Blockchain Primitives

SPEAKER
Dan Boneh, Stanford University

a16z
[CRYPTO STARTUP SCHOOL]

Important Disclosures

The views expressed here are those of the individual AH Capital Management, L.L.C. ("a16z") personnel quoted and are not the views of a16z or its affiliates. Certain information contained in here has been obtained from third-party sources, including from portfolio companies of funds managed by a16z. While taken from sources believed to be reliable, a16z has not independently verified such information and makes no representations about the enduring accuracy of the information or its appropriateness for a given situation.

This content is provided for informational purposes only, and should not be relied upon as legal, business, investment, or tax advice. You should consult your own advisers as to those matters. References to any securities or digital assets are for illustrative purposes only, and do not constitute an investment recommendation or offer to provide investment advisory services. Furthermore, this content is not directed at nor intended for use by any investors or prospective investors, and may not under any circumstances be relied upon when making a decision to invest in any fund managed by a16z. (An offering to invest in an a16z fund will be made only by the private placement memorandum, subscription agreement, and other relevant documentation of any such fund and should be read in their entirety.) Any investments or portfolio companies mentioned, referred to, or described are not representative of all investments in vehicles managed by a16z, and there can be no assurance that the investments will be profitable or that other investments made in the future will have similar characteristics or results. A list of investments made by funds managed by Andreessen Horowitz (excluding investments for which the issuer has not provided permission for a16z to disclose publicly as well as unannounced investments in publicly traded digital assets) is available at https://a16z.com/investments/.

Charts and graphs provided within are for informational purposes solely and should not be relied upon when making any investment decision. Past performance is not indicative of future results. The content speaks only as of the date indicated. Any projections, estimates, forecasts, targets, prospects, and/or opinions expressed in these materials are subject to change without notice and may differ or be contrary to opinions expressed by others. Please see https://a16z.com/disclosures for additional important information.

What Is a Blockchain?

LAYER 3	**User Interface** (e.g., web3)
LAYER 2	**Applications** (Solidity, Move, Motoko)
LAYER 1.5	**Compute Layer** (blockchain computer)
LAYER 1	**Consensus Layer**

LAYER 1: Consensus Layer (Informal)

A **public data structure** (ledger) that provides:

Persistence: once added, data can never be removed*

Consensus: all honest participants have the same data**

Liveness: honest participants can add new transactions

Open(?): anyone can be a participant (no authentication)

LAYER 1	Consensus Layer

a16z

This Not a New Problem …

State machine replication:
studied since the 1980s

Google, Amazon, Bank of America, all have lots of servers:

- need to ensure state is consistent across all servers
- Known # of servers, and all are authorized.

open consensus

e new data to the
uth, unknown #)

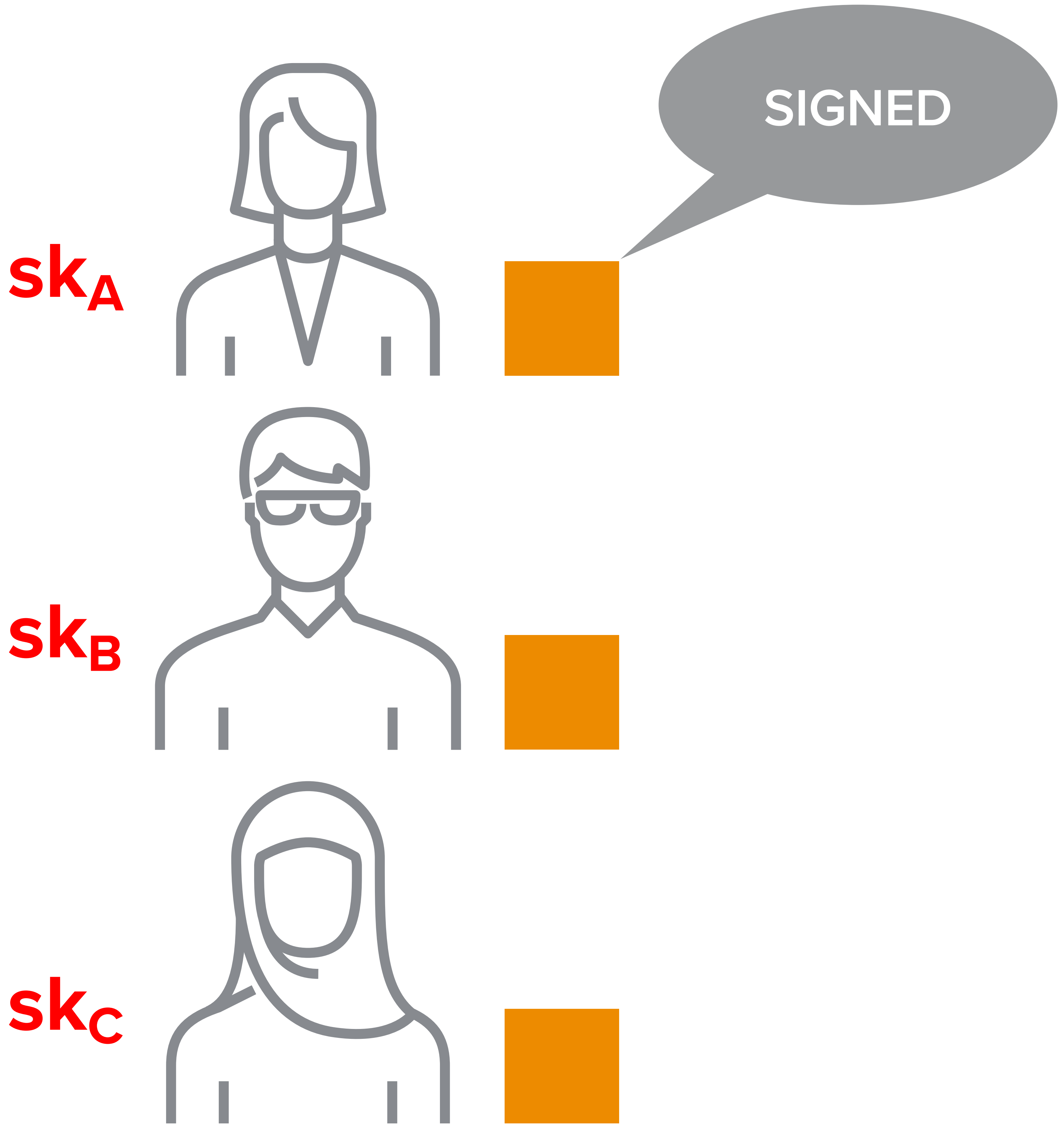
e impossible!
i, Lindell, Pass, Rabin '05]

