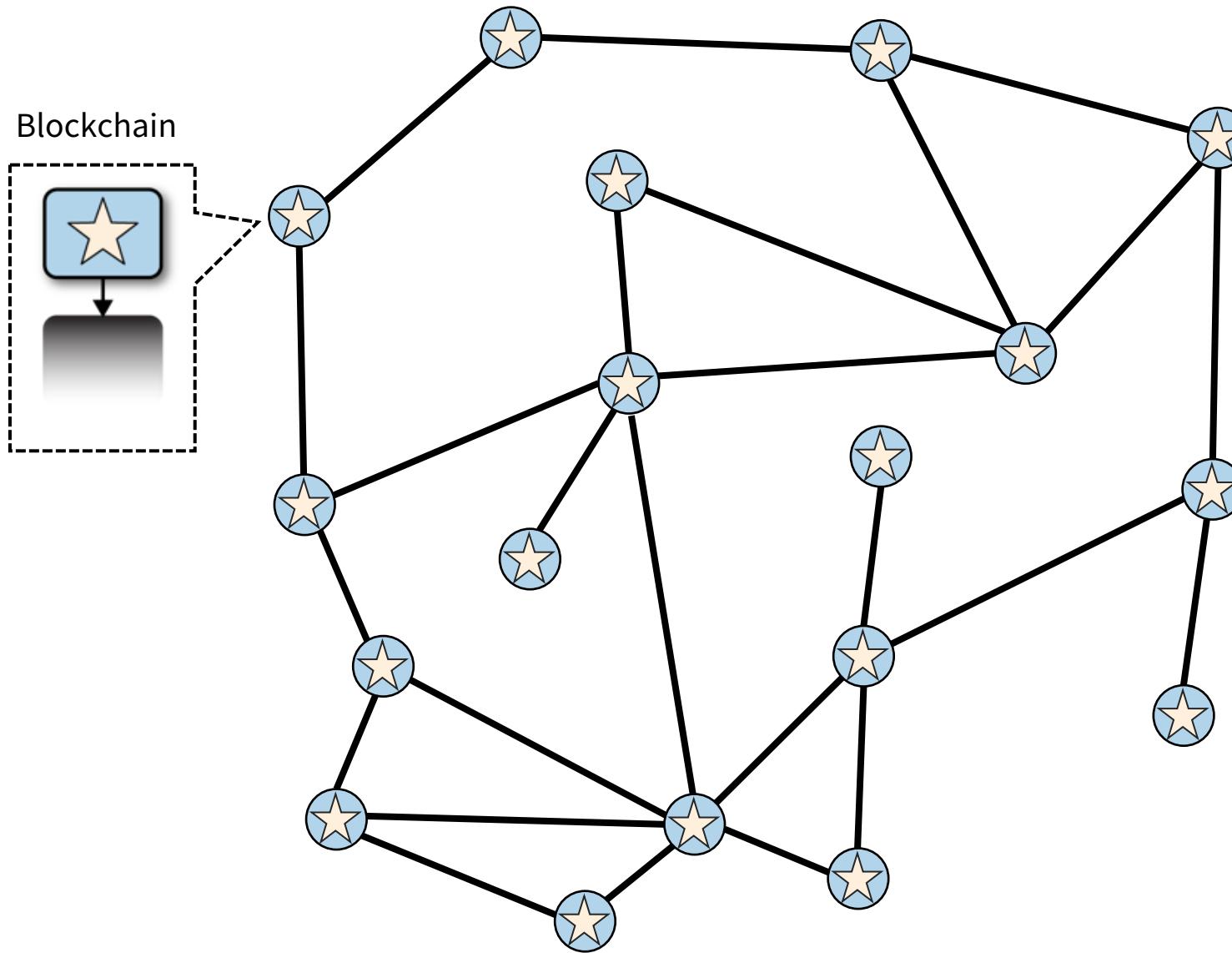
# Life cycle of transaction



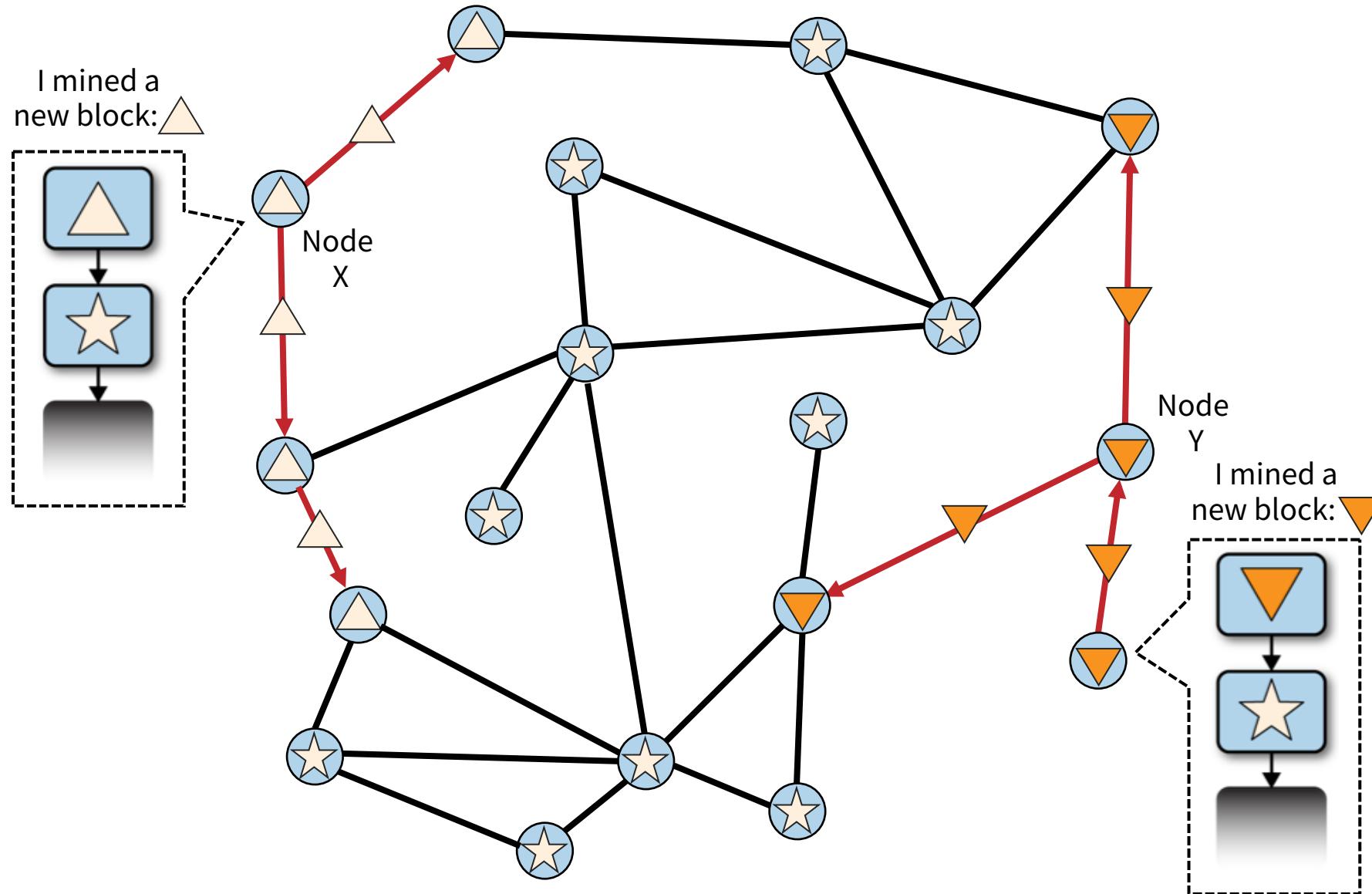
# 블록체인 충돌 해소 과정



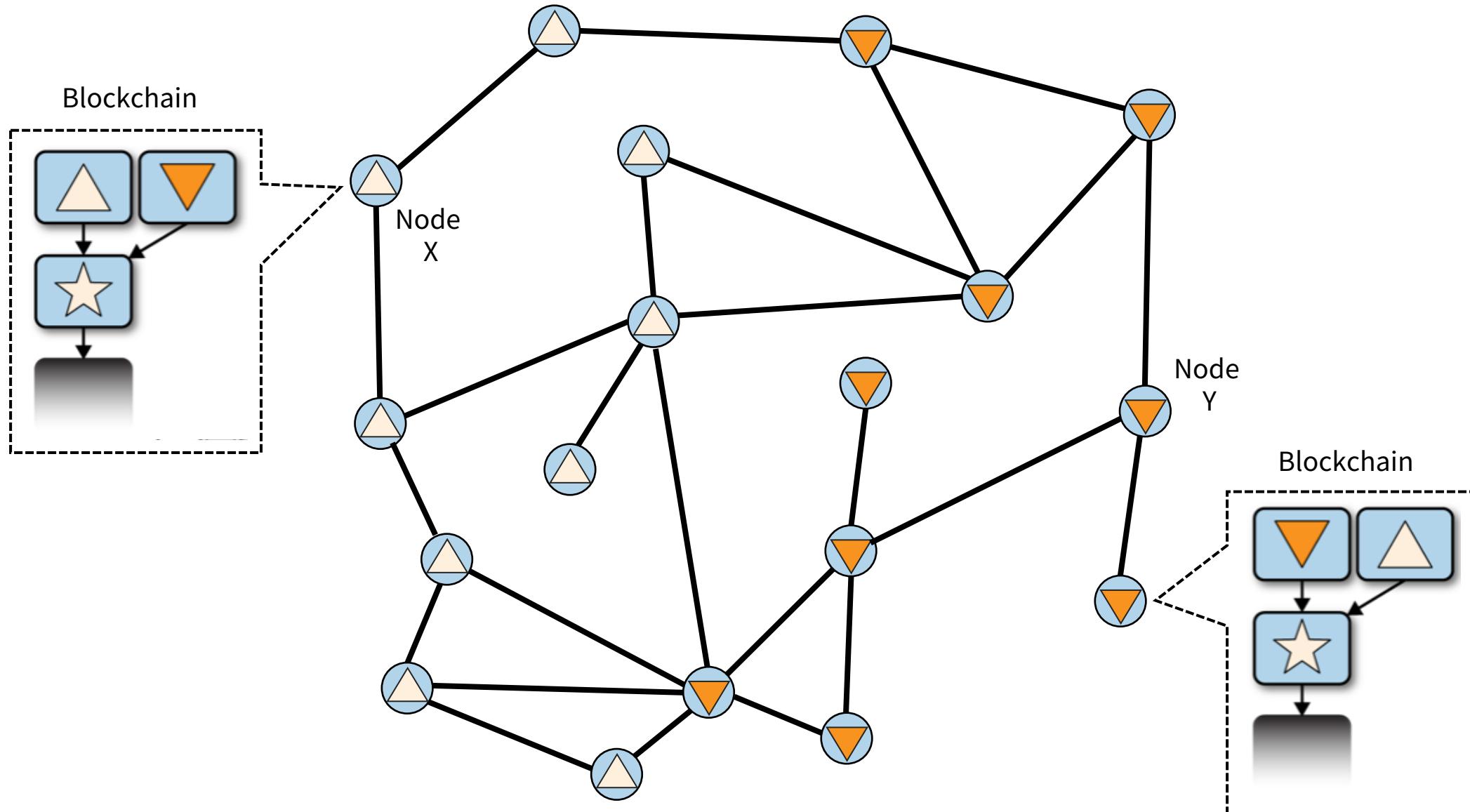
# 블록체인 충돌 해소 과정 (Mastering Bitcoin)



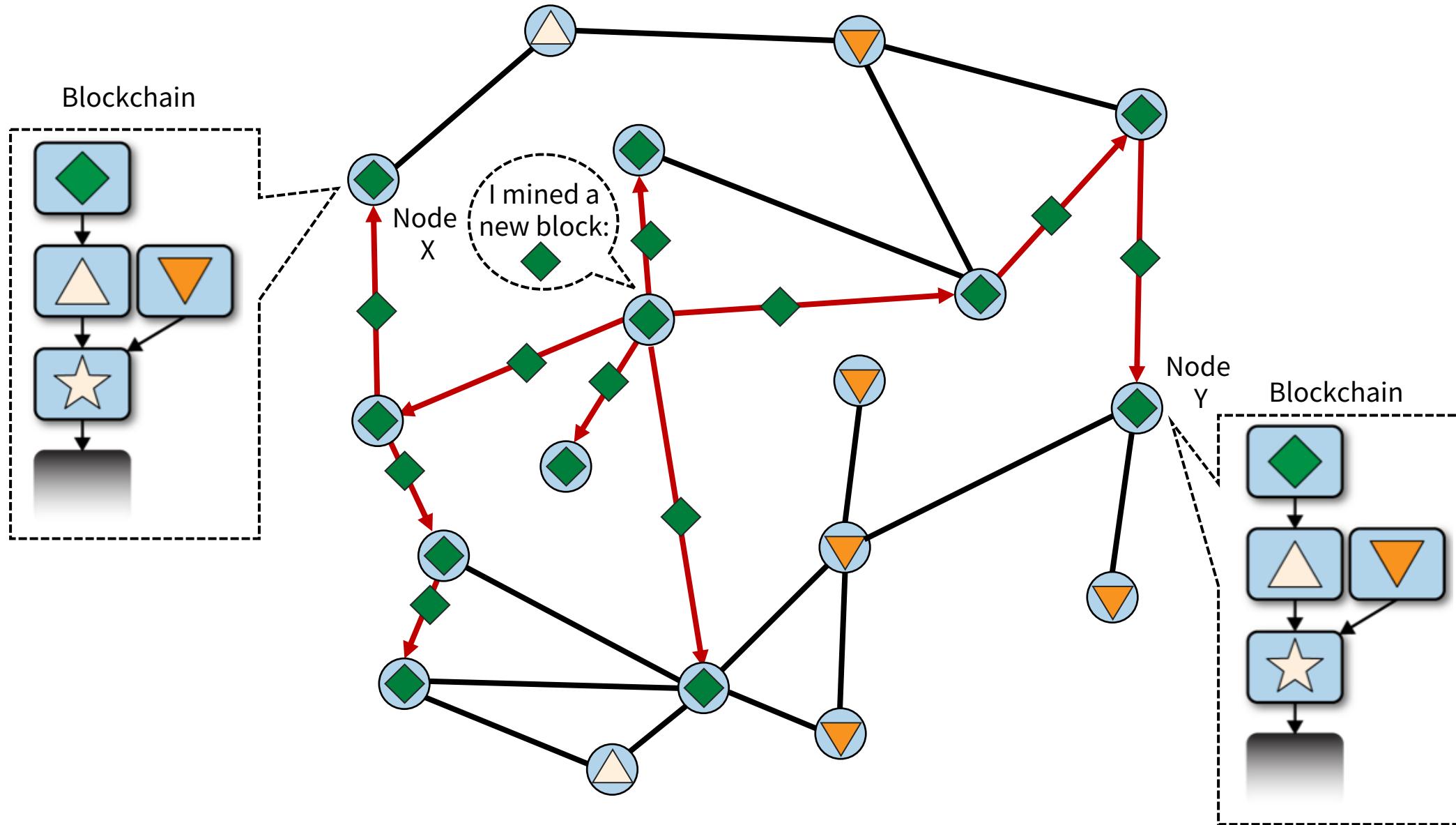
# 블록체인 충돌 해소 과정 (Mastering Bitcoin)



