CS 190
Blockchain Technologies and Security

Yu Feng
University of California, Santa Barbara

UCSB

Slides adapted from Stanford CS251

About me

- Research areas: PL/blockchains/security
- Cofounded two web3 startups
  - Security: Veridise Inc.
  - Distributed systems: Riema Labs
- Conducted Web3/blockchain security research since 2018
- Office: HFH-2157

What is a blockchain for?

Abstract answer: a blockchain provides coordination between many parties, when there is no single trusted party

if trusted party exists ⇒ no need for a blockchain

What is a blockchain?

Abstract answer: a blockchain provides coordination between many parties, when there is no single trusted party

if trusted party exists ⇒ no need for a blockchain

[financial systems: often no trusted party]

Blockchains: what is the new idea?

2009

Bitcoin

Several innovations:

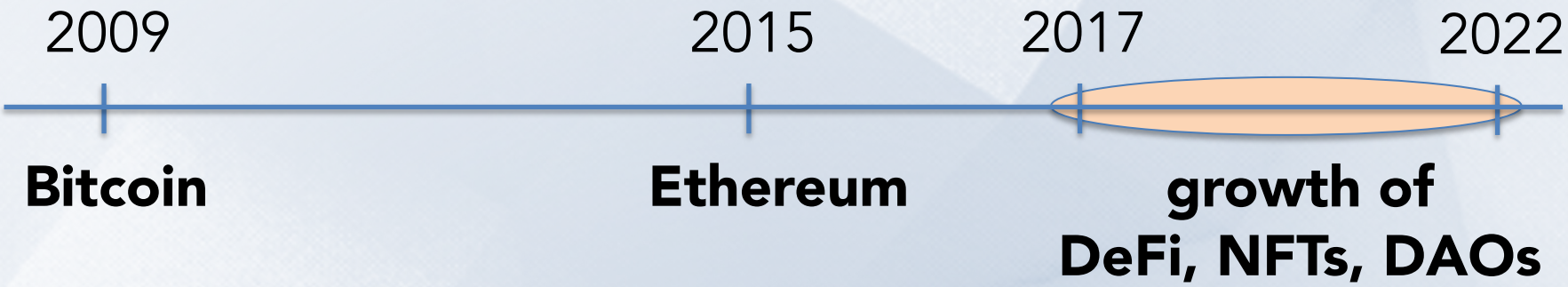
- A practical **public append-only data structure**, secured by replication and incentives
- A fixed supply asset (BTC). Digital payments, and more.

Blockchains: what is the new idea?

Several innovations:

- **Blockchain computer**: a fully programmable environment
  ⟹ public programs that manage digital and financial assets
- **Composability**: applications running on chain can call each other

Blockchains: what is the new idea?



So what is this good for?

(1) Basic application: a digital currency (stored value)

- Current largest: Bitcoin (2009), Ethereum (2015)
- Global: accessible to anyone with an Internet connection

Opinion

The New York Times

Bitcoin Has Saved My Family

"Borderless money" is more than a buzzword when you live in a collapsing economy and a collapsing dictatorship.

By Carlos Hernández

Mr. Hernández is a Venezuelan economist.

Feb. 23, 2019

What else is it good for?

(2) Decentralized applications (DAPPs)

- **DeFi**: financial instruments managed by public programs
  - examples: stablecoins, lending, exchanges, ….
- **Asset management** (NFTs): art, game assets, domain names.
- **Decentralized organizations** (DAOs): (decentralized governance)
  - DAOs for investment, for donations, for collecting art, etc.

(3) New programming model: writing decentralized programs

Assets managed by DAPPs

Sep. 2023

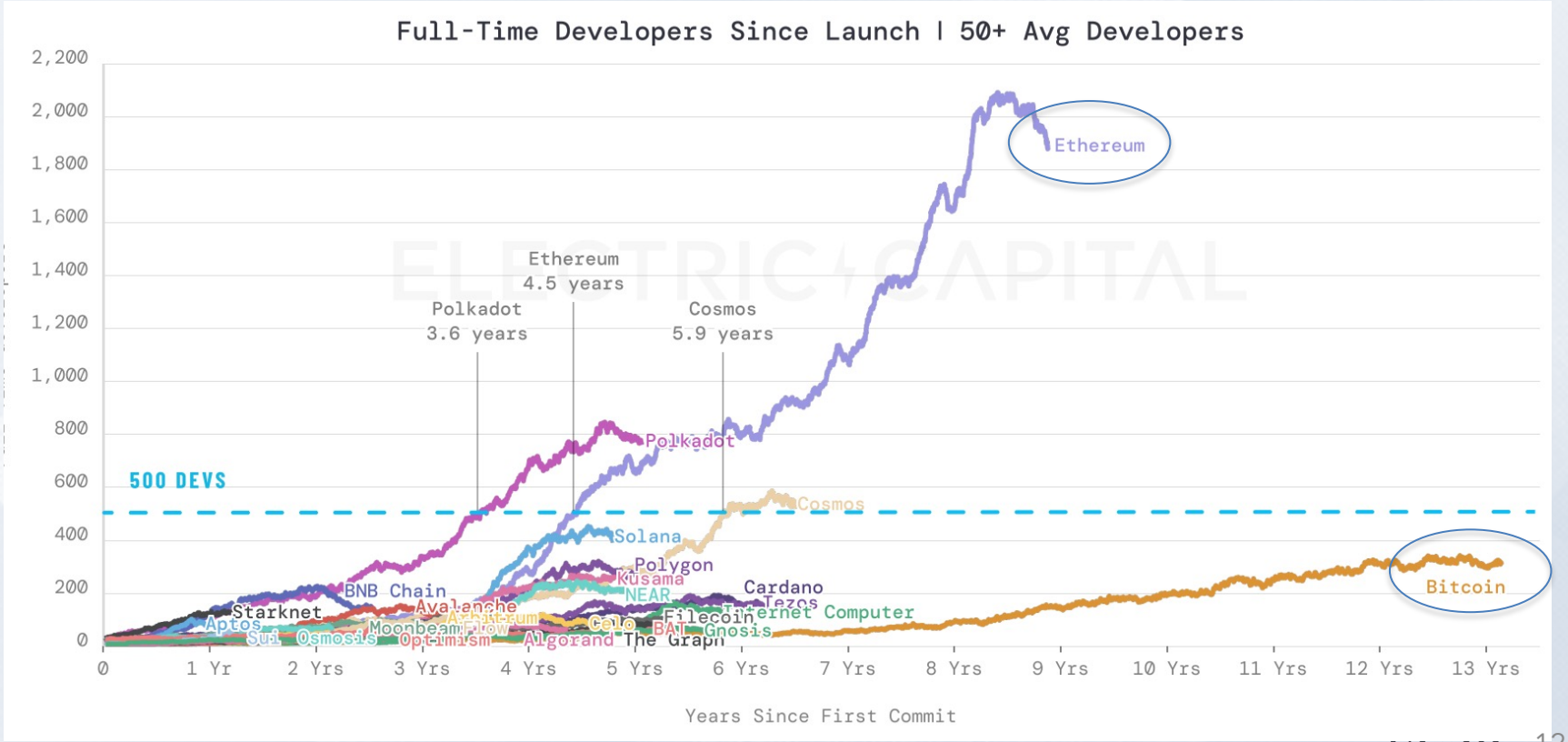
MakerDAO	Ethereum	StableCoin	$4.5B
Curve	Ethereum	Exchange	$2.2B
Aave V3	Ethereum	Lending	$2.3B
Uniswap V3	Ethereum	Exchange	$3.1B
Compound	Ethereum	Lending	$1.8B