3]:

ypass the lower
ng proof-of-work

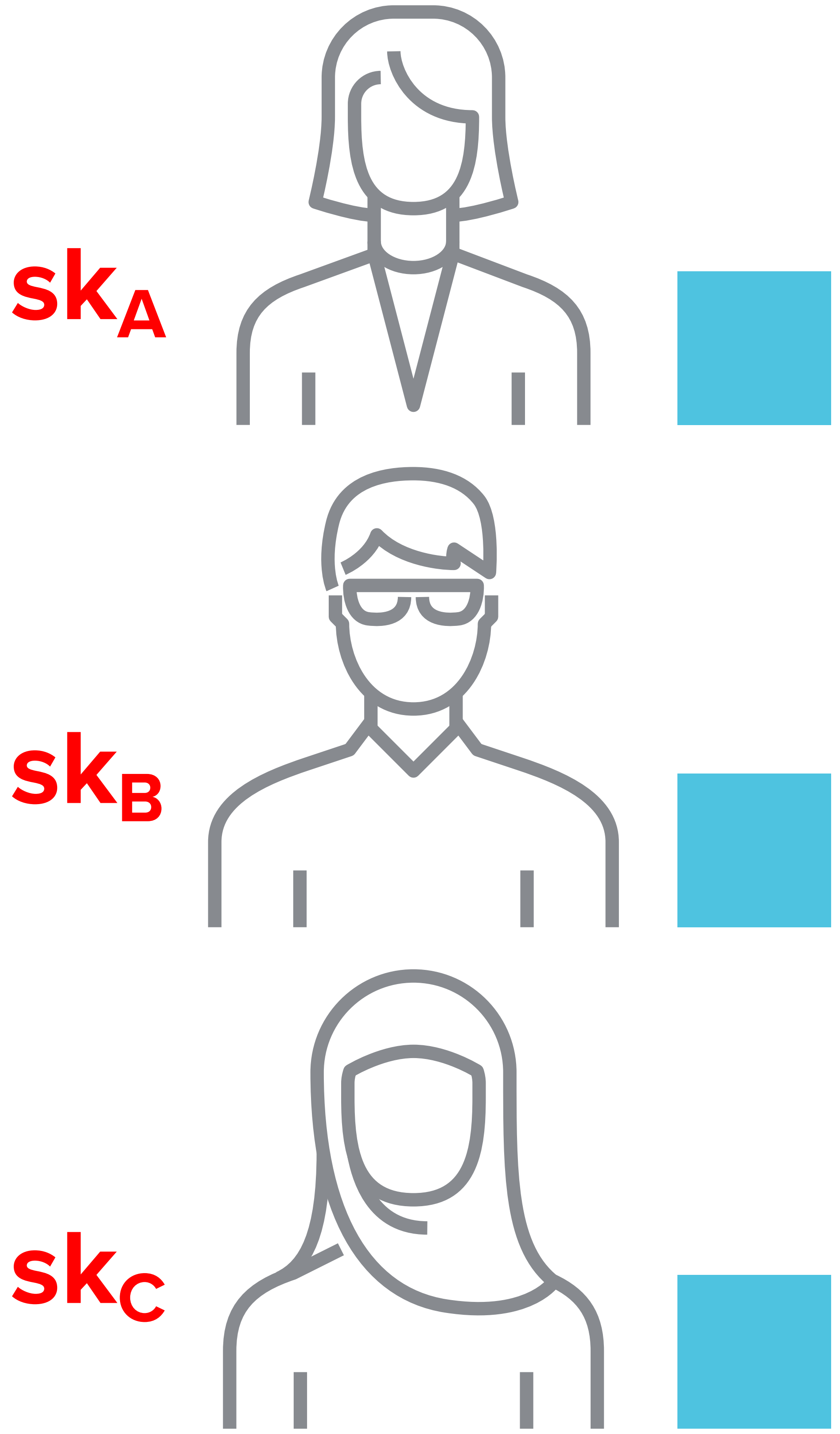
How Are Blocks Added to Chain?

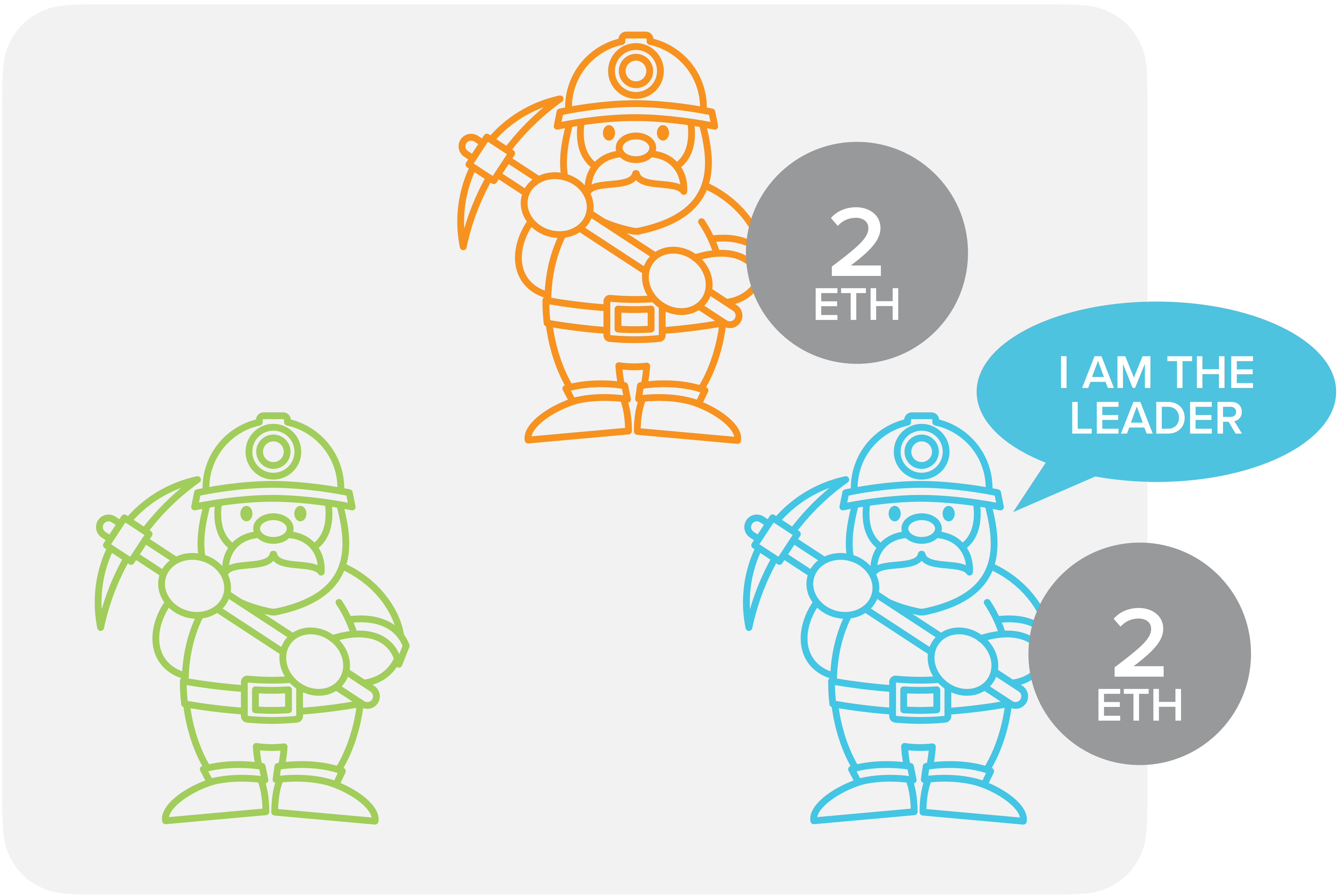
BLOCKCHAIN



How Are Blocks Added to Chain?