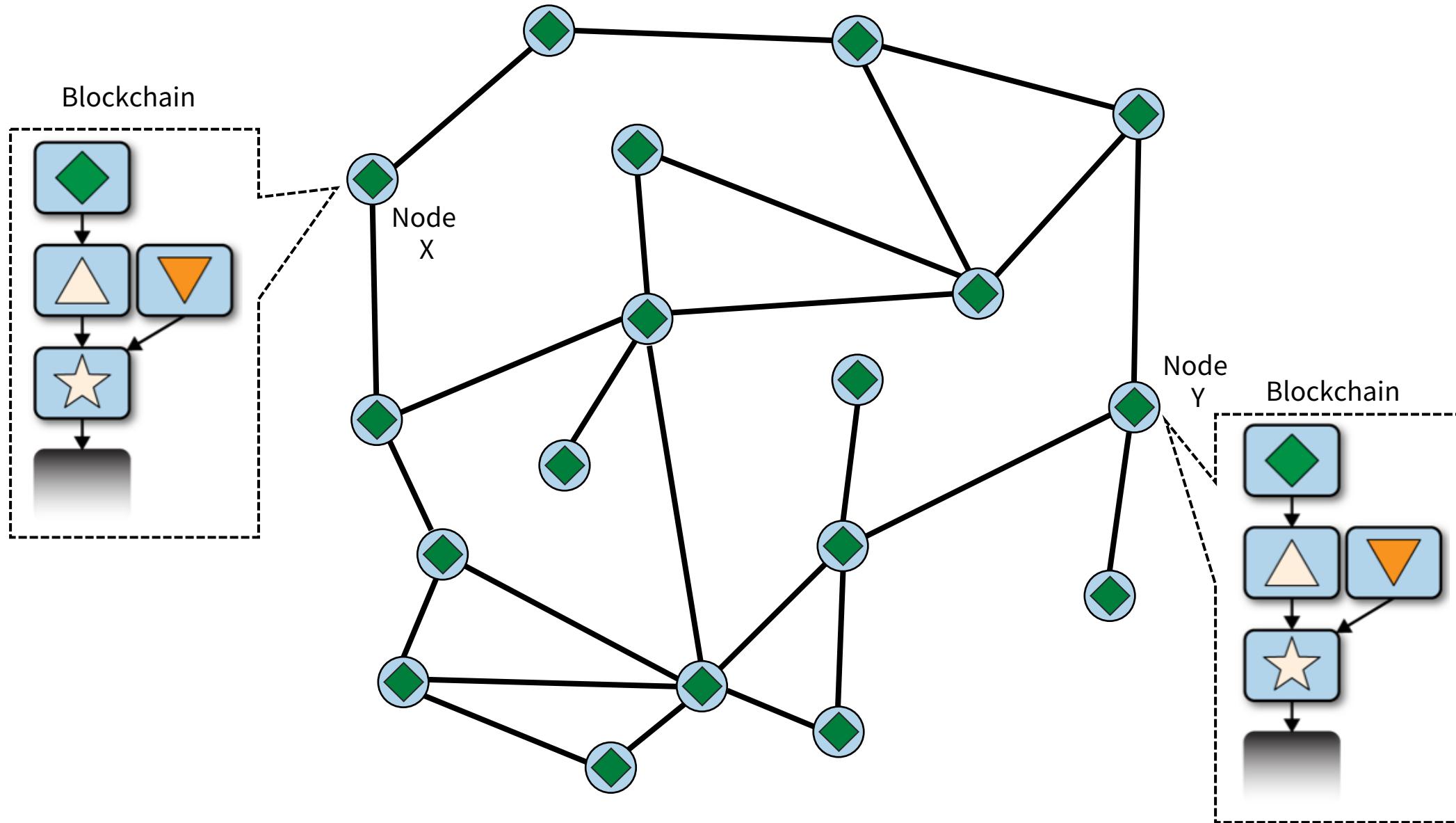
# 블록체인 충돌 해소 과정 (Mastering Bitcoin)



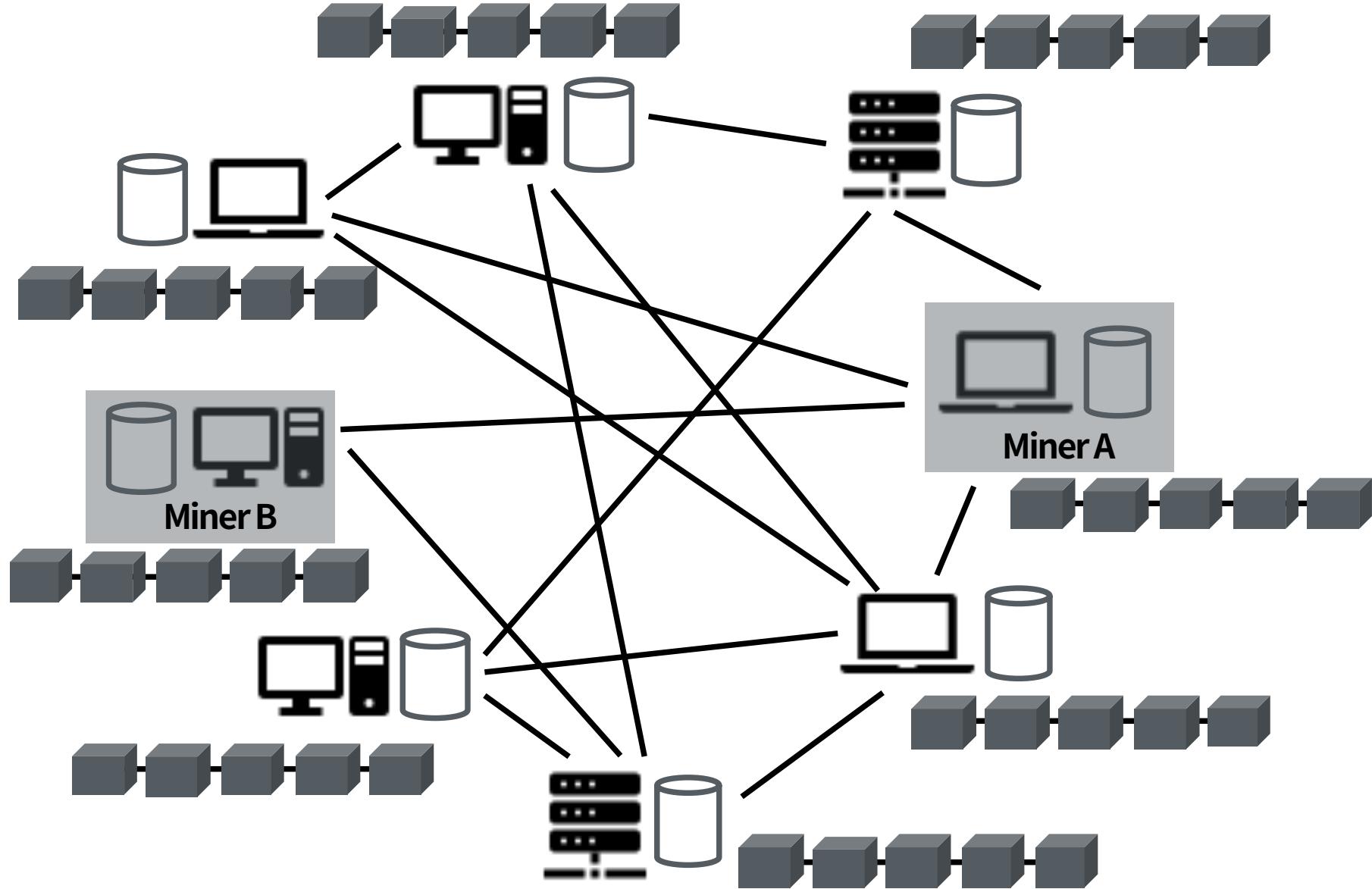
# 블록체인 충돌 해소 과정 (Mastering Bitcoin)



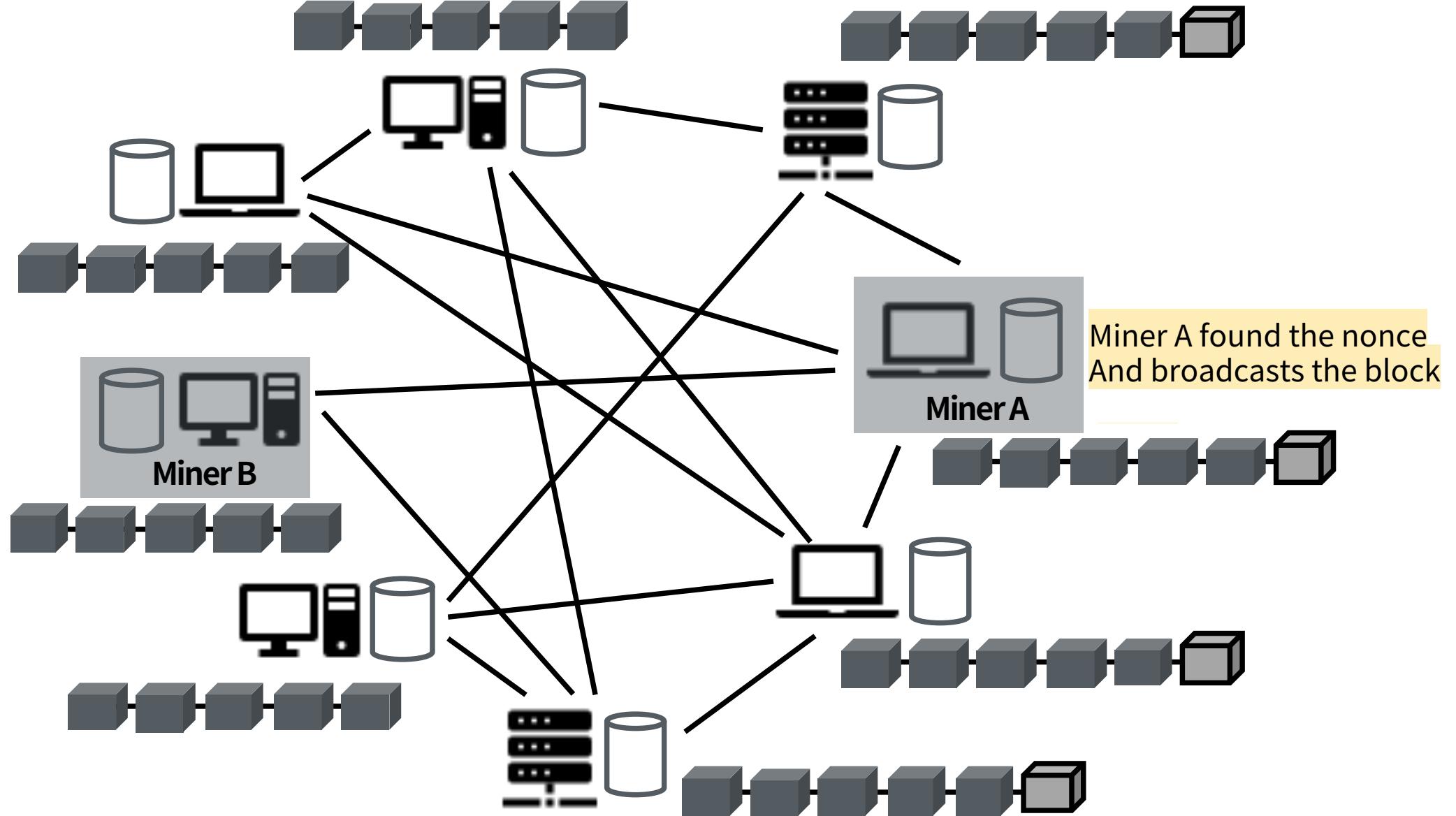
# 블록체인 충돌 해소 과정 (Mastering Bitcoin)