Transaction volume

	24h volume
Bitcoin • BTC	$9.9B
Ethereum • ETH	$3.4B
USDC USDC	$2.7B

Sep. 2023

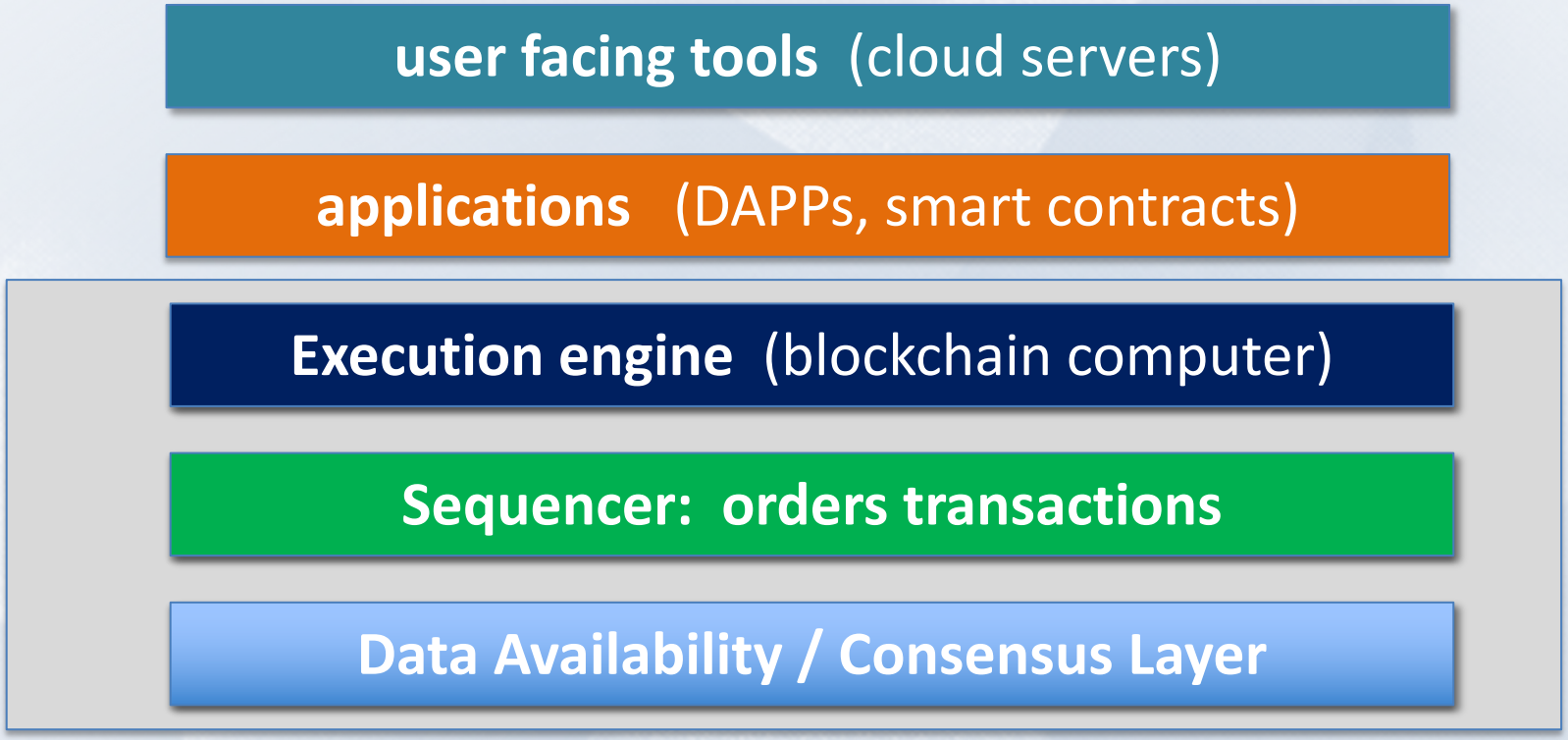
Active developers since launch (as of 12/31/2022)



source: electric capital

Central Bank Digital Currency (CBDC)

What is a blockchain?



Consensus layer (informal)

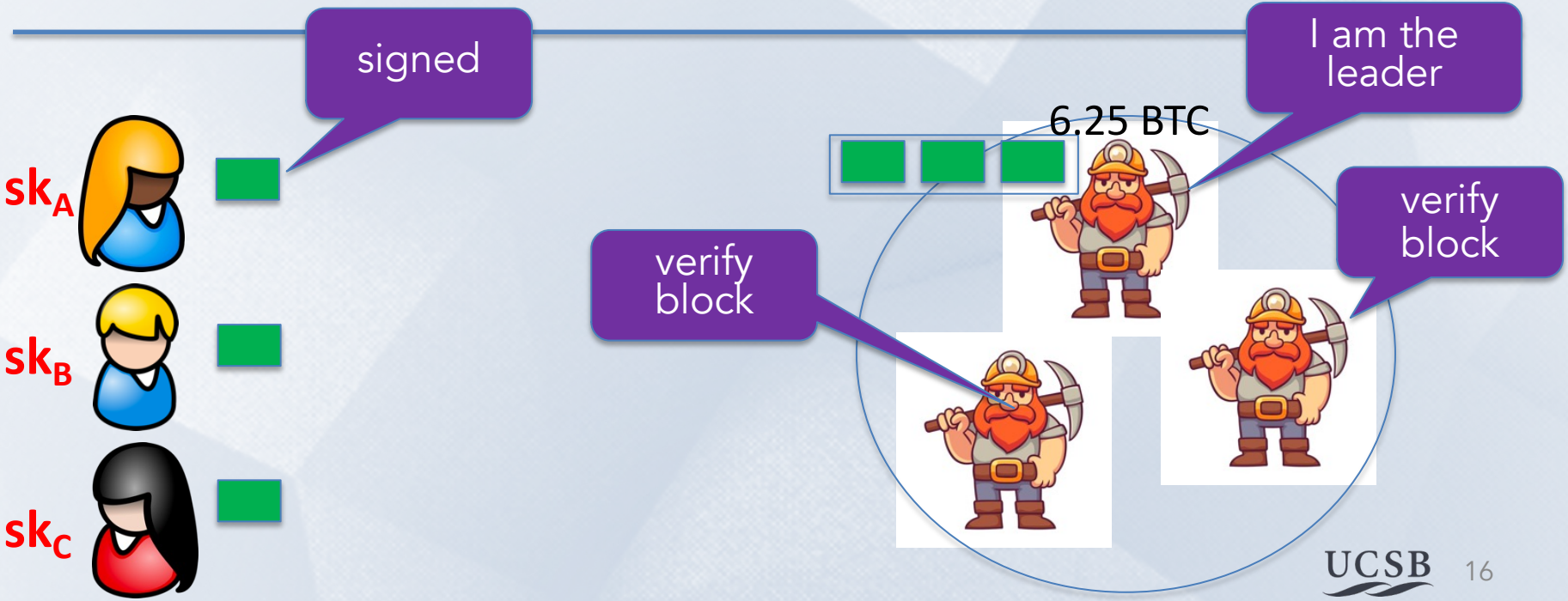
A public append-only data structure:

achieved by replication

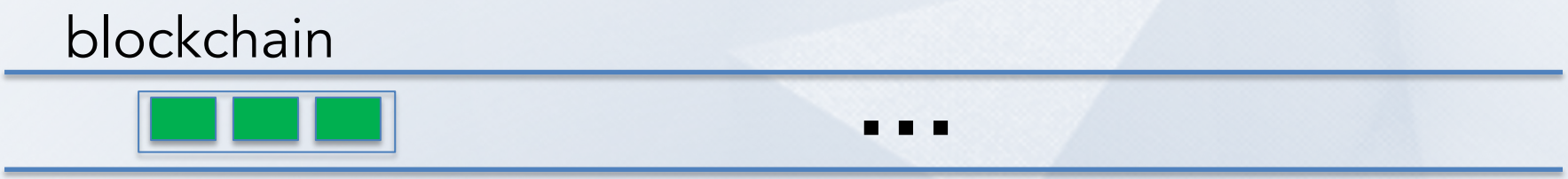
- **Persistence**: once added, data can never be removed*
- **Safety**: all honest participants have the same data**
- **Liveness:** honest participants can add new transactions
- **Open(?)**: anyone can add data (no authentication)

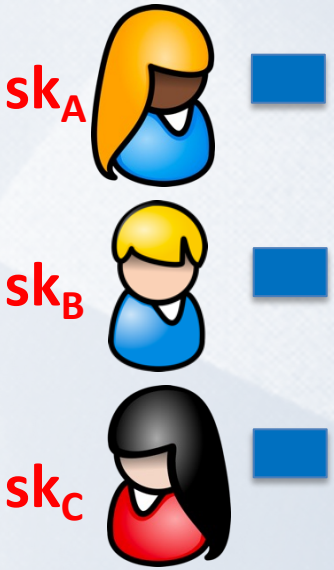
Data Availability / Consensus layer

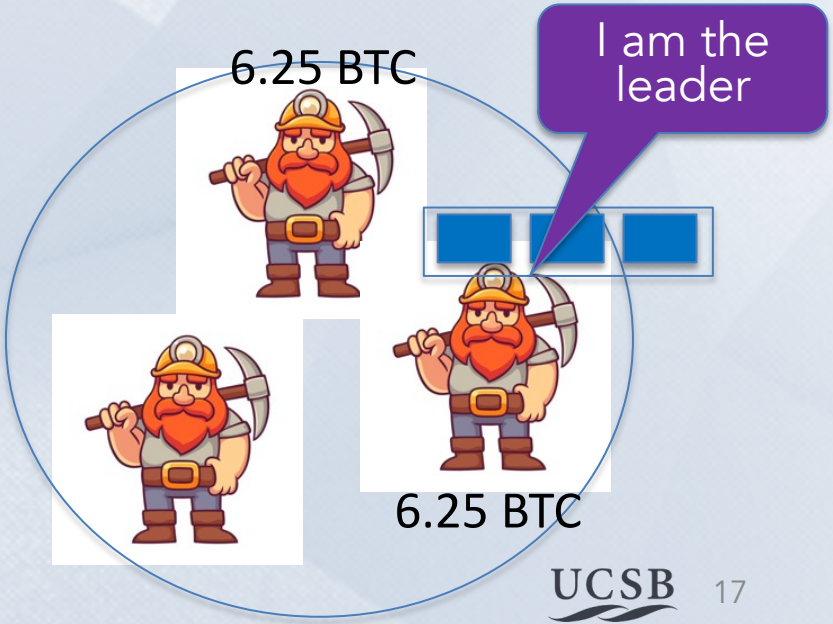
How are blocks added to chain?



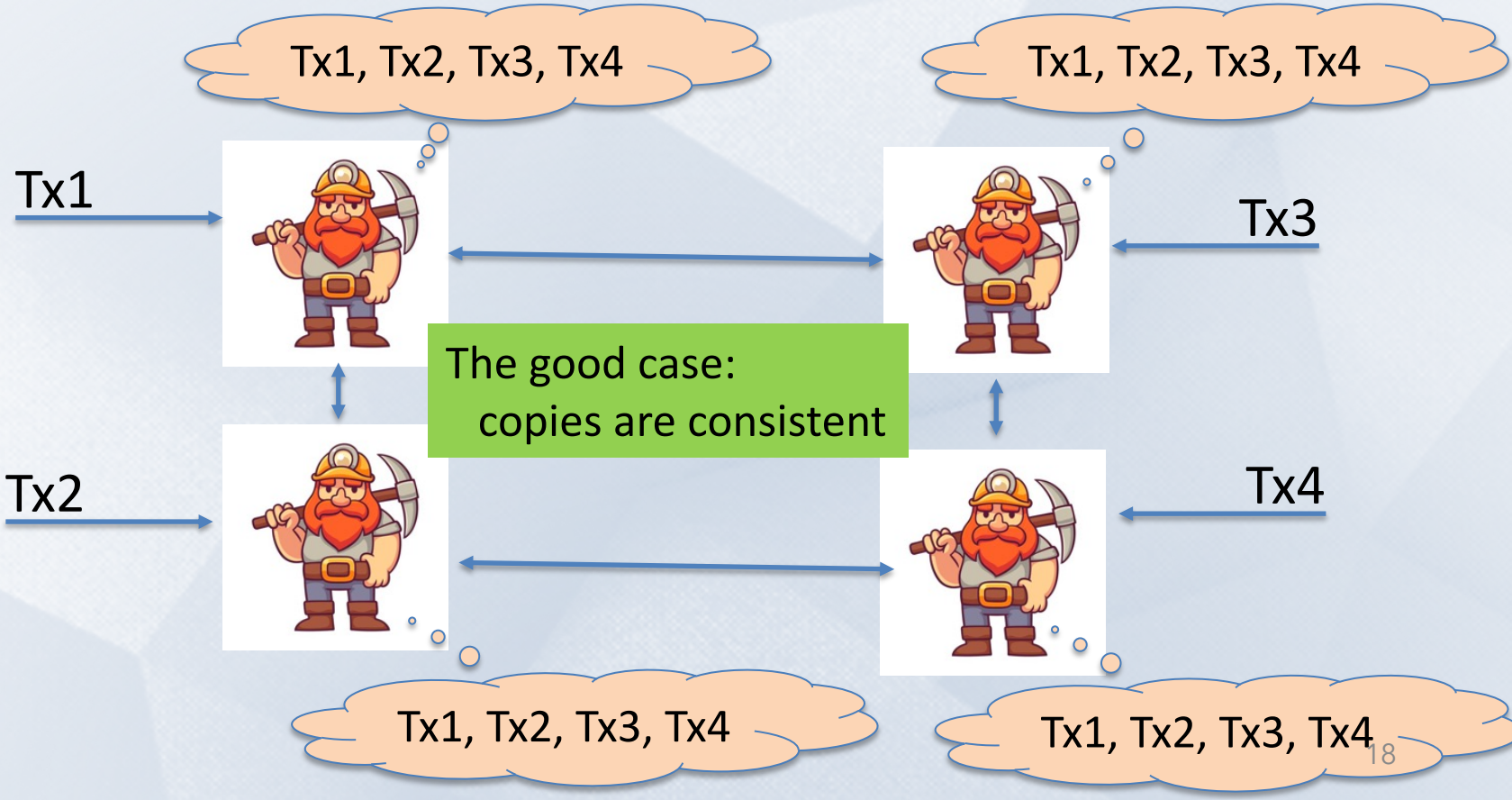
How are blocks added to chain?