BLOCKCHAIN





Open Consensus: How?

PROOF-OF-WORK

First party to solve puzzle creates next block

- sybil resistant selection of a random party

Problems:

- slow, wastes energy

bitcoin

ethereum

PROOF-OF-STAKE

Fast block creation

No energy waste

But more complex

ethereum

Tendermint

DFINITY

celo

Algorand™

PROOF-OF-SPACE

chia

Filecoin

MANY MORE IDEAS

SOLANA

LAYER 1.5: the Blockchain Computer

APP logic is encoded in a program that runs on blockchain

- Rules are enforced by a public program (public source code)
  - ➔ **transparency:** no single trusted 3rd party
- The APP program is executed by parties who create new blocks
  - ➔ **public verifiability:** anyone can verify state transitions

LAYER 1.5	**Compute Layer** (blockchain computer)
LAYER 1	**Consensus Layer**

Running Programs on a Blockchain (APPs)

a16z

Execution Environment

BITCOIN SCRIPT

A LIMITED COMPUTING ENVIRONMENT

- Limited instruction set (no loops)
- Sufficient for some tasks:
  - atomic swaps,
  - payment channels, …

ETHEREUM

GENERAL PROGRAMMING ENVIRONMENT (SOLIDITY, WEB3)

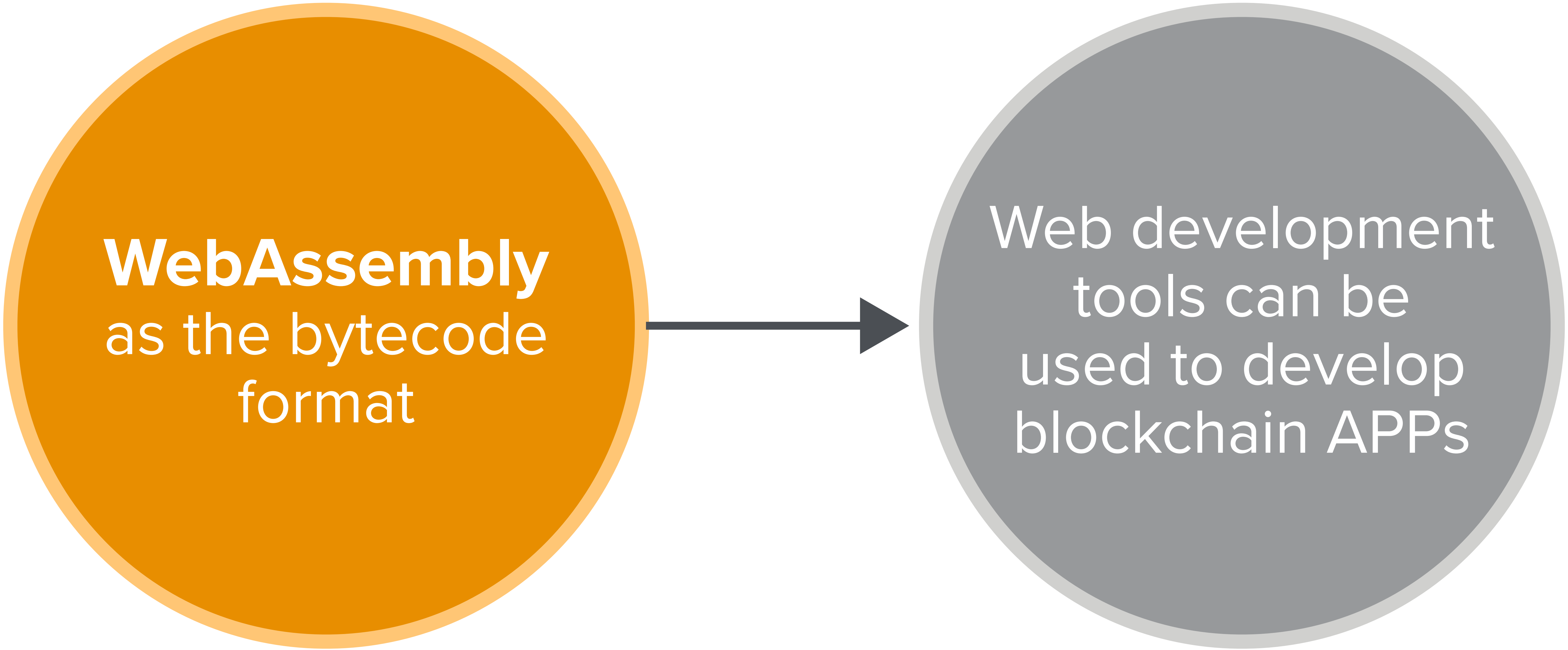
- EVM is a general purpose computing environment
- APP code updates internal state in response to transactions
- Calling APP costs fees (gas)
  - prevents DoS on miners
  - **storing on-chain state costs fees**