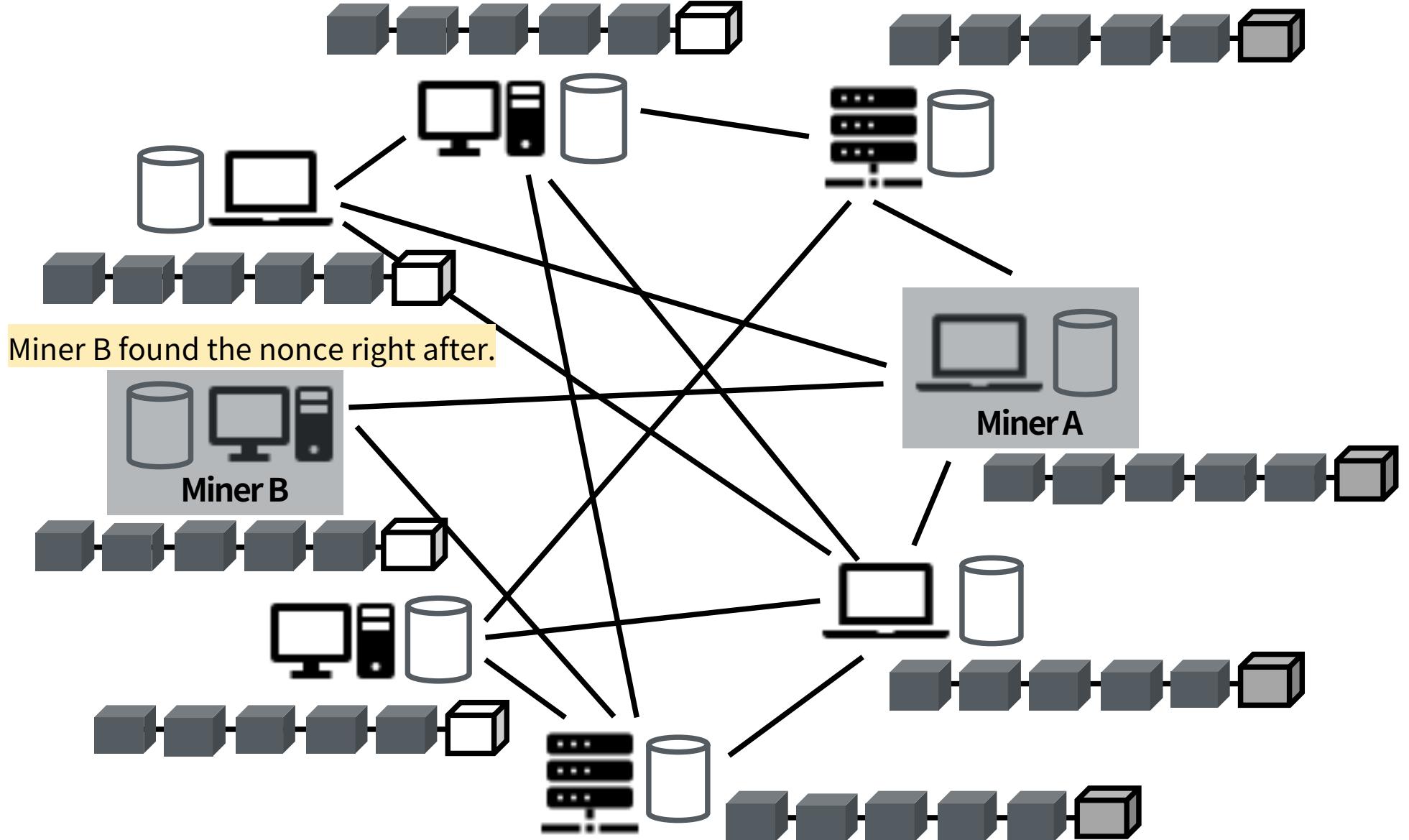
# 블록체인 충돌 해소 과정



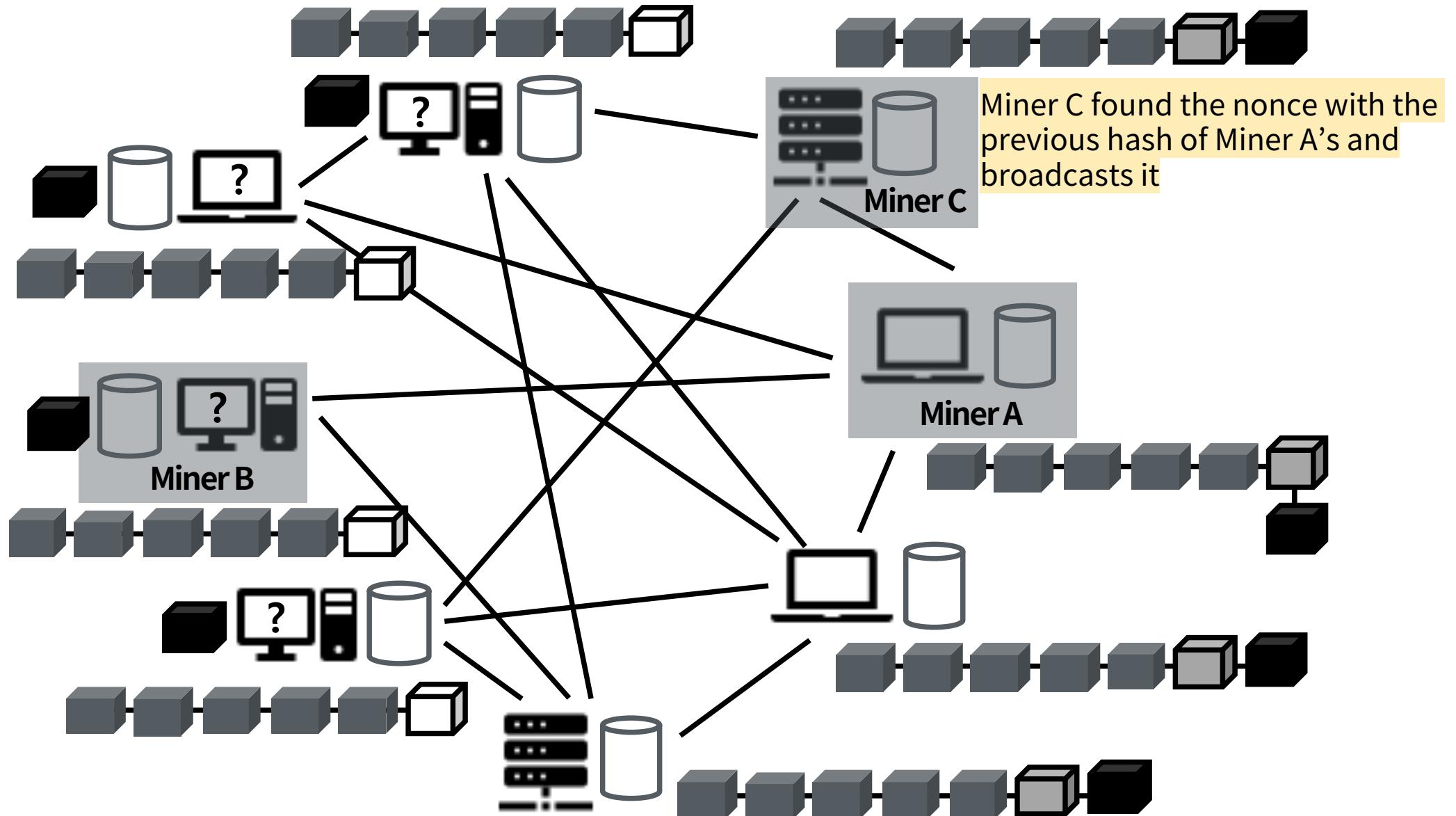
## 블록체인 충돌 해소 과정



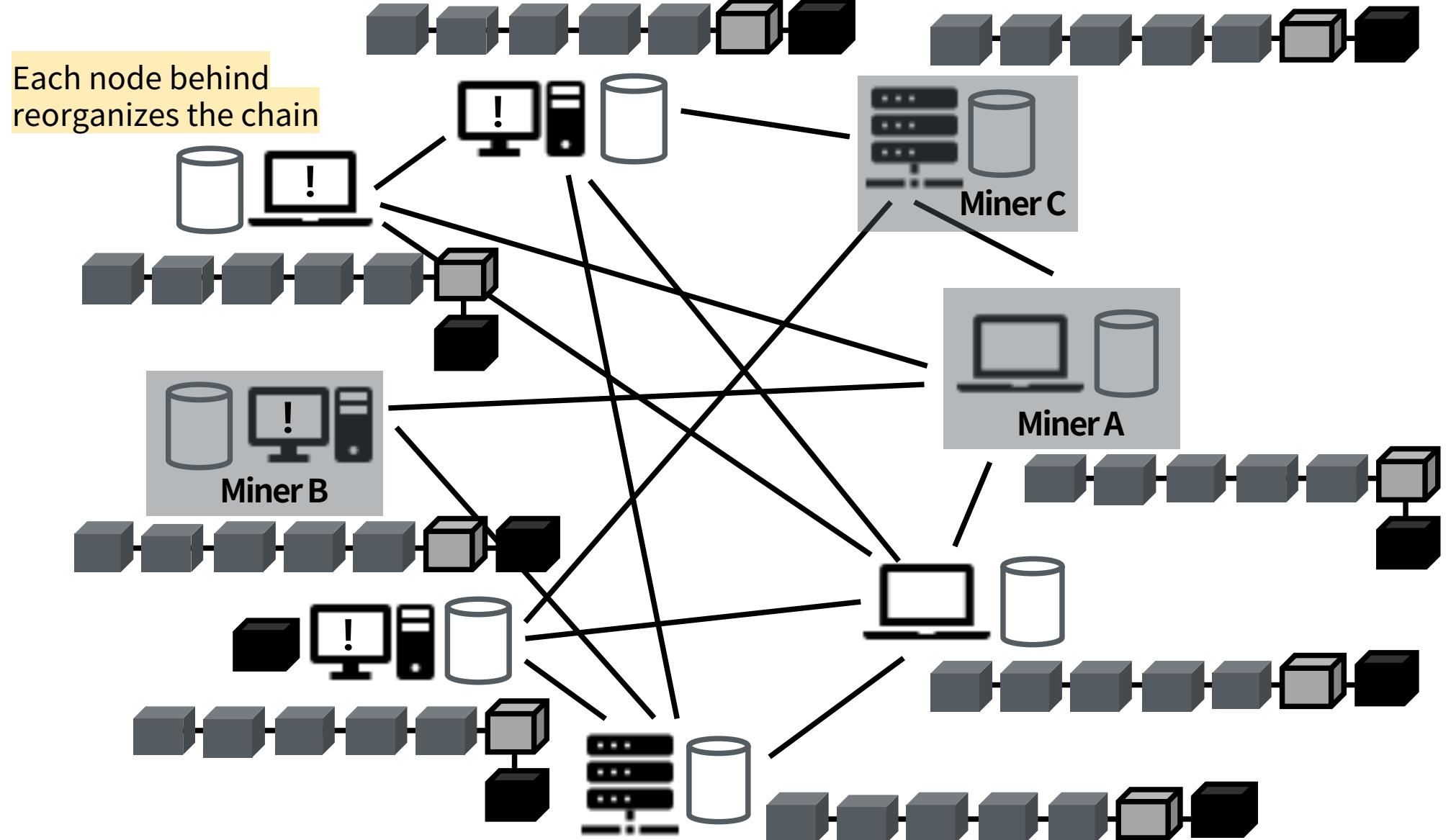
# 블록체인 충돌 해소 과정



# 블록체인 충돌 해소 과정



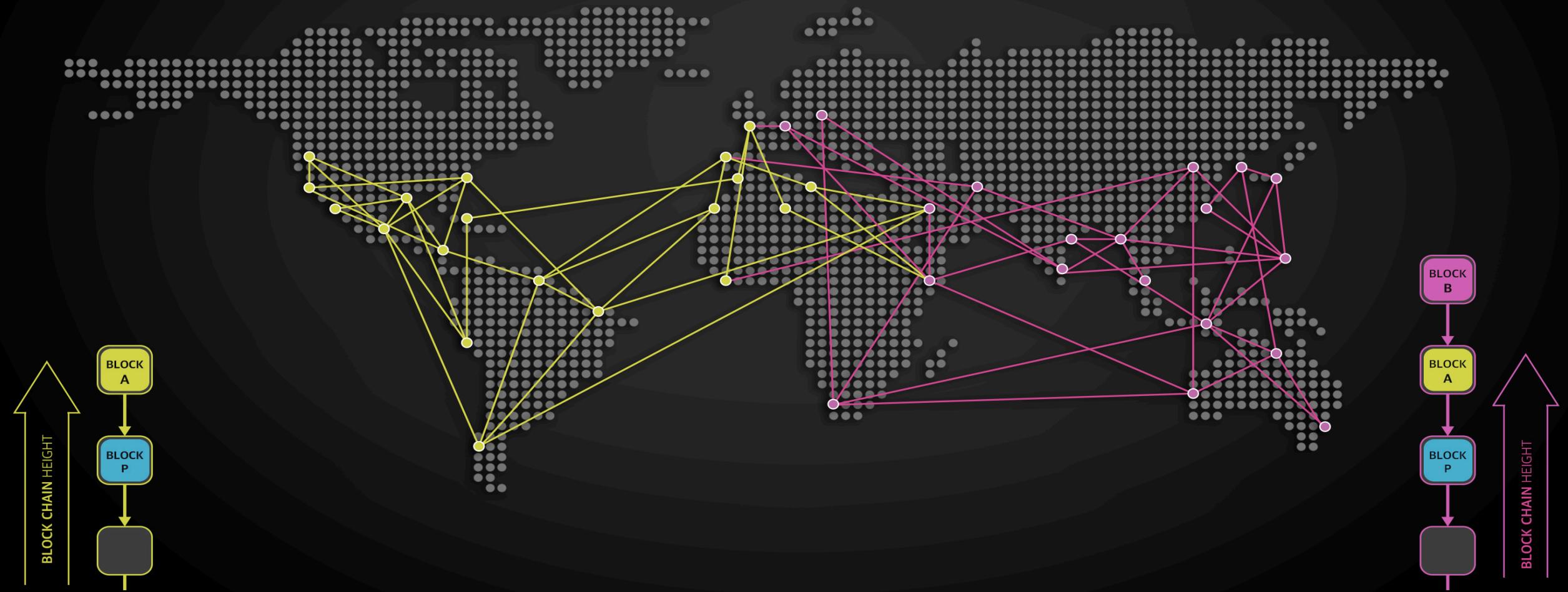
# 블록체인 충돌 해소 과정



# 블록체인 충돌 해소 과정



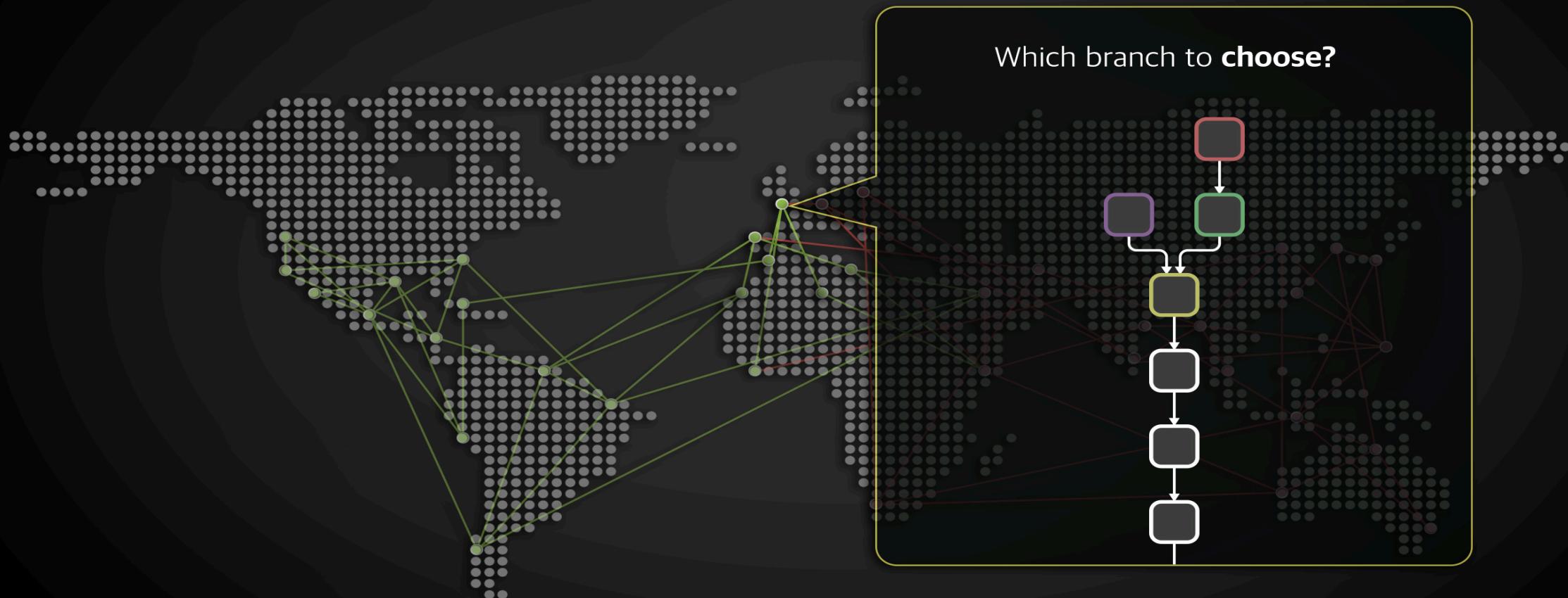
# 블록체인 충돌 해소 과정



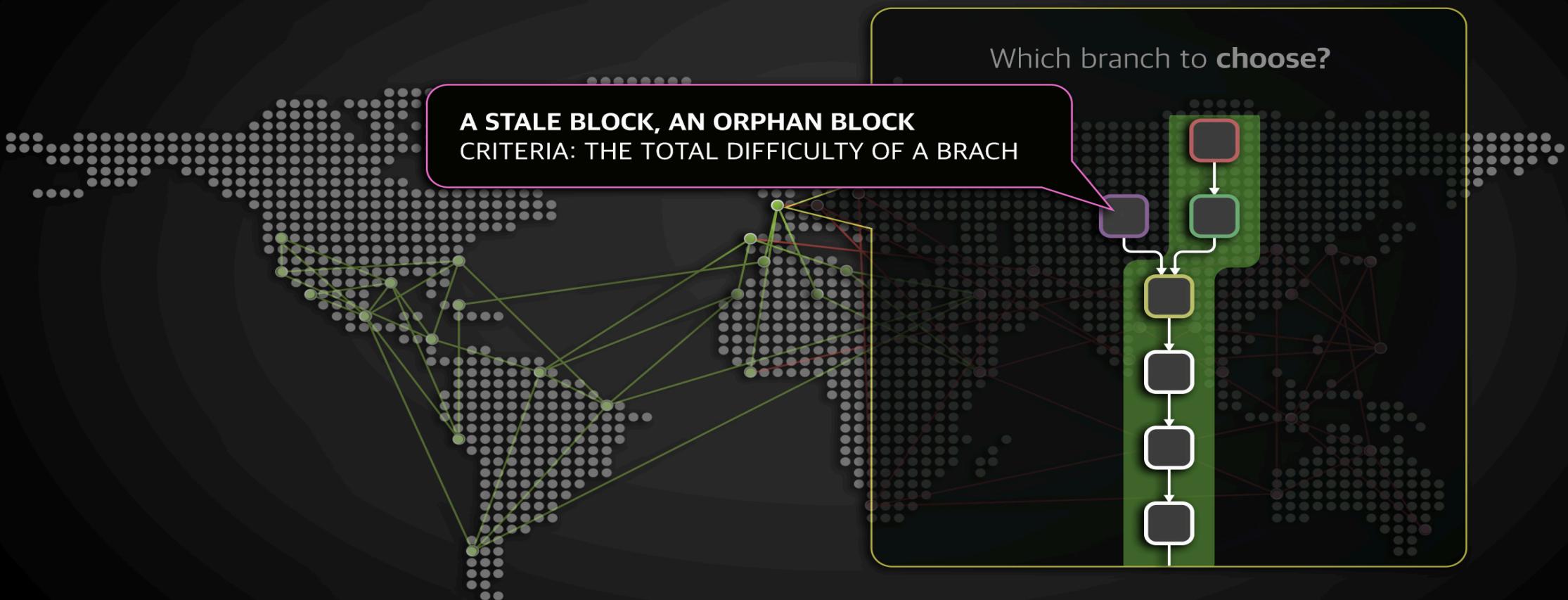
# 블록체인 충돌 해소 과정



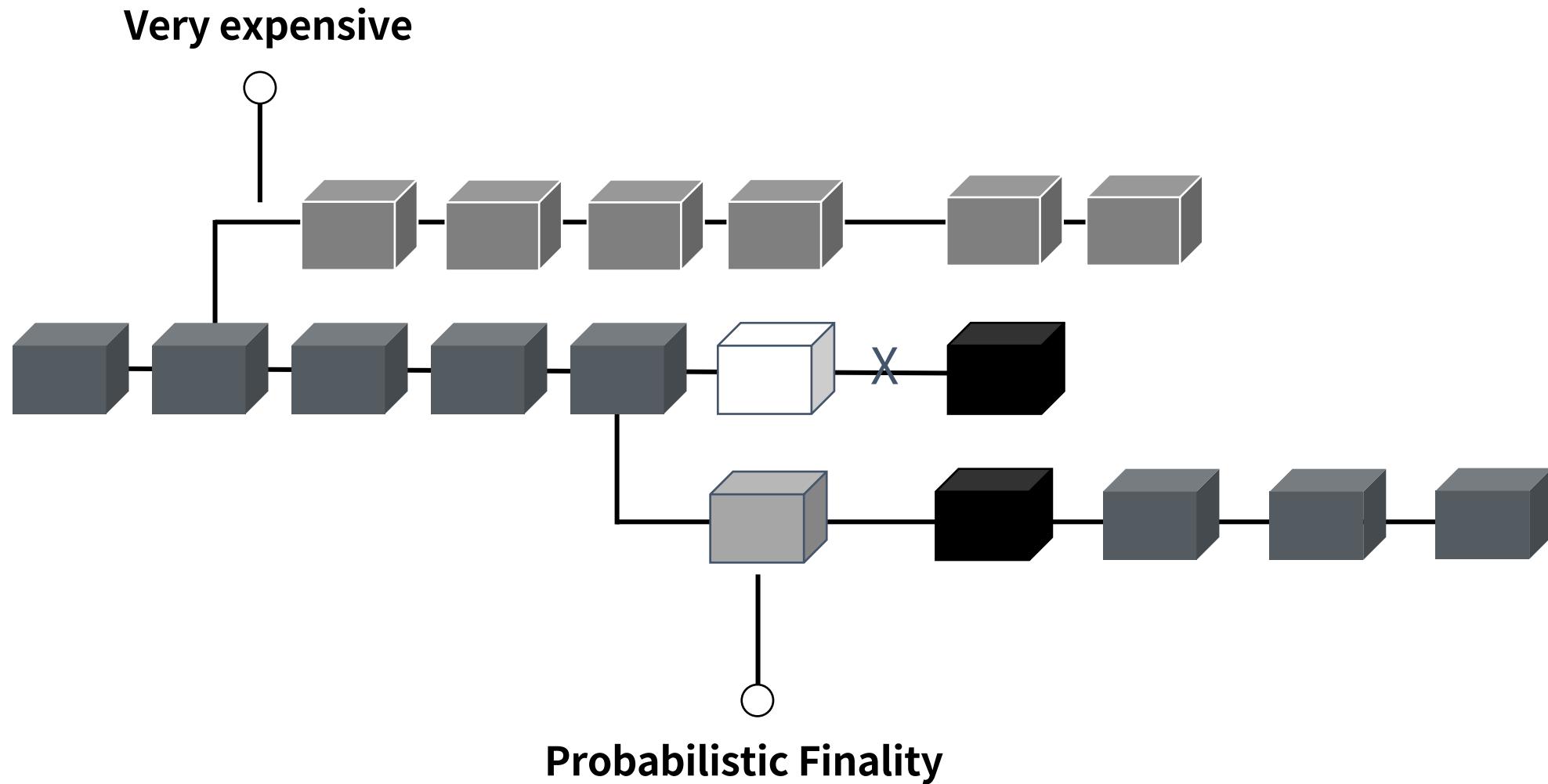