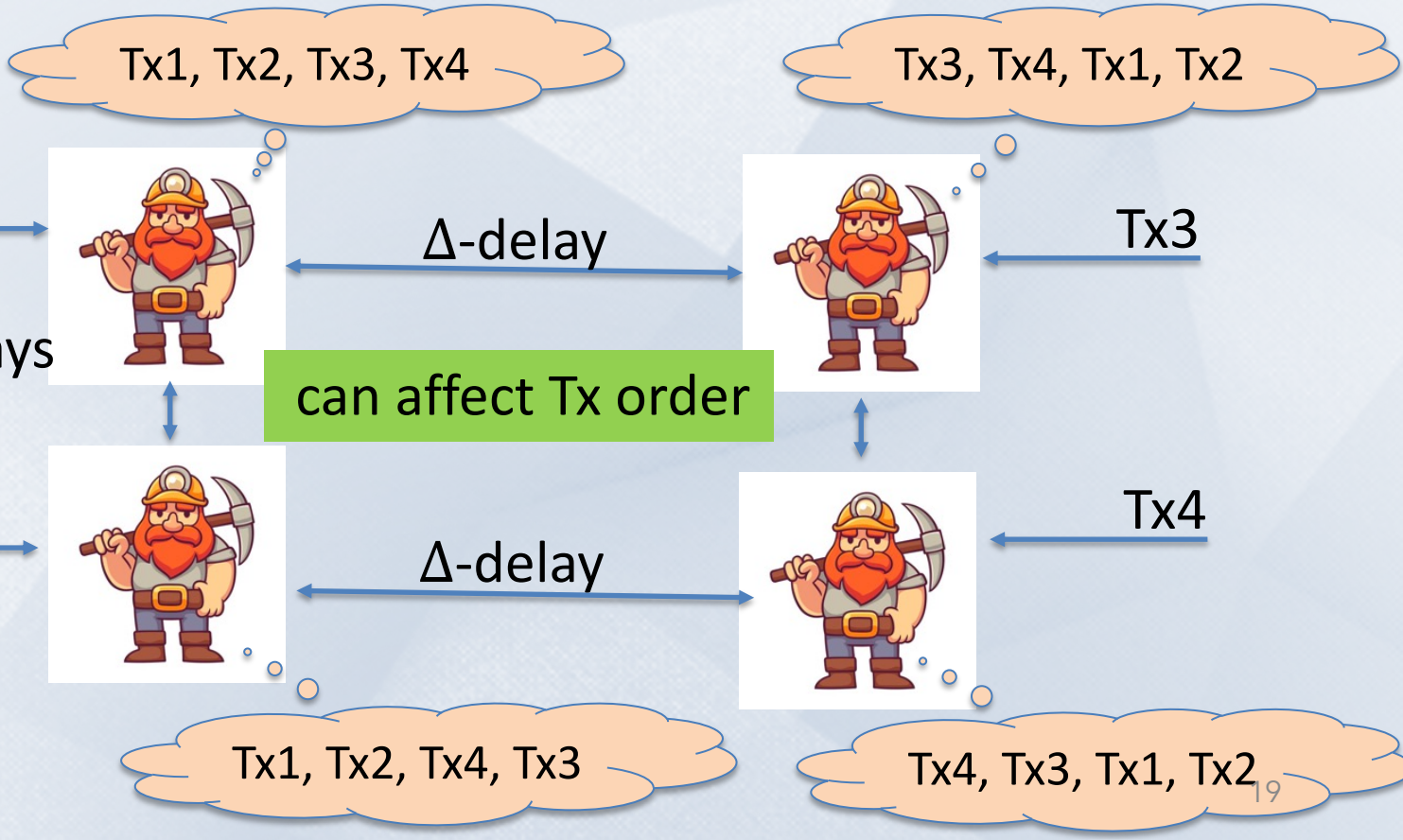




Why is consensus a hard problem?



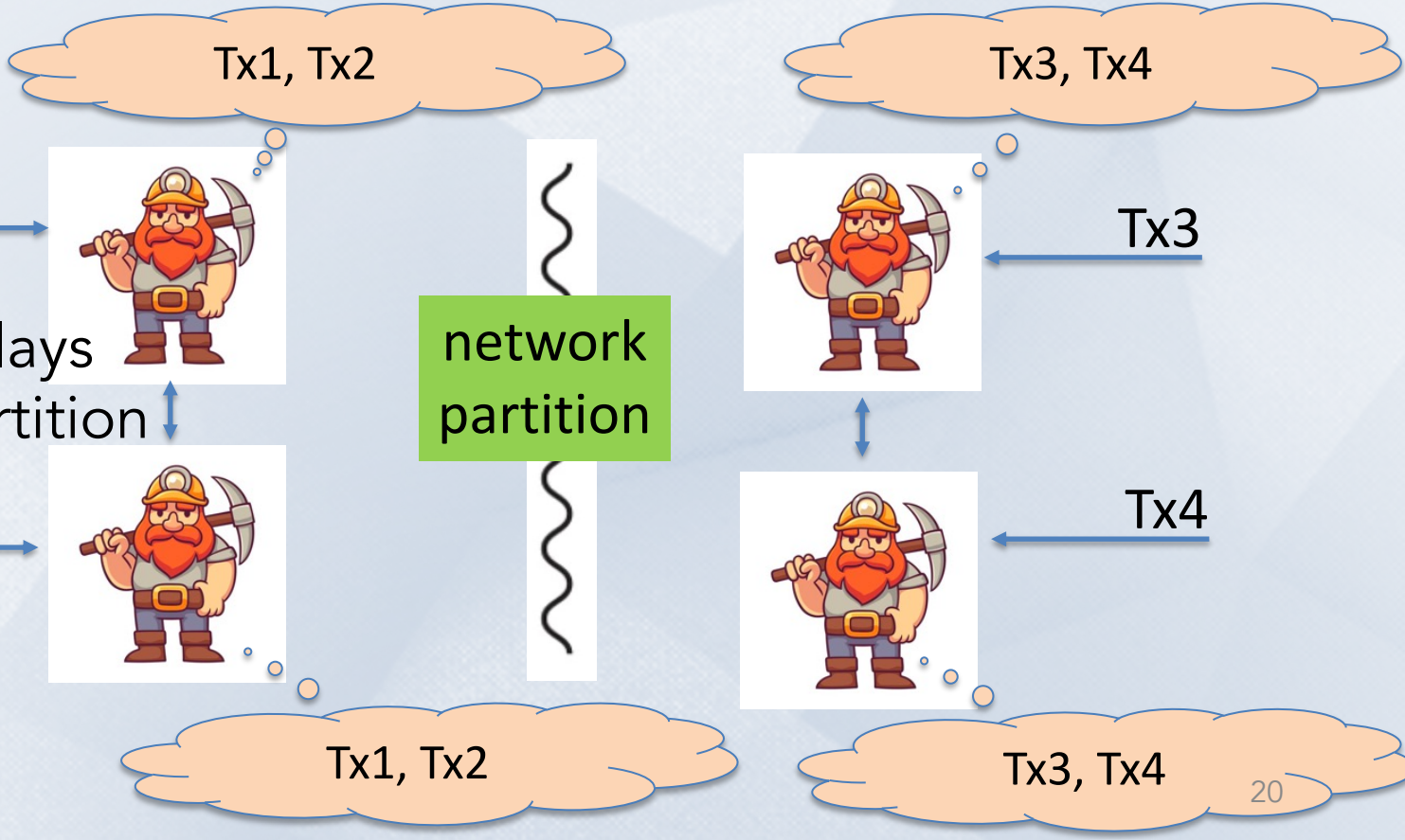
Why is consensus a hard problem?