General Execution Environments

Recent projects

DFINITY

NEAR



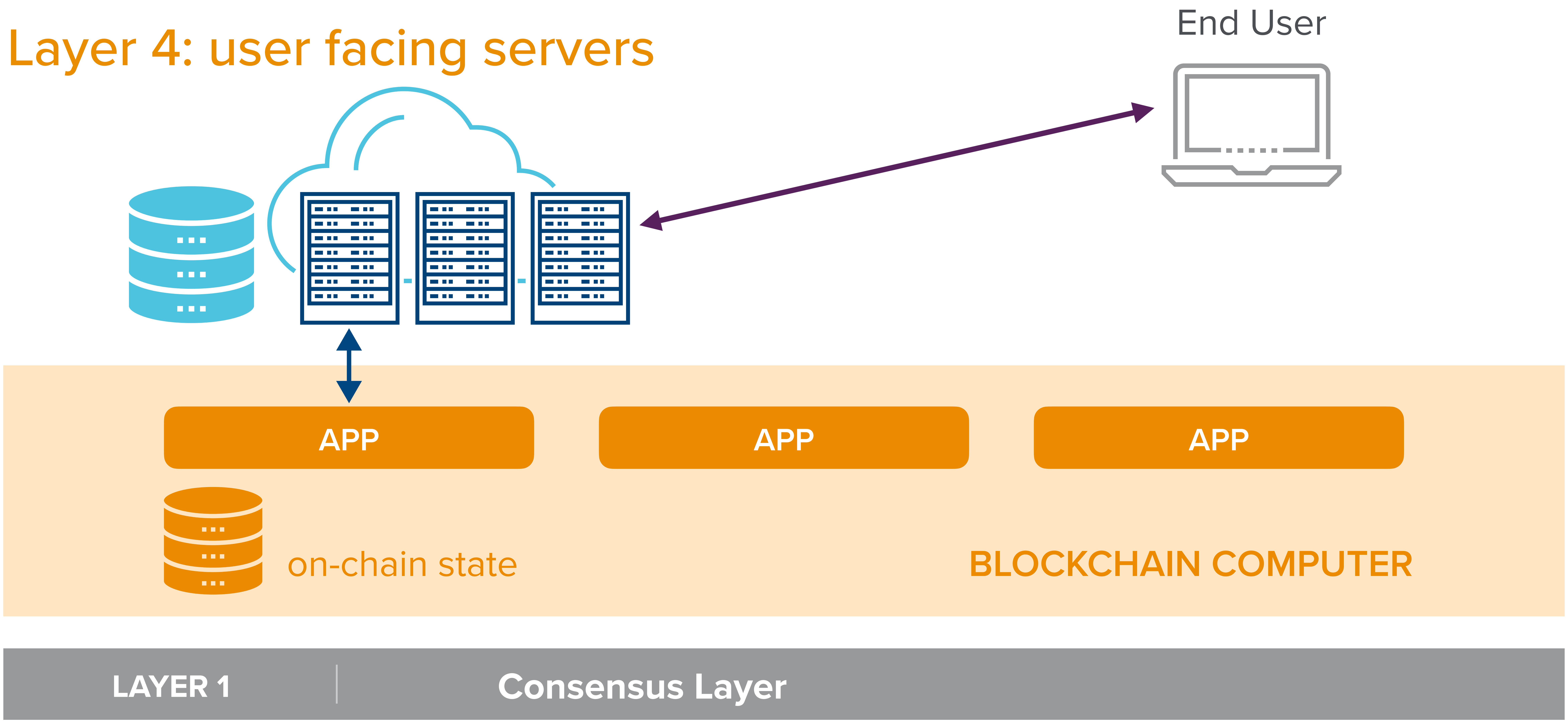
Decentralized Applications (APPs)

Run on blockchain computer

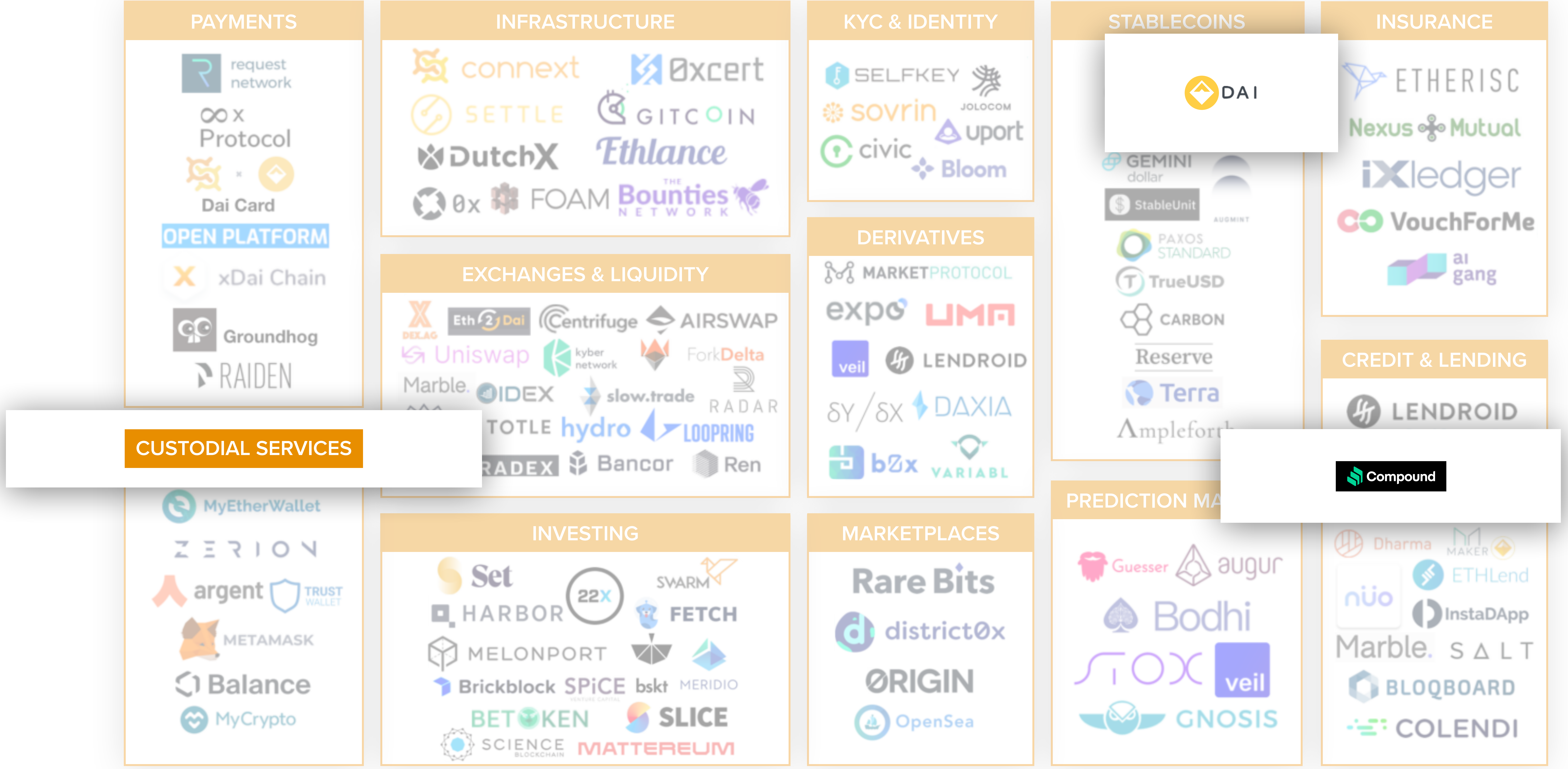
LAYER 2	**Applications** (Solidity, Move, Motoko)
LAYER 1.5	**Compute Layer** (blockchain computer)
LAYER 1	**Consensus Layer**

Common APP Architecture

Layer 4: user facing servers



Ethereum's DeFi



Source: https://www.theblockcrypto.com/genesis/15376/mapping-out-ethereums-defi

a16z

Detailed APPs in Coming Lectures...