# 블록체인 충돌 해소 과정



# 블록체인 충돌 해소 과정



# 블록체인 충돌 해소 과정



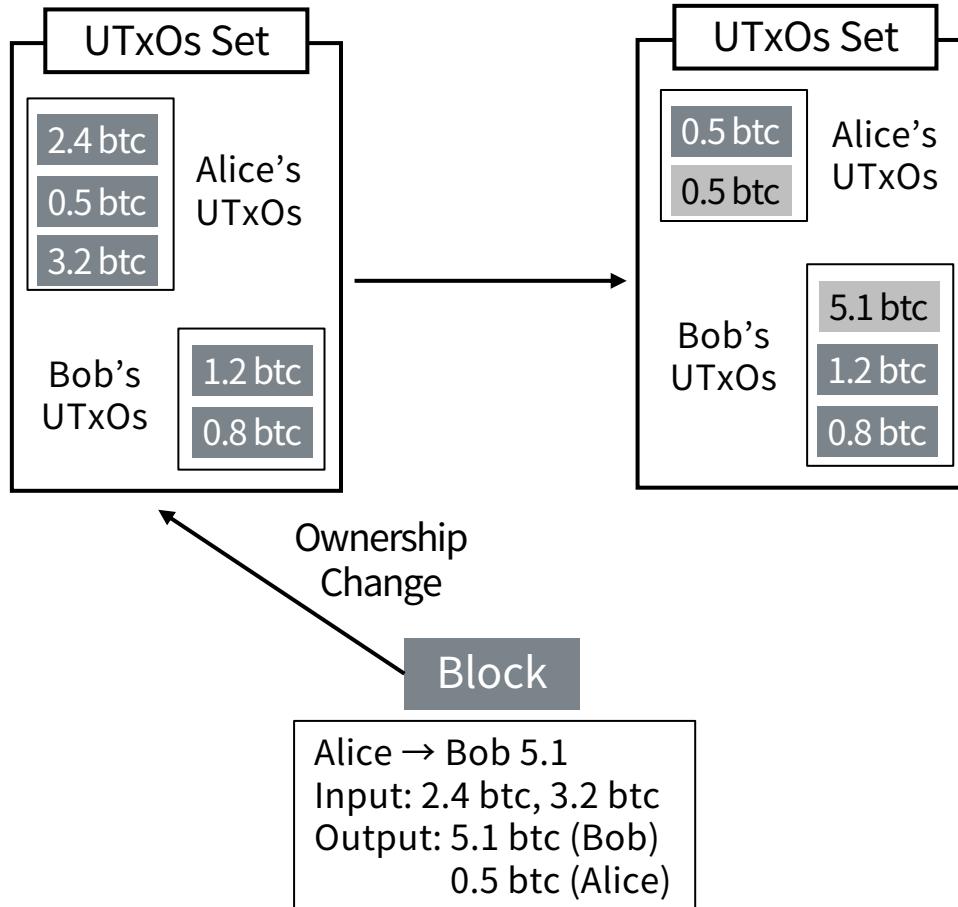
# RULES

---

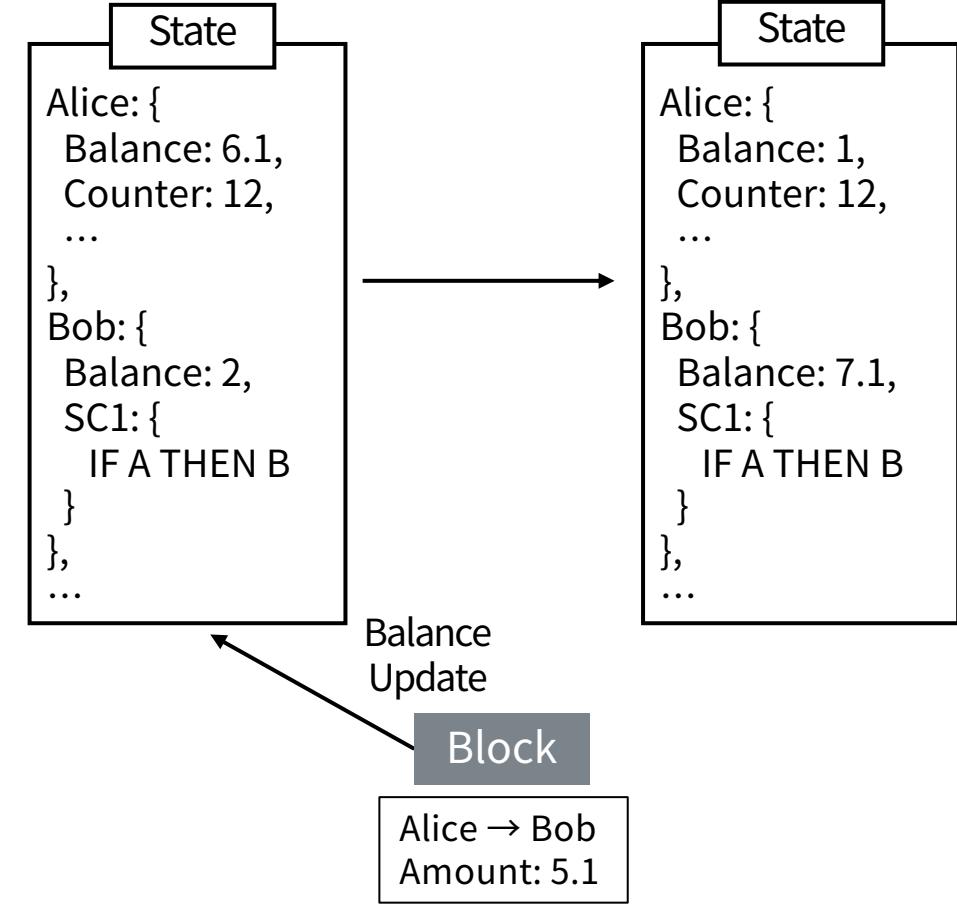
1. 비트코인을 송금하는 사람은 거래에 자신의 디지털 서명을 포함시켜야 한다.
2. 각 블록은 이전 블록의 해시, 오퍼레이션과 논스로 구성된다.
3. 제일 먼저 블록의 해시를 난이도보다 낮게 만든 채굴자가 보상을 받는다.
4. 채굴자는 블록에 보상(코인베이스 트랜잭션)을 포함시킨다.
5. 새 블록의 해시는 *Hash*(이전 블록의 해시 + 오퍼레이션 + 논스)이며, 논스를 바꿔가며 목표 해시를 찾는다.
6. 누적 난이도 총합이 가장 큰 체인이 적법한 체인이다.

# **Abstract Blockchain**

# UTxO vs. Account-based Model



The current **sum of UTxOs** of Alice?