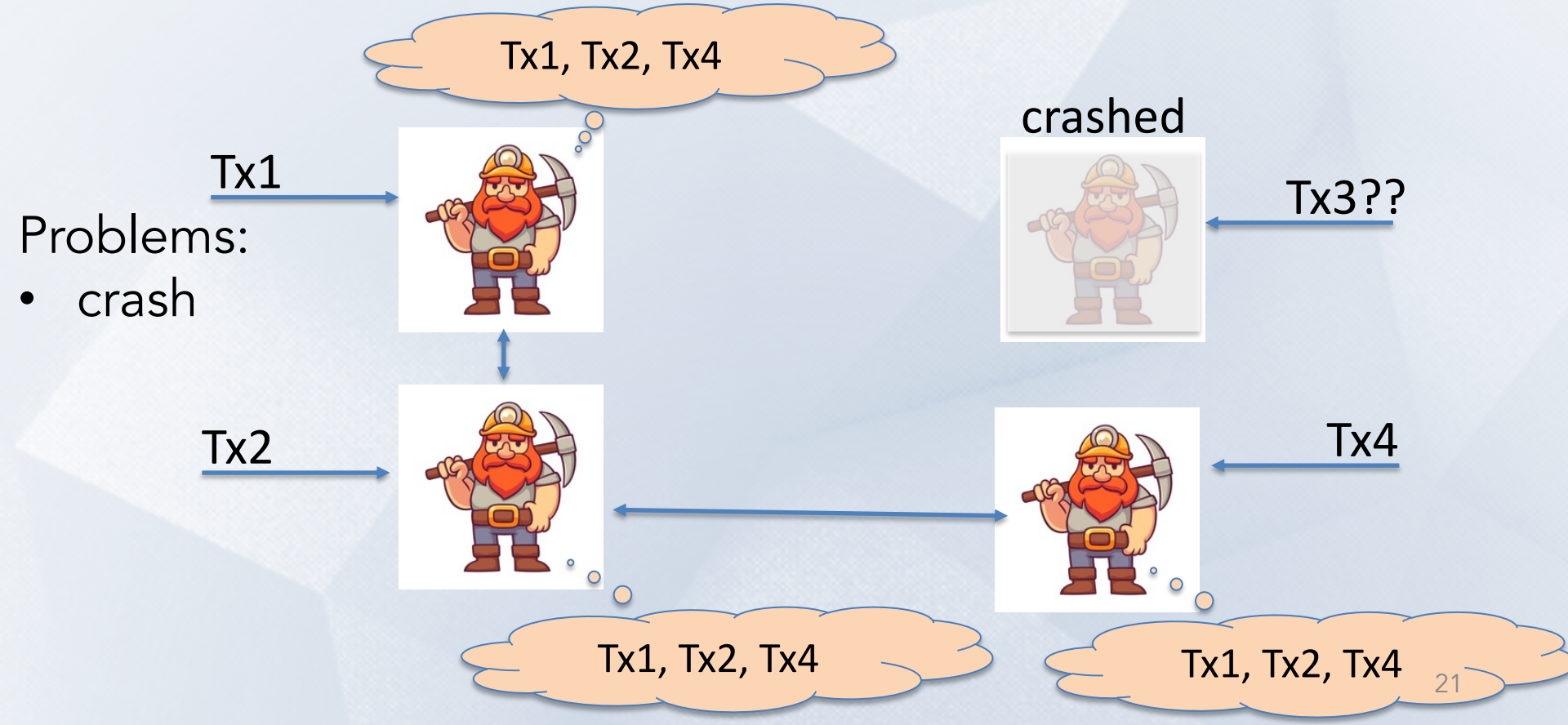
Why is consensus a hard problem?

Problems:

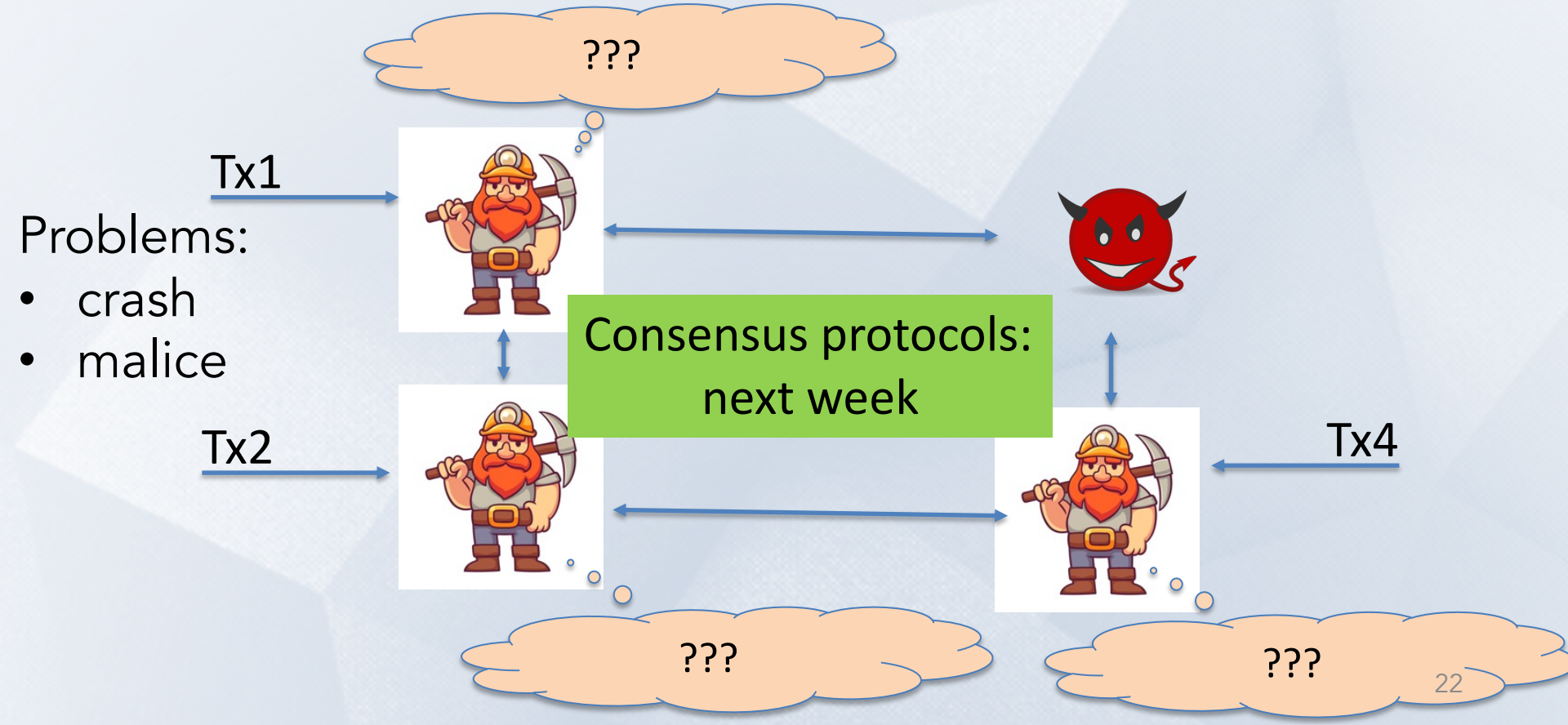
- Network delays
- Network partition



Why is consensus a hard problem?



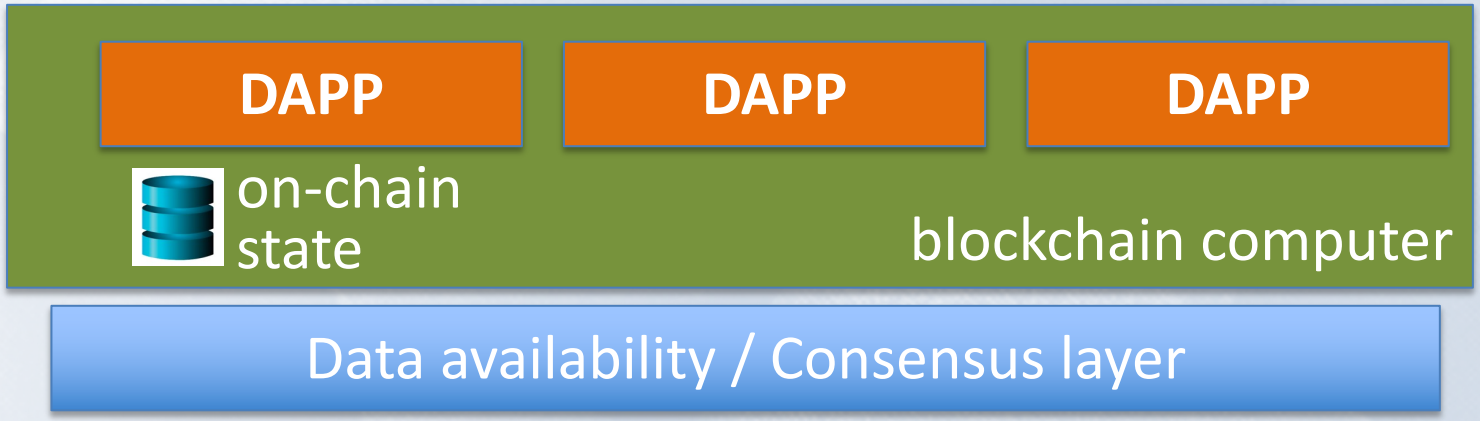
Why is consensus a hard problem?



Next layer: the blockchain computer

Decentralized applications (DAPPs):

- Run on blockchain: code and state are written on chain
- Accept Tx from users ⇒ state transitions are recorded on chain