Cryptographic Primitives

Blockchain Crypto Primitives

Blockchains are a consumer of advanced cryptographic primitives

Digital signatures + aggregation

Merkle commitments

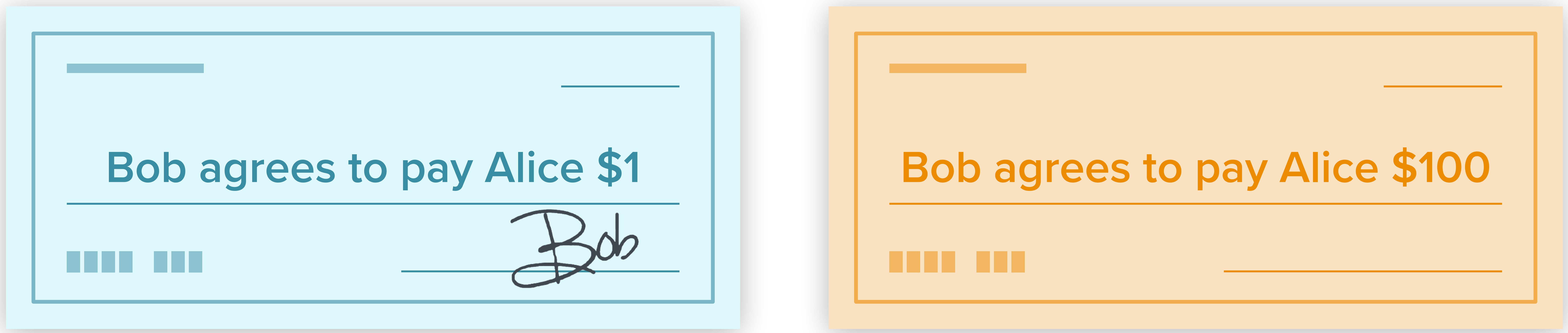
Succinct Zero-Knowledge proof systems

Important Primitives

Digital Signatures

Physical Signatures

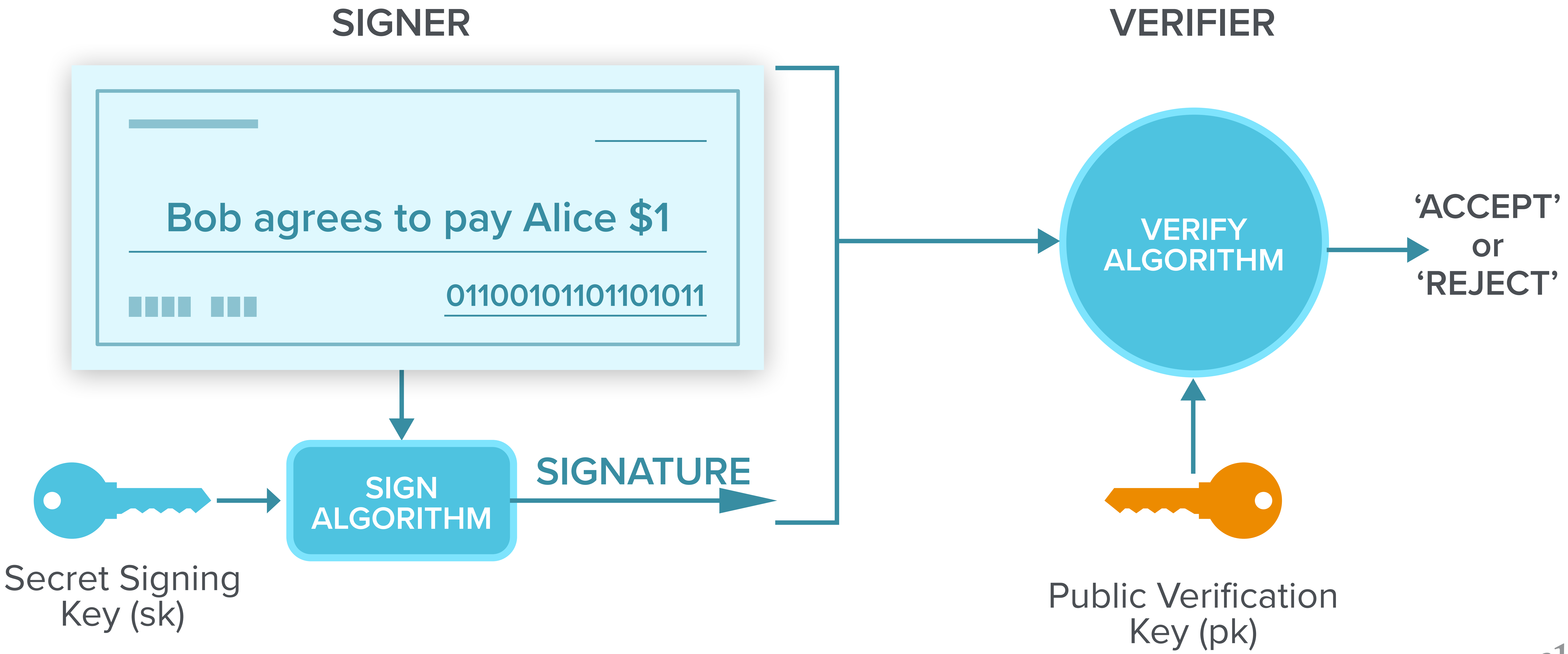
Goal: bind transaction to author



Problem in the digital world…anyone can copy Bob's signature from one doc to another

Digital Signatures

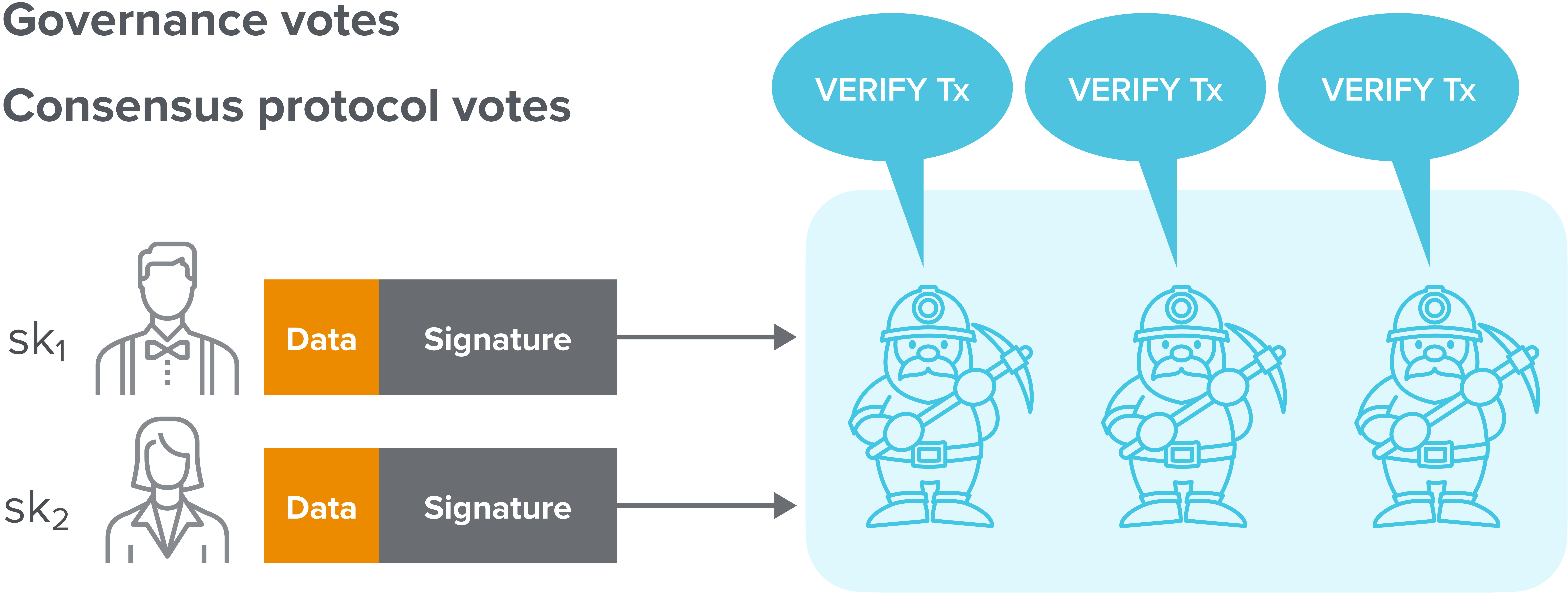
Solution: make signature depend on document