The current **value of balance** in Alice's account?

# Block as an Operator

이전 상태

Ledger	
Alice	2btc
Bob	1btc
Kim	3btc
Lee	4btc

새로운 거래 기록

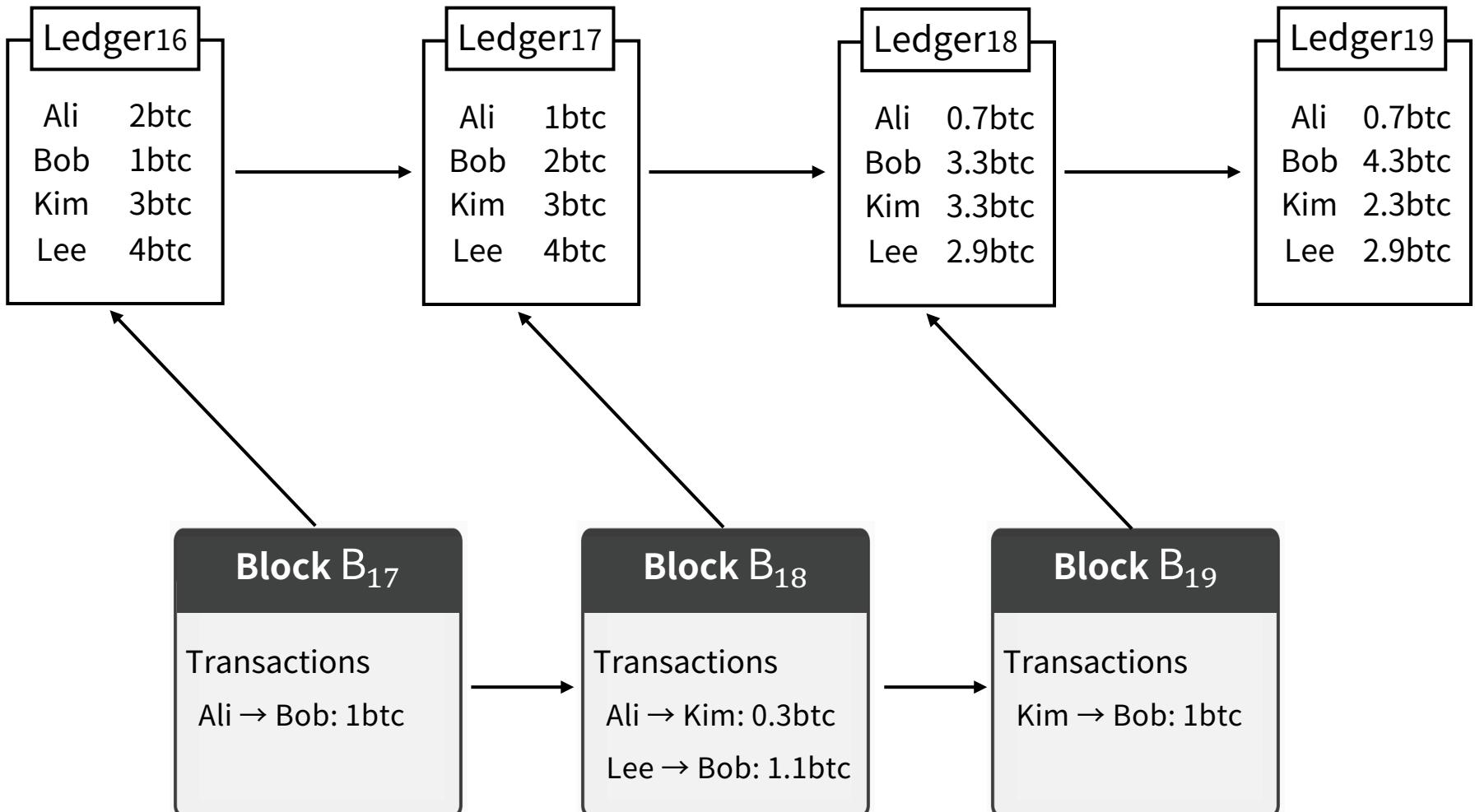
Block B <sub>17</sub>	
Transactions	
Alice → Bob: 1btc	
Kim → Lee: 2btc	

새로운 상태

Ledger	
Alice	1btc
Bob	2btc
Kim	1btc
Lee	6btc

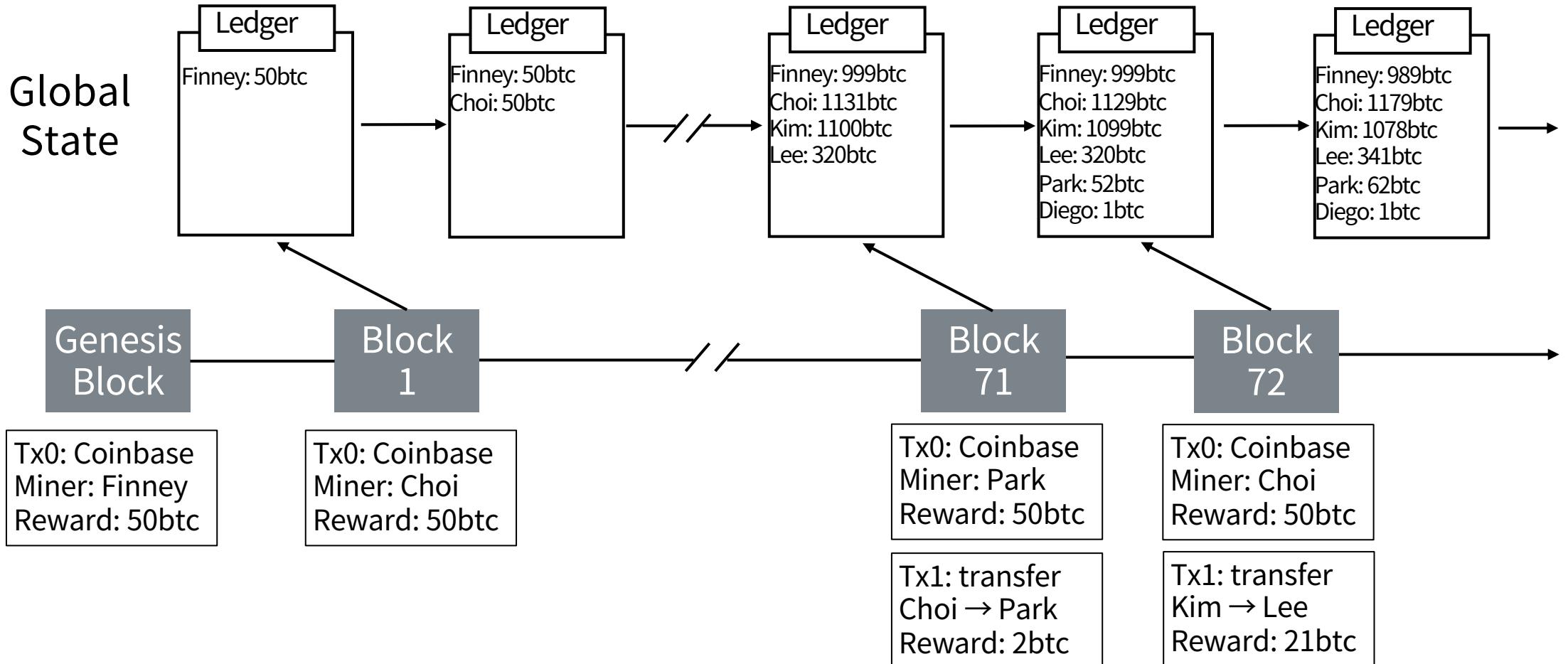
# Blockchain = A chain of blocks

거래의 결과  
(상태 값)

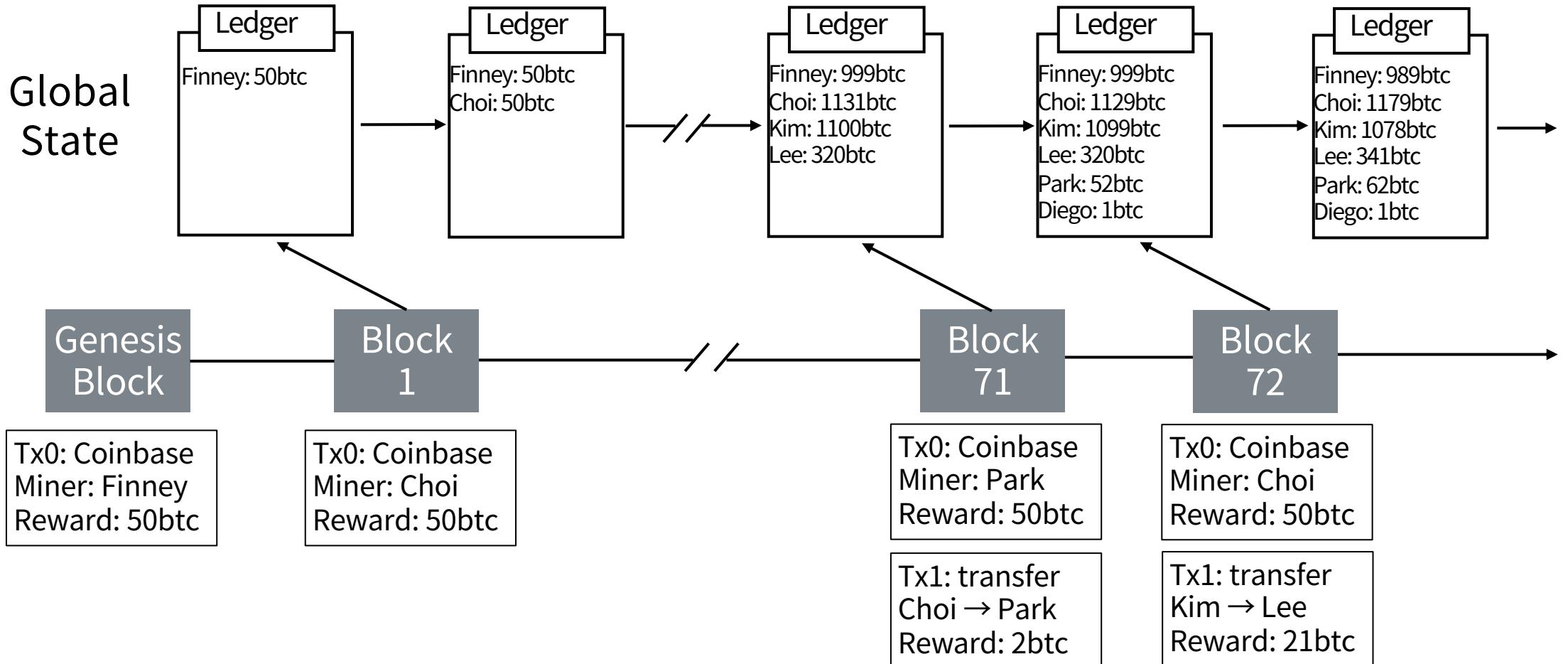


거래 기록들  
(블록체인)