Next layer: the blockchain computer

Top layer: user facing servers

end user

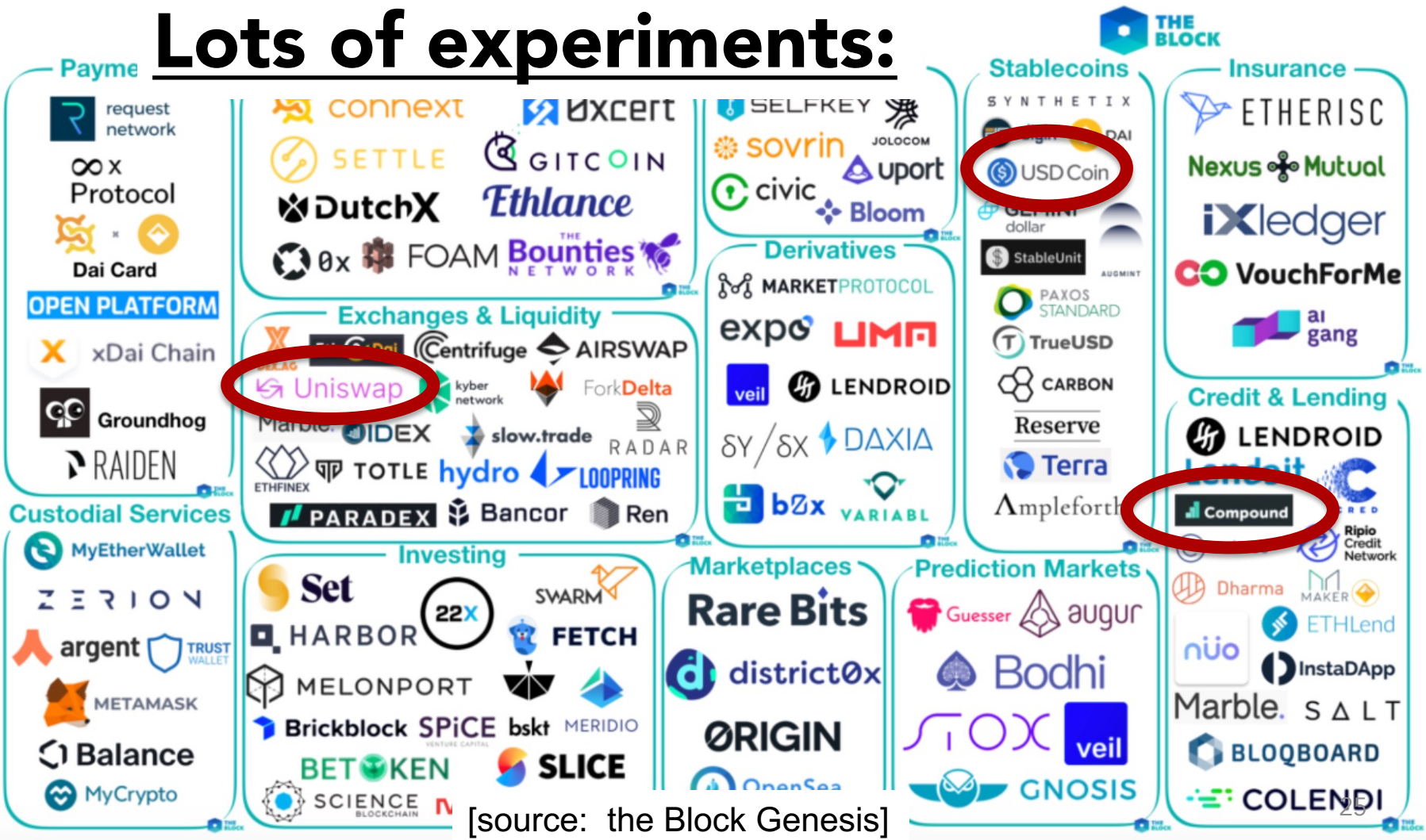
DAPP **DAPP** **DAPP**

on-chain state

blockchain computer

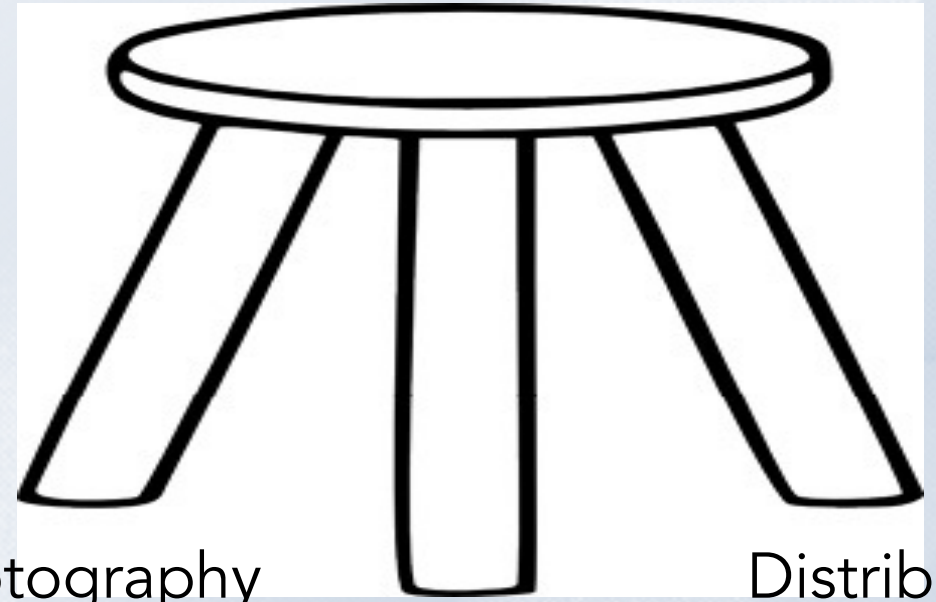
Data availability / Consensus layer

Lots of experiments:



[source: the Block Genesis]

This course



Cryptography

Distributed systems

Economics/Security

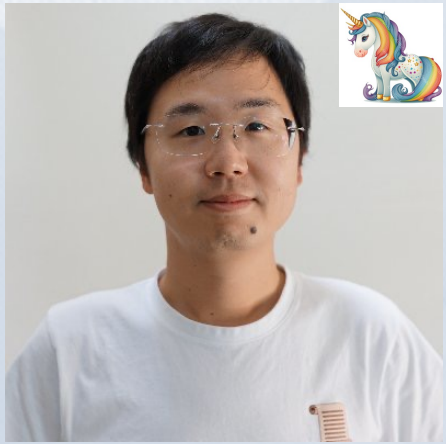
Course organization

1. The starting point: Bitcoin mechanics
2. Consensus protocols
3. Ethereum and decentralized applications
4. DeFi: decentralized applications in finance
5. Private transactions on a public blockchain
   (SNARKs and zero knowledge proofs)
6. Scaling the blockchain: getting to 10K Tx/sec
7. Interoperability among chains: bridges and wrapped coins

Instructors

Yu Feng

Yanju Chen
(Cofounder at Veridise)



Haichen Shen
(Cofounder at Scroll)

Scott Sunarto
(Cofounder at Argus)

Course organization

https://github.com/fredfeng/CS190-blockchain

- Homework projects, final project
- Optional weekly sections on Friday

Please tell us how we can improve …
Don’t wait until the end of the quarter

Course organization

- HW1: Merkle tree
- HW2: Complex DeFi Protocol with Solidity
- HW3: Hack DeFi: a series of web3 protocols that contain vulnerability; students are asked to hack them (generate attacks) using Foundry.
- HW4: Hack ZK: a series of ZK circuits that contain vulnerability; students are asked to hack them (generate attacks).

Course logistics

https://github.com/fredfeng/CS190-blockchain

- 4 homework projects (15% x 4 = 60%),
- 1 final project (proposal:10%, poster:15%, report: 15%)
- (Optional weekly sections on Friday)

  https://tinyurl.com/ywz3zzw4

Let’s get started …

UCSB

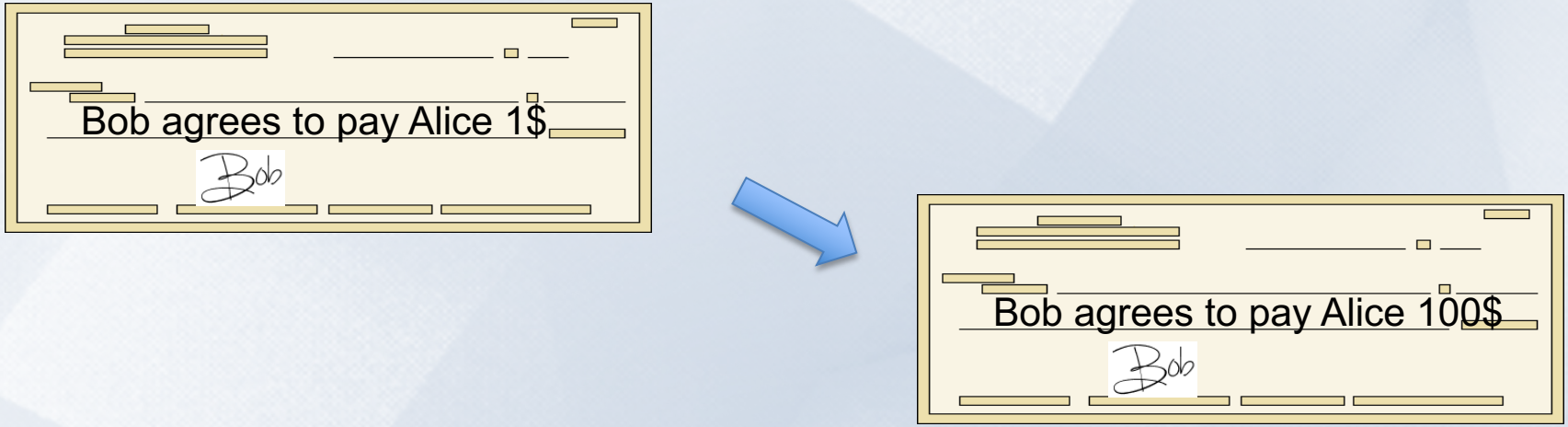
Cryptography background: Digital Signatures