a16z

Signatures on the Blockchain: Used Everywhere

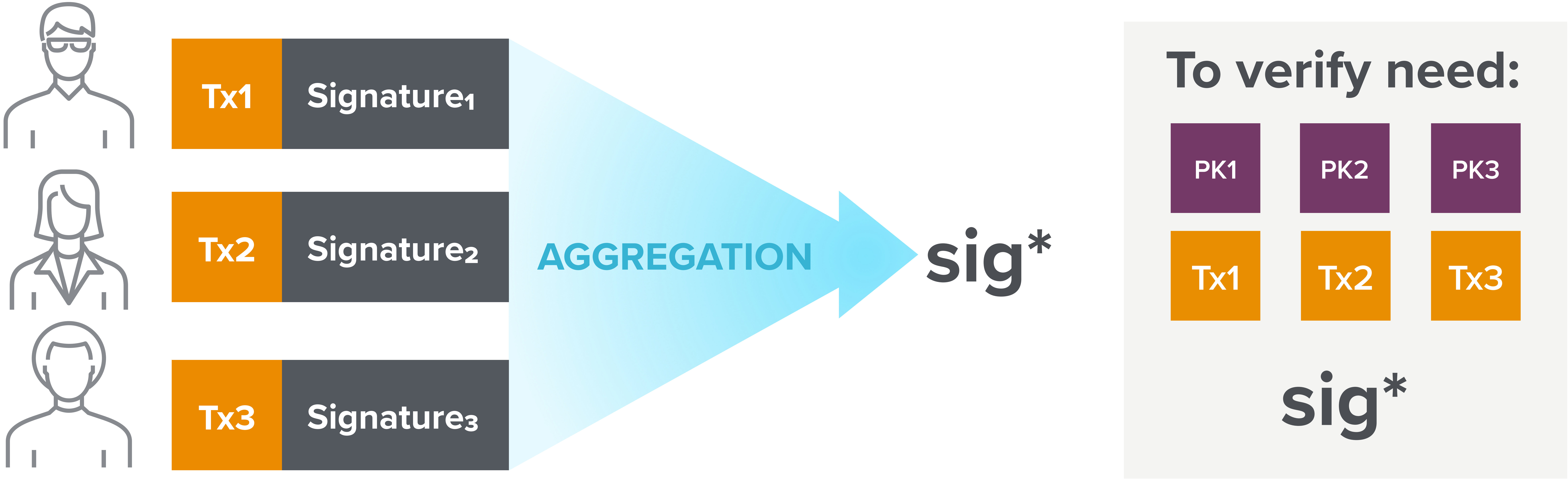
- **Ensure Tx authorization**
- **Governance votes**
- **Consensus protocol votes**



a16z

BLS Signature Aggregation

Anyone can compress n signatures into a single signature



No need to store list of signatures on the blockchain

Aggregation on the Blockchain

Merkle Commitments

Commitments

Cryptographic commitment: emulates an envelope

Many applications: e.g., an APP for a **sealed bid auction**

- Every participant **commits** to its bid,
- Once all bids are in, everyone opens their commitment

a16z

Crypto Commitments

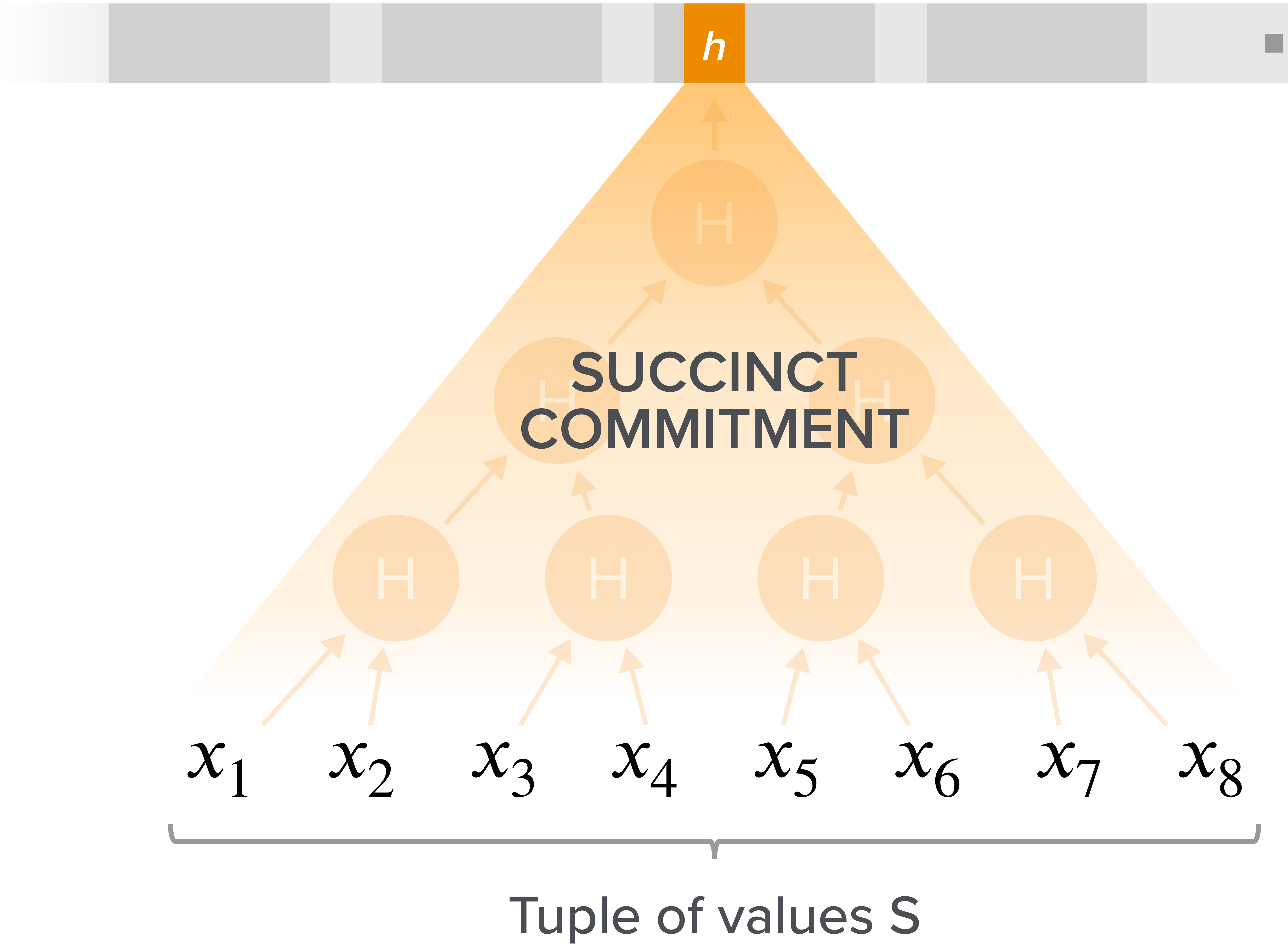
Syntax: a commitment scheme is two algorithms