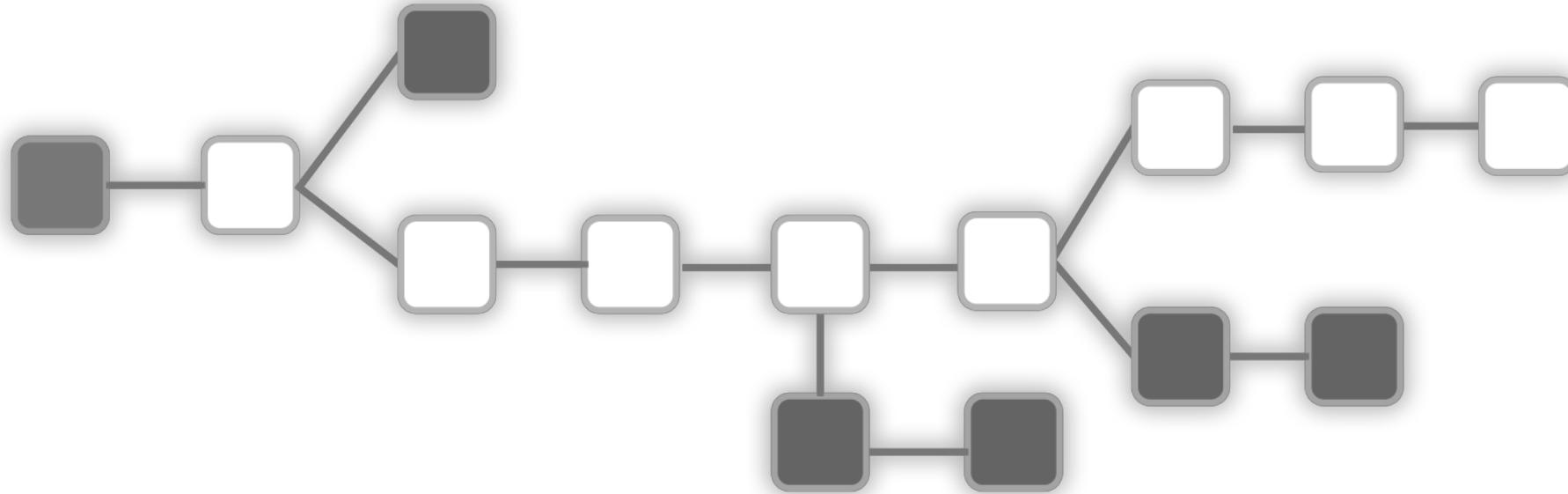
# Block as an Operator



# Block as an Operator



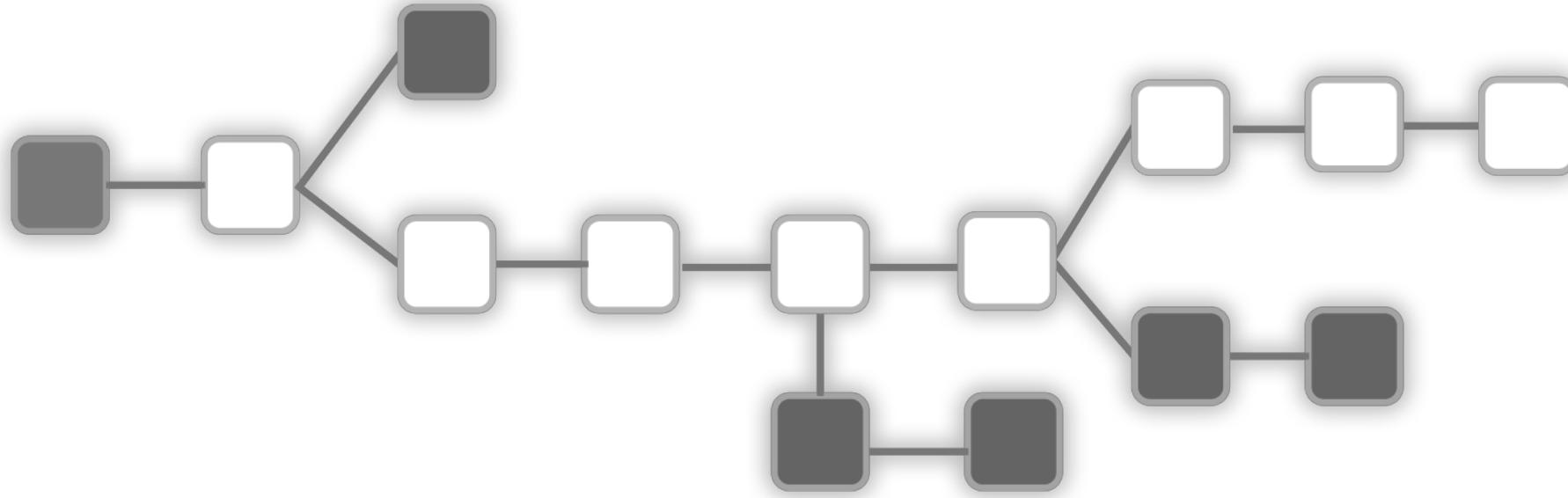
# Abstract Blockchain



## A blockchain

represents **a single state** being concurrently edited and  
keeps the state as **a shared ledger** to avoid conflicts  
as a series of **transformations** applied to an initial state (genesis state).  
**A set of blocks**, each **bundling** together multiple operations in it.

# Abstract Blockchain

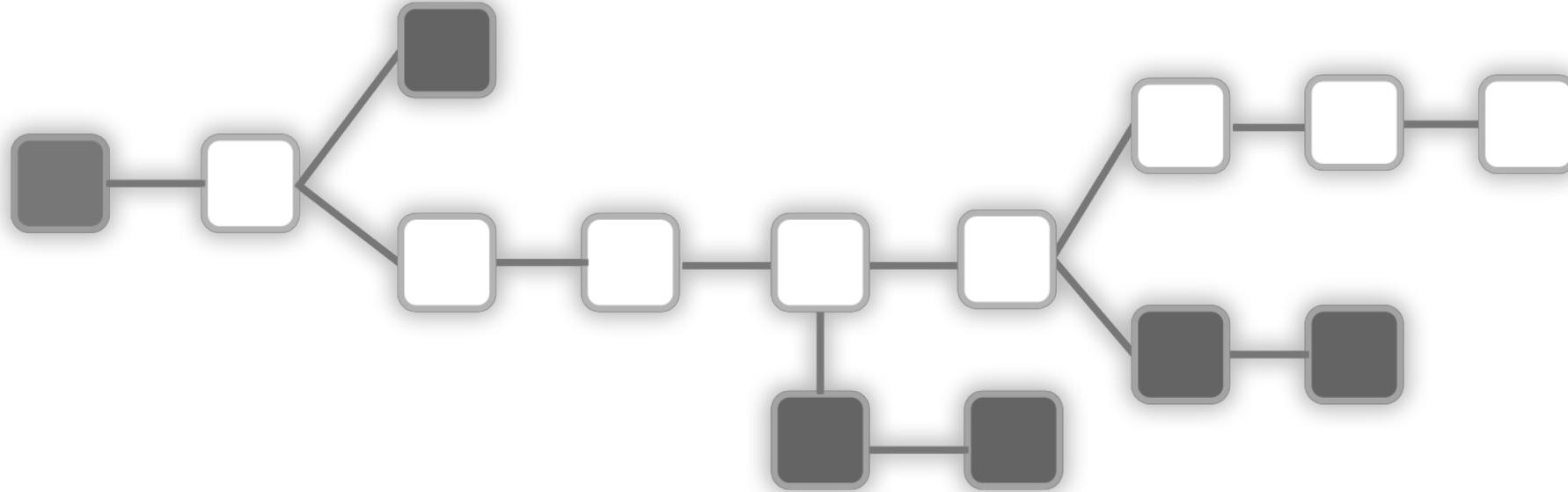


## A blockchain protocol

is a monadic implementation of **concurrent mutations** of a **global state** through a series of operations such that blocks are defined as operators.

**apply**:  $S \mapsto O \mapsto S$  or  $S' = f(S, O)$

# Abstract Blockchain



## Blocks

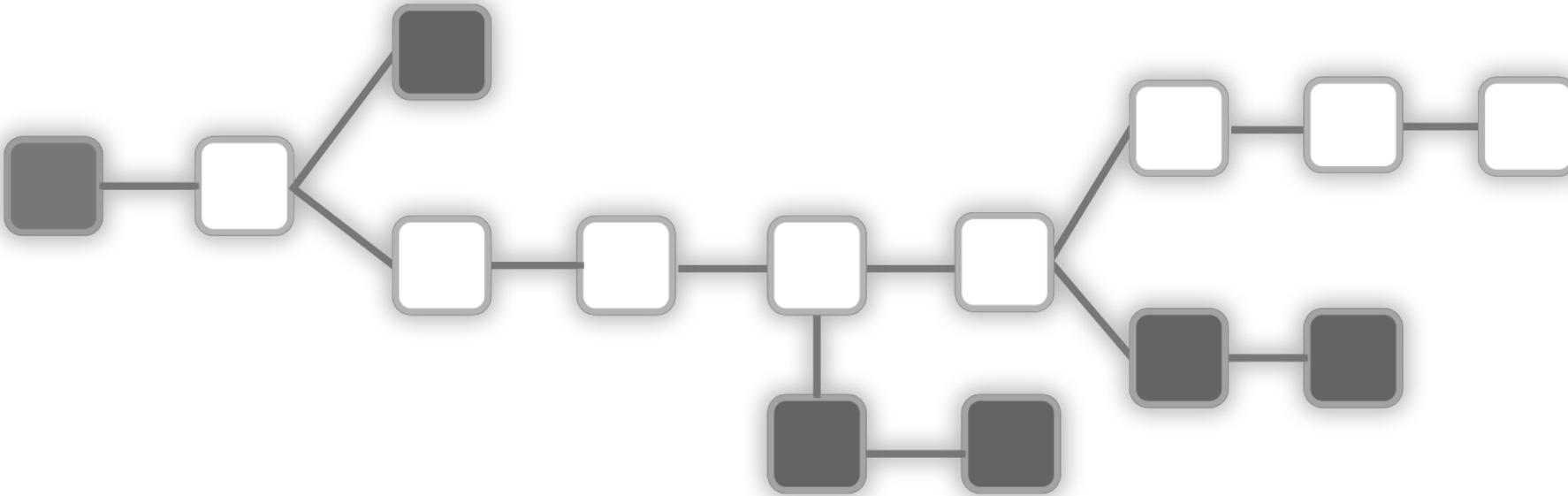
are created asynchronously by many concurrent nodes

forms a **tree structure**,

which needs **well-ordering** for choosing **the valid one** (unique, canonical).

**score**:  $S \mapsto N$  or  $N = f(S)$

# Abstract Blockchain

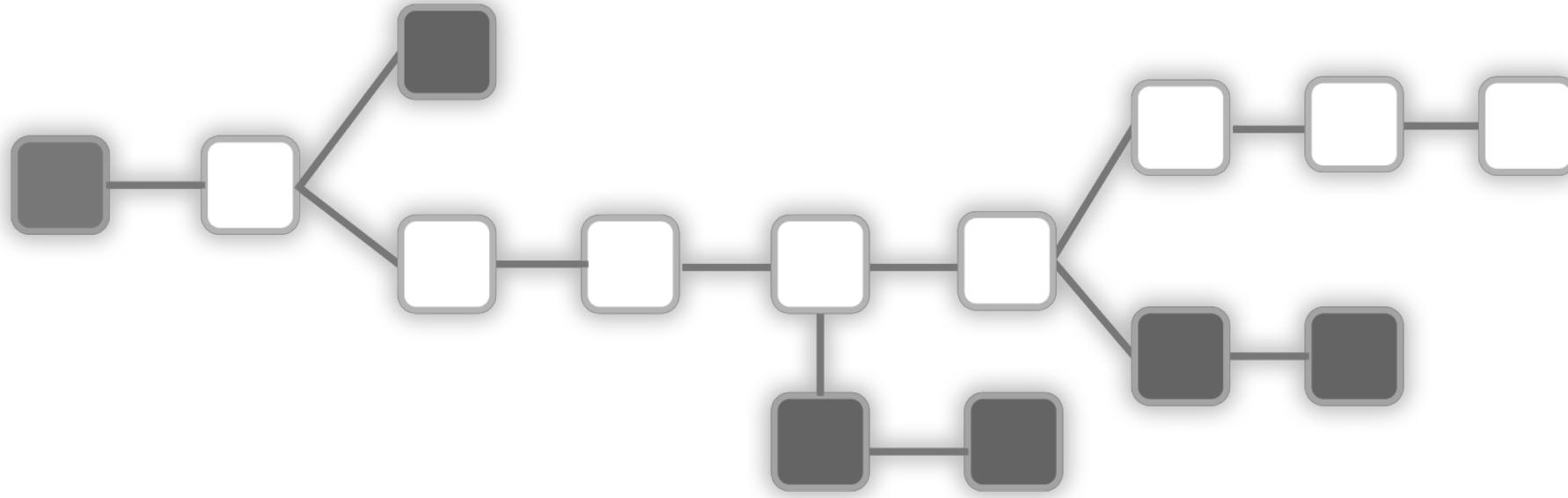


All existing block chain implementation generally

**apply**:  $S \mapsto O \mapsto S$  or  $S' = f(S, O)$

**score**:  $S \mapsto N$  or  $N = f(S)$

# Abstract Blockchain



## Bitcoin

State = A set of UTxOs + total work + block index

## Operations = transactions

Score = the one with the greatest total difficulty

# **테조스 플랫폼**

# Tezos , **the last** crypto currency



## Tezos: A Self-Amending Crypto-Ledger Position Paper (2014)

### Conclusion

We've presented issues with the existing cryptocurrencies and offered Tezos as a solution. While the irony of preventing the fragmentation of cryptocurrencies by releasing a new one does not escape us, *Tezos truly aims to be the *last* cryptocurrency.*

No matter what innovations other protocols produce, it will be possible for Tezos stakeholders to adopt these innovations. Furthermore, the ability to solve collective action problems and easily implement protocols in OCaml will make Tezos one of the most reactive cryptocurrency.