How to authorize a transaction

UCSB

Signatures

Physical signatures: bind transaction to author

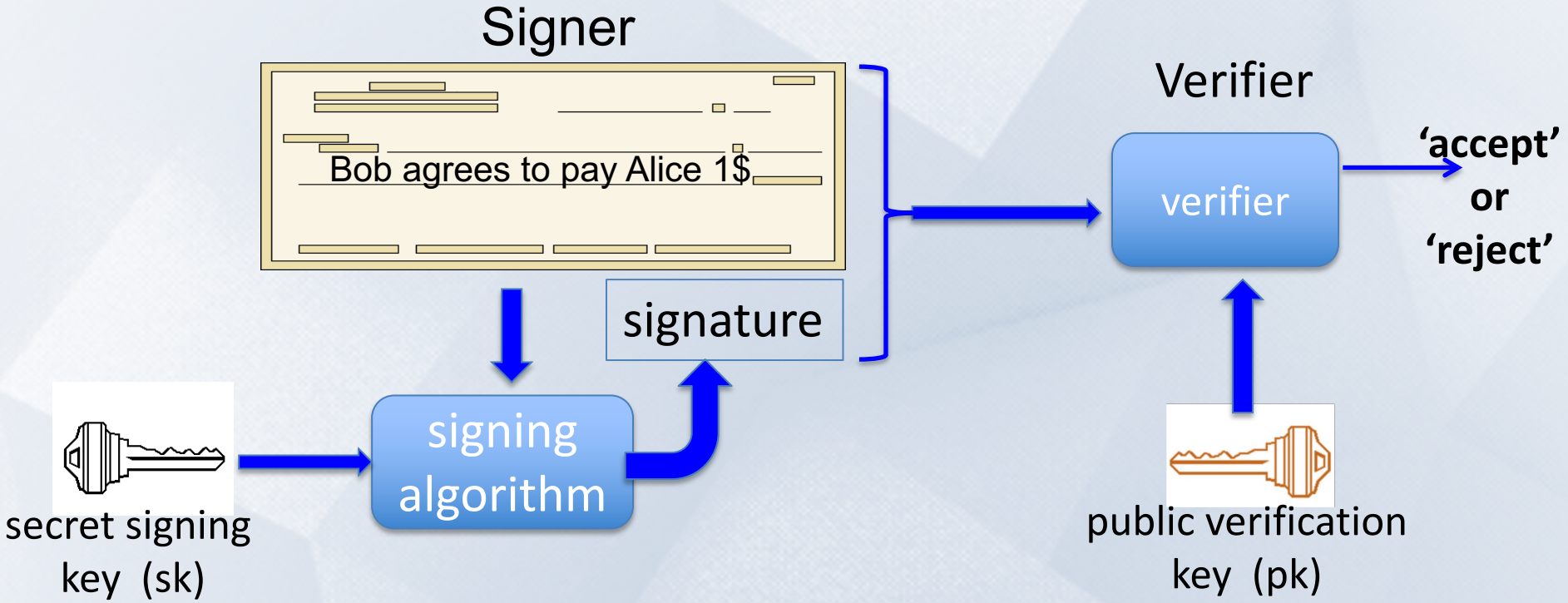


Problem in the digital world:

anyone can copy Bob's signature from one doc to another

Digital signatures

Solution: make signature depend on document



Digital signatures: syntax

Def: a signature scheme is a triple of algorithms:

- **Gen**(): outputs a key pair (pk, sk)
- **Sign**(sk, msg) outputs sig. **σ**
- **Verify**(pk, msg, **σ**) outputs 'accept' or 'reject'

Secure signatures: (informal)

Adversary who sees signatures **on many messages** of his choice,

cannot forge a signature on a new message.

Families of signature schemes

1. RSA signatures (old … not used in blockchains):
   - long sigs and public keys (≥256 bytes), fast to verify
2. Discrete-log signatures: Schnorr and ECDSA (Bitcoin, Ethereum)
   - short sigs (48 or 64 bytes) and public key (32 bytes)
3. BLS signatures: 48 bytes, aggregatable, easy threshold

   (Ethereum 2.0, Chia, Dfinity)

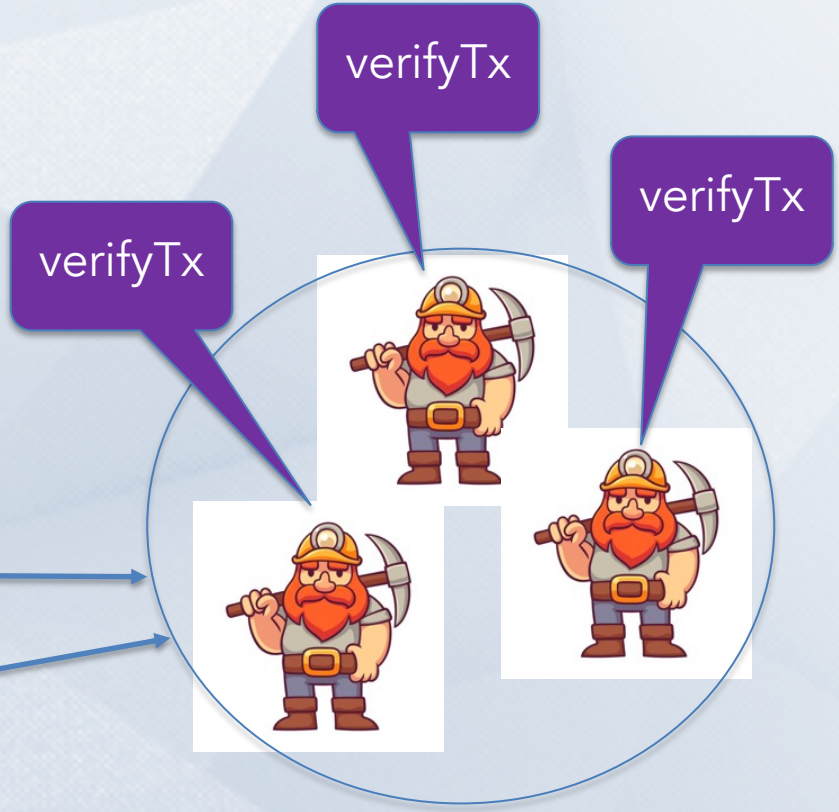
---

4. Post-quantum signatures: long (≥600 bytes)

Signatures on the blockchain

Signatures are used everywhere:

- ensure Tx authorization,
- governance votes,
- consensus protocol votes.



sk_1

data | signatures

sk_2

data | signatures

END OF LECTURE

Next lecture: the Bitcoin blockchain

UCSB