- Commit (data) $\Longrightarrow$ (com, open)
- Verify (data, com, open) $\Longrightarrow$ ‘accept’ or ‘reject’

Security properties (informal):

- **Binding**: Bob cannot produce two valid openings
- **Hiding**: com reveals nothing about committed data

Committing to a Tuple of Values: Merkle Trees



GOAL:

- Commit to tuple S
- Later, provide a **short** proof that x_4 is the 4th element in S

Proof length = $O(\log |S|)$

Many Applications: **1)** Short Proof of Payment

Bob: has all block hashes

Alice: wants to prove she paid Bob 2ETH

- Alice sends a short Merkle proof to Bob
- 1000 Tx in block → short proof

a16z

2) Keeping State off the Chain

Database of account balances

Alice	10	pk_A
Bob	5	pk_B
Carol	12	pk_C
⋮		⋮
Zoe	8	pk_Z

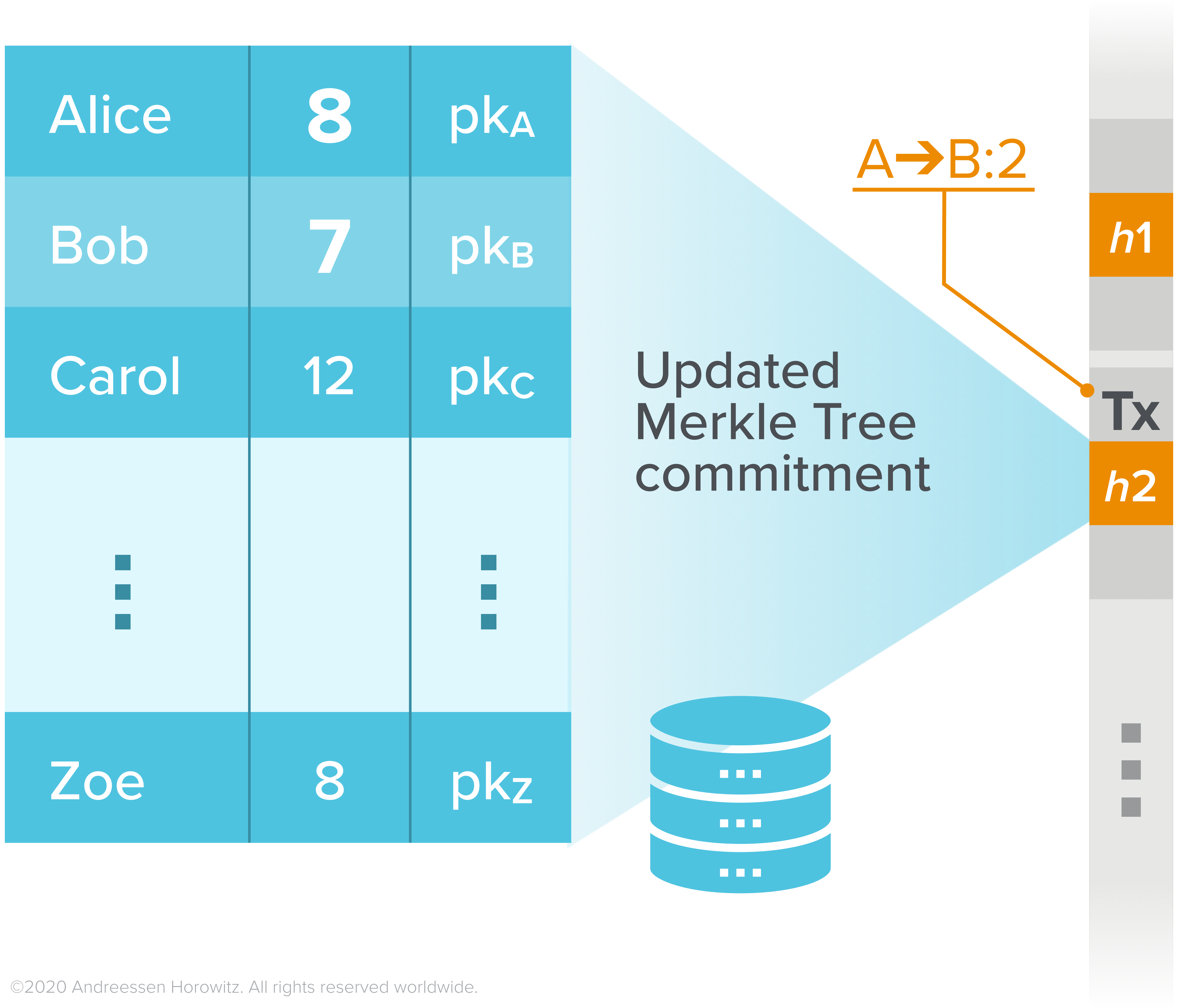
off-chain servers
store balances,
on-chain:
only **short** commitment