# Tezos , the last crypto currency

## Problems

### The Protocol Fork Problem

### Shortcomings of Proof-of-Work

### Smart Contracts

### Correctness

## Contents

<b>1</b>	<b>Motivation</b>	<b>2</b>
1.1	The Protocol Fork Problem . . . . .	3
1.1.1	Keeping Up With Innovation . . . . .	3
1.1.2	Economics of Forks . . . . .	4
1.2	Shortcomings of Proof-of-Work . . . . .	5
1.2.1	Mining Power Concentration . . . . .	5
1.2.2	Bad incentives . . . . .	6
1.2.3	Cost . . . . .	7
1.2.4	Control . . . . .	8
1.3	Smart Contracts . . . . .	8
1.4	Correctness . . . . .	9

<b>2</b>	<b>Abstract Blockchains</b>	<b>10</b>
2.1	Three Protocols . . . . .	10
2.1.1	Network Protocol . . . . .	10
2.1.2	Transaction Protocol . . . . .	11
2.1.3	Consensus Protocol . . . . .	11
2.2	Network Shell . . . . .	11

<b>3</b>	<b>Proof-of-Stake</b>	<b>12</b>
3.1	Is Proof-of-Stake Impossible? . . . . .	12
3.2	Mitigations . . . . .	13
3.2.1	Checkpoints . . . . .	13
3.2.2	Statistical Detection . . . . .	13
3.3	The Nothing-At-Stake Problem . . . . .	14
3.4	Threat Models . . . . .	14

<b>4</b>	<b>Potential Developments</b>	<b>15</b>
4.1	Privacy Preserving Transactions . . . . .	15
4.1.1	Ring Signatures . . . . .	15
4.1.2	Non Interactive Zero-knowledge Proofs of Knowledge . . . . .	15
4.2	Amendment Rules . . . . .	15
4.2.1	Constitutionalism . . . . .	15
4.2.2	Futarchy . . . . .	16
4.3	Solving Collective Action Problems . . . . .	16

# Tezos , **the last** crypto currency

## Problems

**The Protocol Fork Problem**

**Shortcomings of Proof-of-Work**

**Smart Contracts**

**Correctness**

## Solutions

**Self-Amending & On-chain governance**

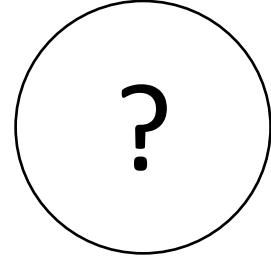
**LPoS**

**Smart Contracts**

**Formal Verification**

**Abstract Blockchain**

# Tezos , **the last** crypto currency



## 1st Gen. Bitcoin

First ever Cryptocurrency  
and blockchain

## 2nd Gen. Ethereum

Turing complete  
Smart contract

## Who's the next?

Scalability  
Inter-operability  
Sustainability

**Motivations**  
Protocol fork  
Proof of Work  
Smart Contract  
Correctness



## Abstract blockchain

Self-Amending  
On-chain governance  
LPoS  
Smart contract  
Formal Verification  
Abstract Blockchain

# PoS 합의 알고리즘

# Why PoS? No more PoW

## Inefficient

PoW의 지나친 에너지 소모와 환경 파괴  
Hash puzzle 경쟁으로 인한 중복 계산

## Centralization

ASIC과 마이닝 풀의 등장

## Mis-alignment of interest

채굴자의 동기는 채굴 보상  
(ex. 자동 해시 파워 이동 서비스)

## Not scalable

Hash puzzle 경쟁으로 인한 중복 계산  
Not solved by just shorter block interval (more stale blocks)

# But why PoW is still the King?

## **Time Tested**

Old-school blockchains (Bitcoin, Litecoin, Ethereum) are using PoW  
Bitcoin network has never been hacked

## **Secure**

Real proof by burning physical resources  
Hard to attack  
(Dis)Incentive – Block reward and computing power is a kind of stake

## **Fair**

Very fair guaranteed by cryptography

# What is PoS?

## PROOF

블록에 포함된 트랜잭션과 해당 블록이 적법하다는 증거 또는 증명

## STAKE

특정 퍼블릭 키 해시(주소)가 보유한 지분의 상대적 가치

특정 주소의 staking 토큰 수 / 네트워크 전체 staking 토큰 수

## PROOF OF STAKE

지분의 비율에 따라 선택된 주체가 새 블록을 생성

## FOLLOW-THE-SATOSHI (Follow-the-coin)

새롭게 생성(mined, minted)된 토큰의 **모든 최소 단위에** 고유한 시리얼 넘버를 부여

시리얼 넘버 한 개를 (무작위로) 선택하고, 이 토큰을 보유한 주소가 블록을 생성함

더 많은 토큰을 보유할수록, 블록 생성 확률이 높음

PoS also should be **fair** and **secure**

# Fair & Secure

블록 생성의 기회가 지분의 비율에 따라 공정하게 주어져야 함

스케줄 생성에 개인이 큰 영향을 끼쳐서는 안 됨

이미 정해진 스케줄이 변경되어서도 안 됨

악의적인 행동을 효과적으로 막을 수 있어야 함 (Nothing at stake, Long range attack)

PoS also should be **fair** and **secure**

**Validator selection rule**

**Disincentive for malicious behavior**

**Incentive for honest behavior**

PoS also should be **fair** and **secure**

**Validator selection rule**

**Disincentive for malicious behavior**

**Incentive for honest behavior**

# PoS of Tezos, A mix of several ideas

Validator  
Selection Rule

Dis-incentive for  
Malicious Behavior

Incentive for  
Honest Behavior

**Multiple snapshots**  
**Random seed**

**Safety deposit**  
**Plenitude rule**

**Block reward**  
**Accussing reward**

# PoS of Tezos, A mix of several ideas

Validator  
Selection Rule

Dis-incentive for  
Malicious Behavior

Incentive for  
Honest Behavior

**Multiple snapshots**  
**Random seed**

Safety deposit  
Plenitude rule

Block reward  
Accussing reward

# 용어 정리

## Baking

Block producing , Staking

## Baker

Block producer, validator, miner, forger

## Delegation

Only staking right, NOT ownership

## Cycle

Period defined in protocol

4,096 Blocks (1 block = 1 min)

Validator selection rule = Roll snapshot + Random seed

# Follow the roll

1 roll = 8,000 xtz (베이커 보유분 + 위임 받은 수량)

베이킹 파워가 를 단위로 내림(Rounded down)

8,000 ~ 15,999 xtz = 1 roll

FTS 알고리즘 측면에서는 효율적

Roll snapshot을 통해 베이킹 파워를 측정

Tezos's PoS

# **PoS with Optional Delegation**

Validator selection rule = Roll snapshot + Random seed

# Random seed

베이커들이 결정

예측이 사실상 불가능한 무작위 숫자열

# Validator selection rule = Roll snapshot + Random seed

## **Roll snapshot**

지분을 얼마나 보유하고 있는가

+

## **Random seed**

예측이 사실상 불가능한 무작위 숫자열



## **Baking rights**

블록 B는 베이커 A가 생성한다

# Validator selection rule = Roll snapshot + Random seed

## Cycle

1 Cycle = 4,096 블록 = 4,096분 = 2일 20시간 16분 (best case)

사이클 단위로 베이킹 스케줄이 미리 결정됨



## Baking rights in cycle 1

Block 4,097: 베이커(Alice, Diego, Arthur, Charlie, Satoshi, ...)

Block 4,098: 베이커(Satoshi, Charles, Dan, Vitalik, Kwon, ...)

Block 4,099: 베이커(Justin, Kate, Arthur, Satoshi, Satoshi, ...)

...

Block 8,192: 베이커(Alice, Diego, Arthur, Charlie, Satoshi, ...)

# Validator selection rule = Roll snapshot + Random seed

Block 8,192: 베이커(Alice, Diego, Arthur, Charlie, Satoshi, ...)



**가장 높은** 우선순위의 베이커는 이전 블록 생성 후 **1분** 후 베이킹 가능

**2번째** 우선순위의 베이커는 이전 블록 생성 후 **2분** 후 베이킹 가능

...