Alice can prove her balance (10) to anyone with a **short** proof

a16z

2) Keeping State off the Chain

Database of account balances



off-chain servers
store balances,
on-chain:
only **short** commitment

Alice can prove her balance (10) to anyone with a **short** proof

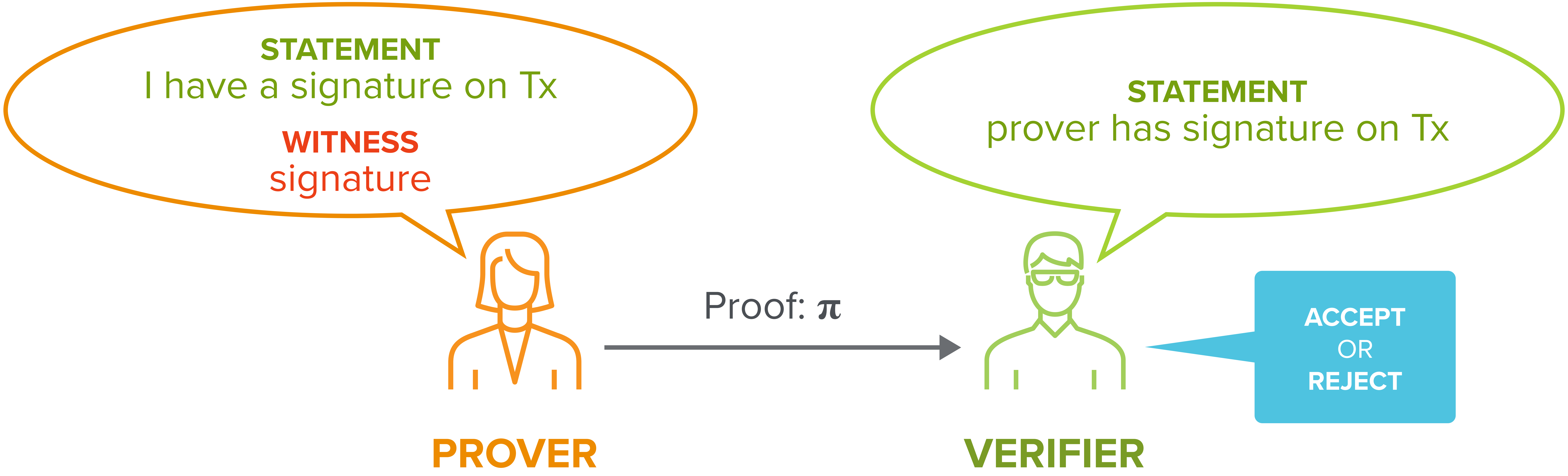
Tx can update committed state. Chain validates Tx.

a16z

Zero Knowledge Proof Systems

What Is a Proof System? (Informal)

GOAL: prover wants to convince a **verifier** that a **statement** is true



What is a statement: **program**(**statement**, **witness**) → ‘0’ or ‘1’

Properties of a Proof System

Complete: if statement is true, prover can convince verifier

Succinct proof: proof is short (logarithmic in statement size)

Fast verification: verification is fast (logarithmic in statement size)

Efficient proof generation: generating the proof takes linear time

SNARK

SECURITY:

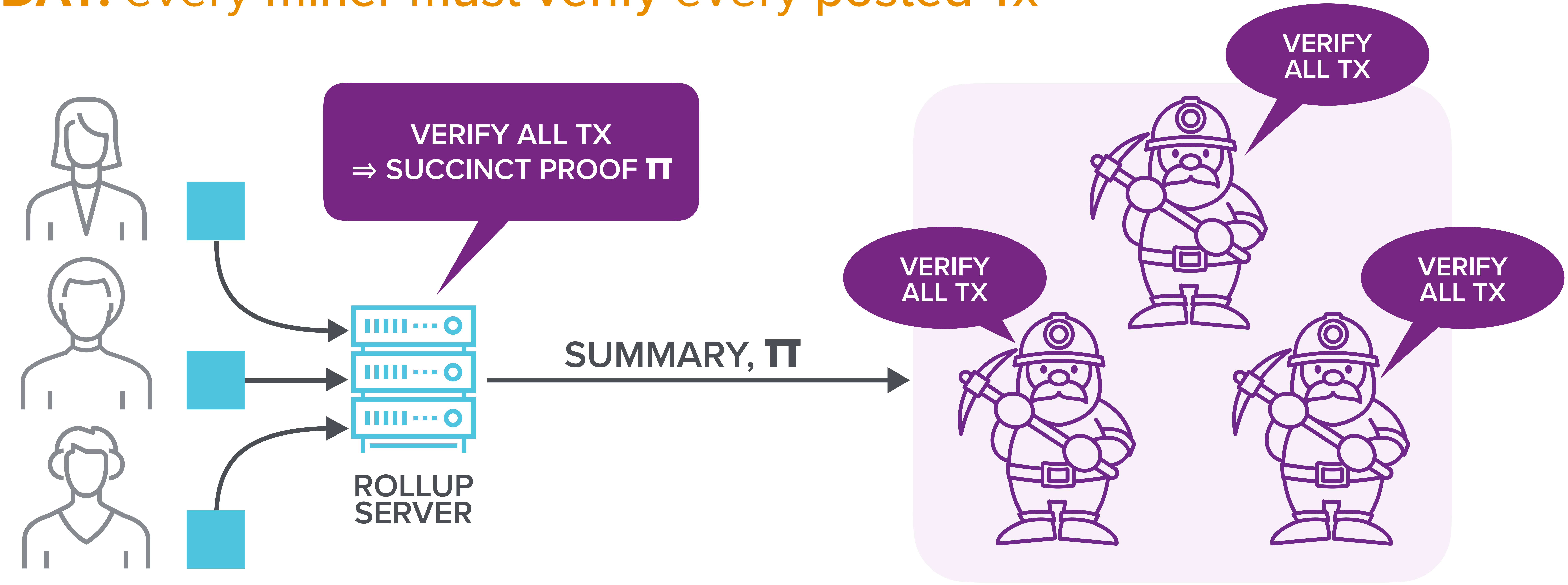
Sound: prover cannot convince verifier of a false statement

Zero knowledge (optional): verifier learns nothing about the witness

zkSNARK

Application 1: Scalability (Rollup) … no ZK

TODAY: every miner must verify every posted Tx



Application 2: Private Data on a Public Blockchain

TODAY: Data on blockchain is public ➔
Businesses cannot use for private B2B applications

PUBLIC VERIFIABILITY

Tx1 | Tx2 | Tx3 | ■ ■ ■

A different approach: only post **hiding commitments** to blockchain

com(APP code), com($state_0$)	com(Tx_1), com($state_1$) ZKP_1	com(Tx_2), com($state_2$) ZKP_2

ZKP: ZK proof that $\underbrace{[state_i,\ Tx_i \rightarrow State_{i+1}]}_{\text{state transition}}$ is consistent with APP code

Final Thoughts

When To Use a Blockchain?

BLOCKCHAIN ≠ DATABASE

Always ask: why not use a centralized system?

- **Blockchain positives:**
  - used when there is no single party trusted by everyone
- **Negatives:** slower and more complex than a centralized system

Trends

The End

Excited To See Your Blockchain Apps!!