**16분**이 지난 경우, **보증금 없이** 베이킹 가능

체인별 블록 생성 시간을 통해서 체인의 score를 가늠할 수 있음

# Validator selection rule = Roll snapshot + Random seed

Roll snapshot in **Cycle N**

지분을 얼마나 보유하고 있는가

+

Random seed in **Cycle N+1**

예측이 사실상 불가능한 무작위 숫자열



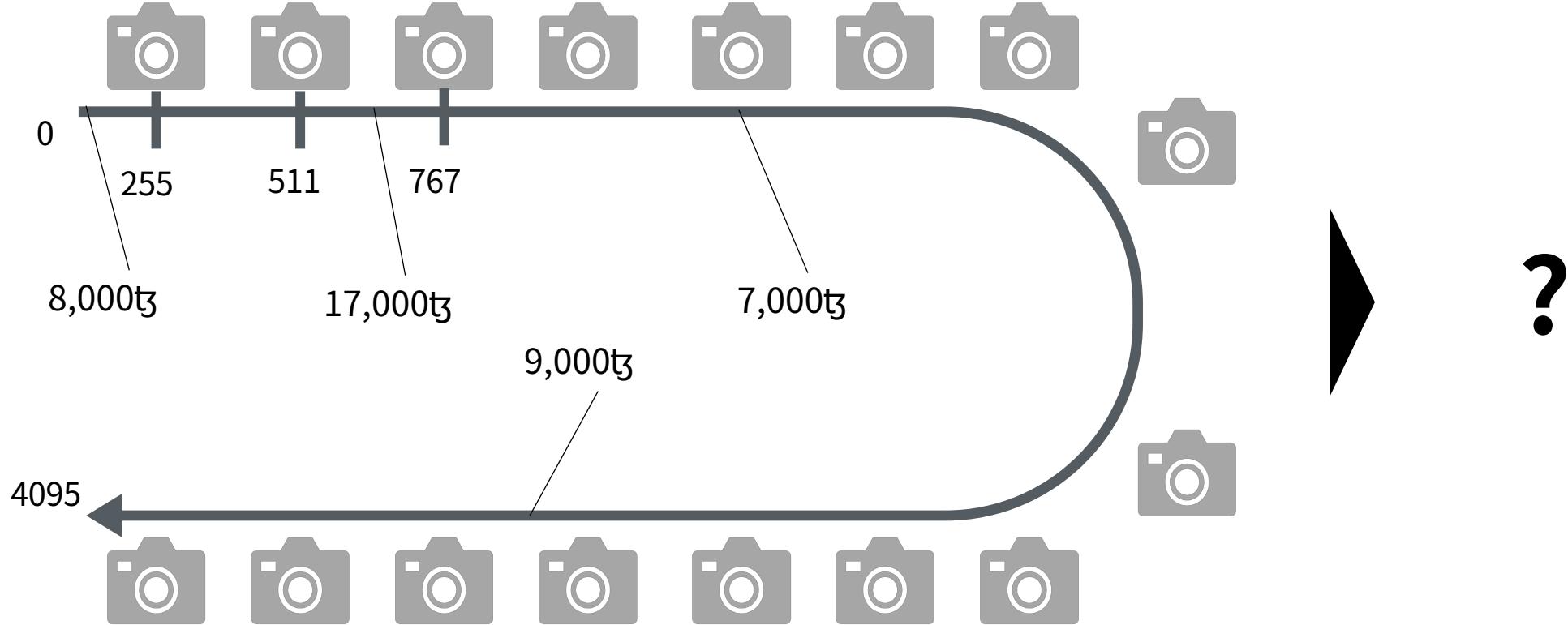
Baking rights in **Cycle N+7**

블록 B는 베이커 A가 생성한다

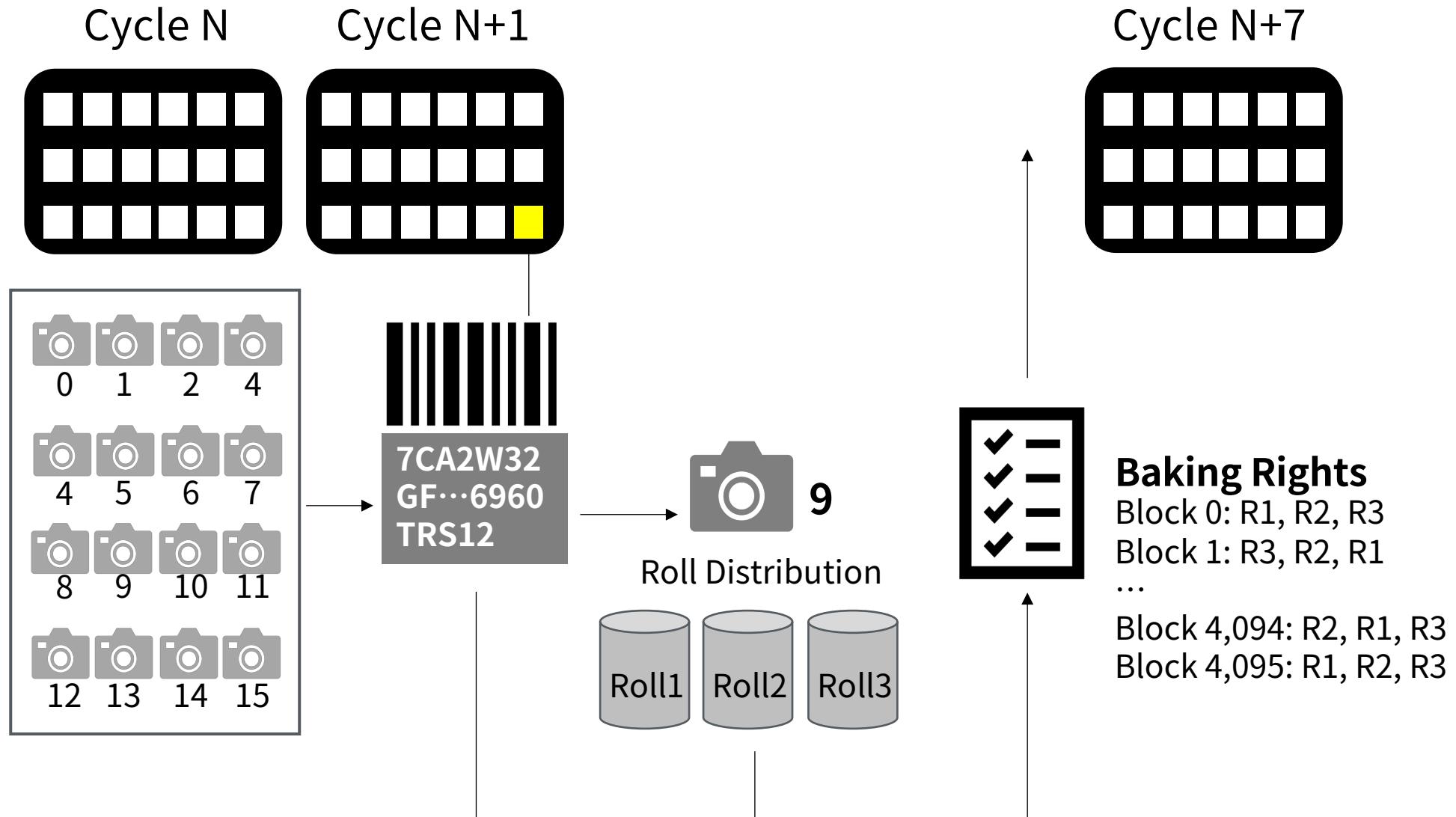
# Validator selection rule = Roll snapshot + Random seed

Multiple snapshots (16 in a cycle) and random selection

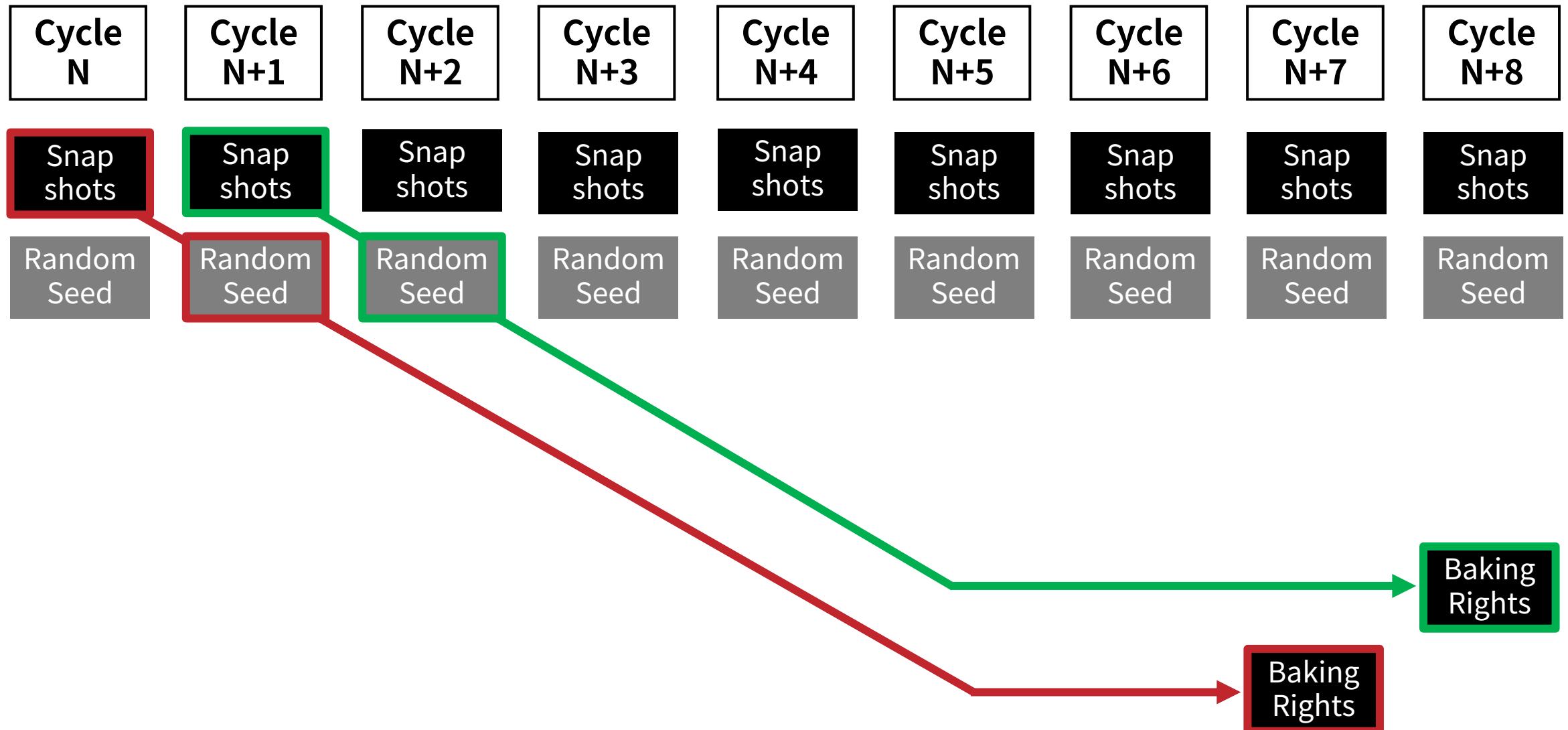
Proof of **STAKING**



# Validator selection rule = Roll snapshot + Random seed



# Validator selection rule = Roll snapshot + Random seed



Validator selection rule = Roll snapshot + Random seed

## Random seed

베이커는 미리 정해진 베이킹 스케줄에 따라 블록B를 생성

임의의 숫자를 암호화(**hash commitment**)하여 블록B에 포함

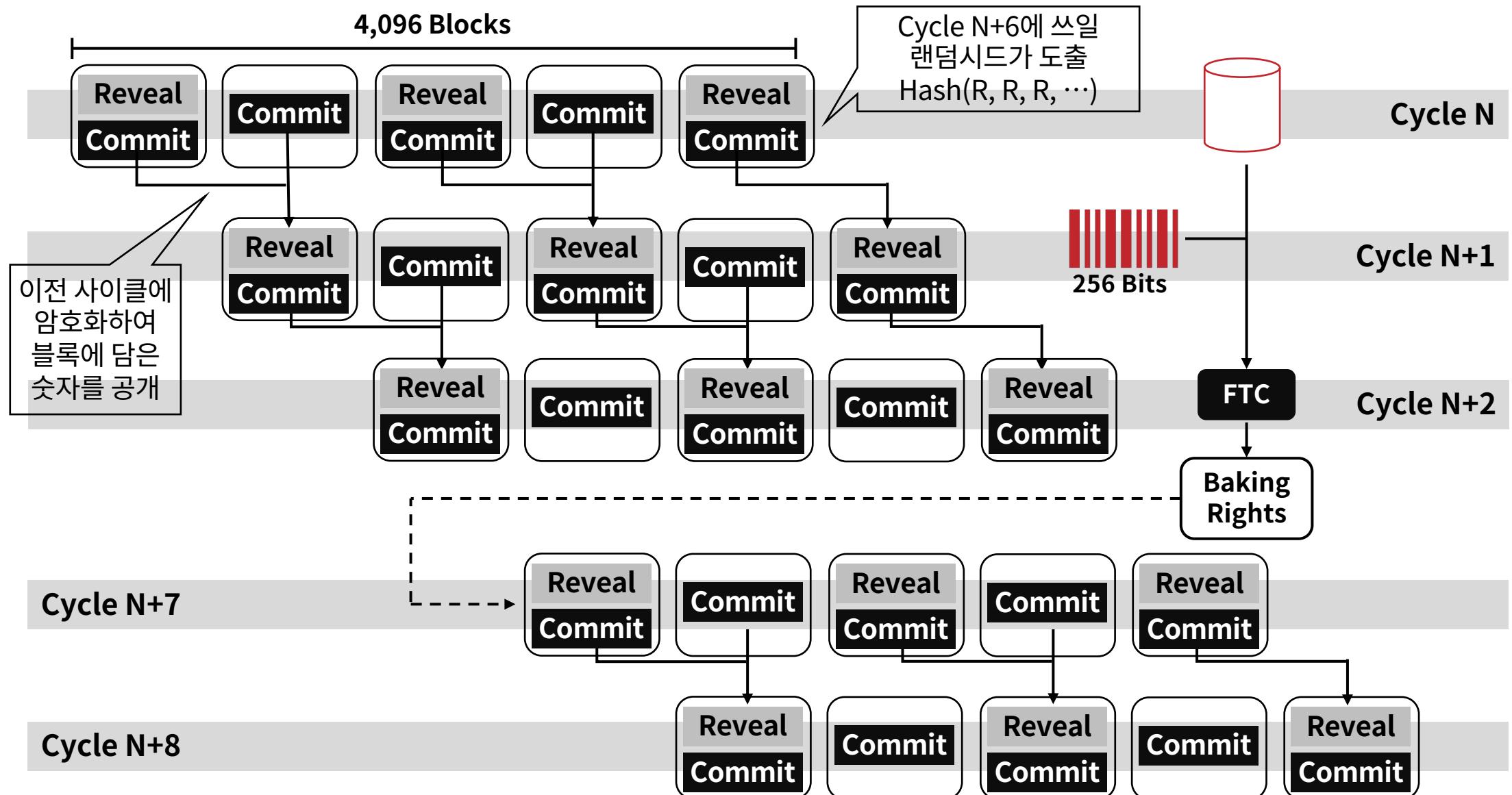
이 숫자는 다음 사이클에 공개(**reveal**)해야 함

숫자를 공개하지 않은 경우, 블록B의 보상과 수수료 몰수

매 사이클 마지막에 공개된 숫자를 모두 결합, 해싱하여 랜덤 시드 생성

공개는 32블록마다 자유롭게 가능

# Validator selection rule = Roll snapshot + Random seed



# PoS of Tezos, A mix of several ideas

Validator  
Selection Rule

Dis-incentive for  
Malicious Behavior

Incentive for  
Honest Behavior

Multiple snapshots  
Random seed

**Safety deposit  
Plenitude rule**

Block reward  
Accussing reward

Disincentive = Deposit + Endorsement

# Endorsement

블록당 32명의 Endorser가 미리 정해짐 (Roll snapshot + Random seed)

이전 블록에 투표(endorse, notarize)

$\mathcal{B}lock_i$  가 베이킹 된 후, 인도서들은  $\mathcal{B}lock_i$  에 대해 서명을 제출함

$\mathcal{B}lock_i$  베이커가 이 투표를 블록에 포함

체인의 score(fitness) = 인도스먼트의 합

인도성이 블록에 담기고, Canonical chain이 되어야 보상

보상 = 최대  $2 \frac{\epsilon}{\delta T}$  ( $2 / dT$ )

Disincentive = Deposit + Endorsement

# Safety Deposit

베이킹, 인도싱에 요구되는 조건

5사이클 동안 동결(locked, frozen)

악의적인 행동 시 몰수(forfeit, slashed)

Baking	Number	Deposits	Rewards
Baking	1	512 $\text{tS}$	16 $\text{tS}$
Endorsing	Up to 32	64 $\text{tS}$	2/dT $\text{tS}$

## **PoS with Optional Delegation**

**Chain-based PoS consensus with a use of endorsements to speed-up confirmation times and reduce selfish baking**

# PoS of Tezos, A mix of several ideas

Validator  
Selection Rule

Dis-incentive for  
Malicious Behavior

Incentive for  
Honest Behavior

Multiple snapshots  
Random seed

Safety deposit  
Plenitude rule

**Block reward**  
**Accussing reward**

# Incentive

## Baking reward

Steal: 후순위 베이커가 베이킹 하는 경우

## Endorsing reward

최대  $2 \frac{t_3}{dT}$  (2 / dT)

## Accusing reward

악의적인 행동을 신고

악의적인 행동이 적발된 베이커는 Safety deposit 몰수

몰수된 토큰의 절반은 소각(proof of burn), 절반은 신고자에게 지급