CS251 Fall 2022

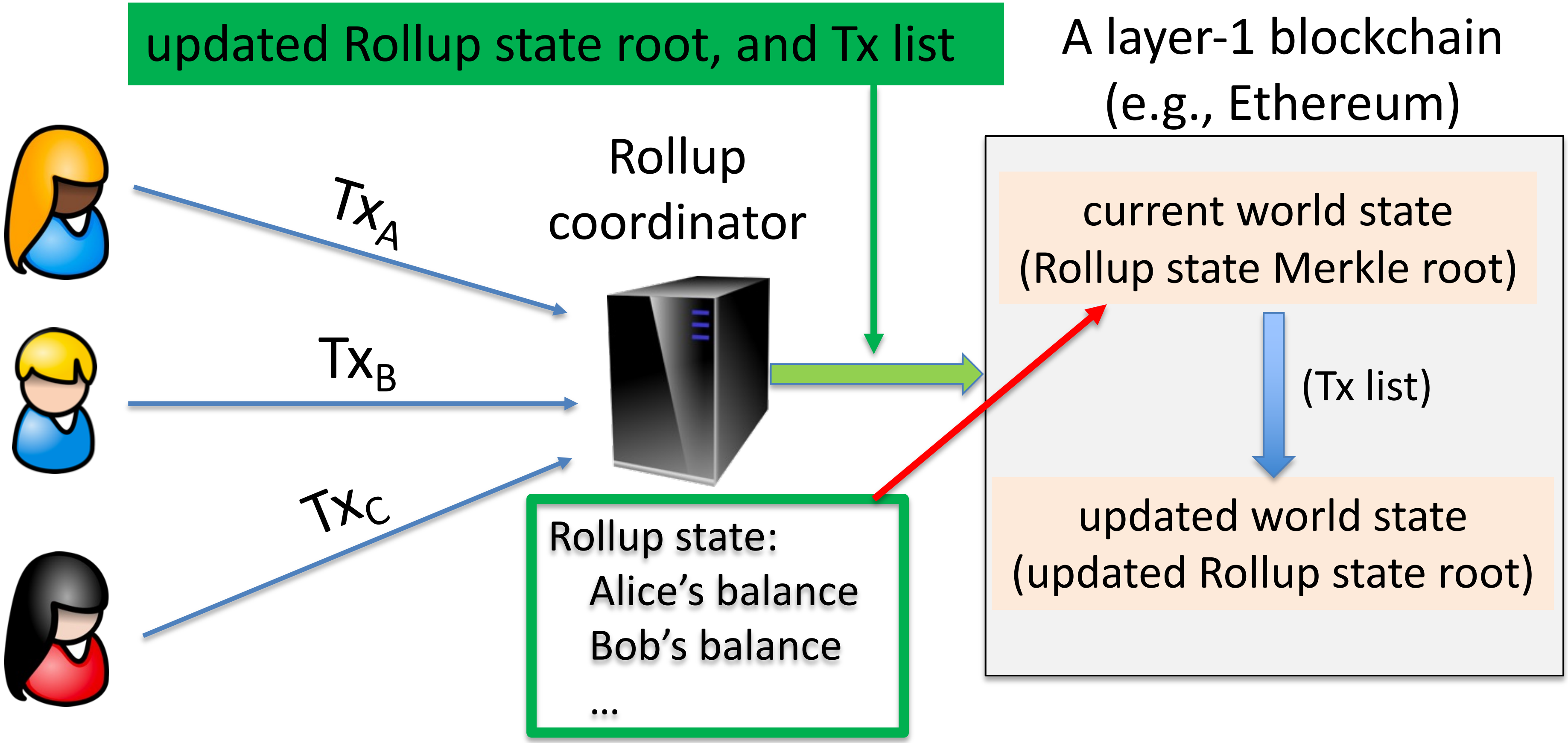
(cs251.stanford.edu)

Recursive SNARKs

Dan Boneh

… but first, more on Rollups

Review: Rollup core idea



updated Rollup state root, and Tx list

A layer-1 blockchain (e.g., Ethereum)

Tx_A

Tx_B

Tx_C

Rollup coordinator

Rollup state:
Alice's balance
Bob's balance
…

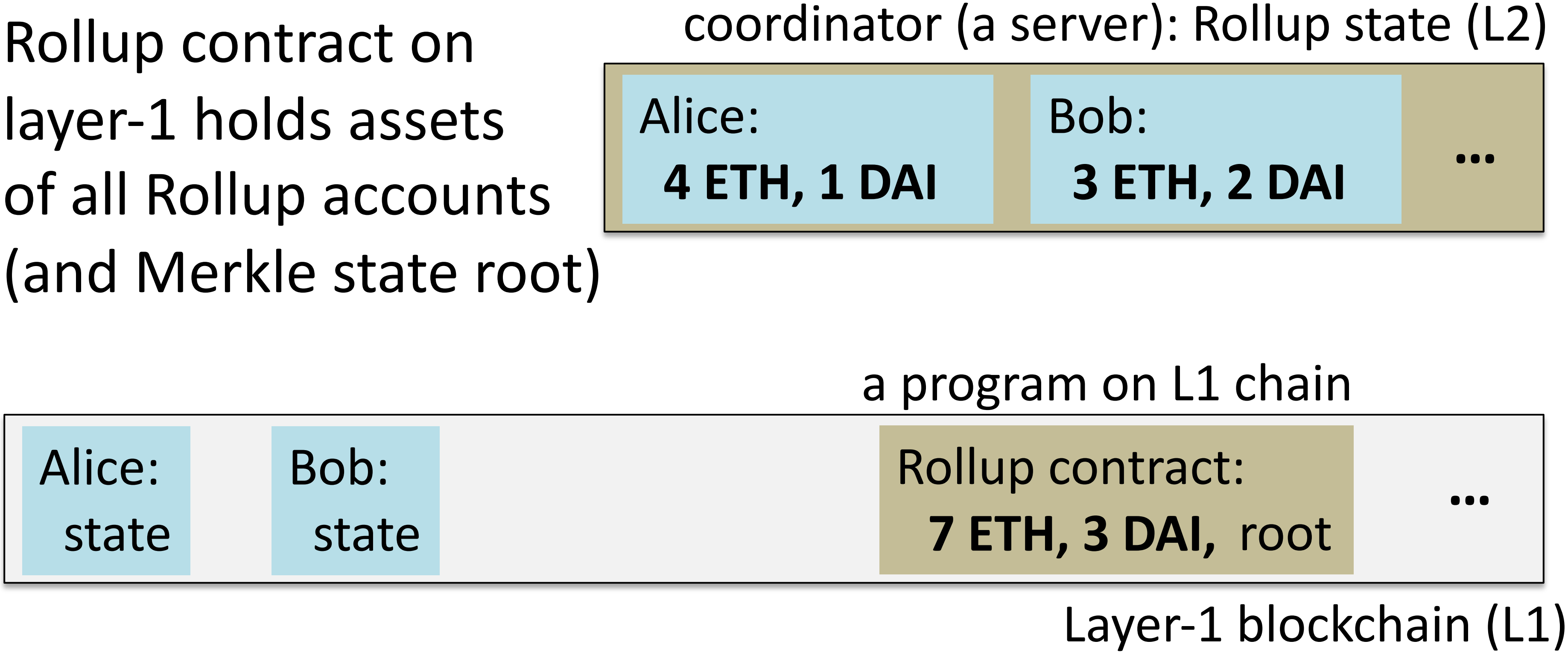
current world state
(Rollup state Merkle root)

(Tx list)

updated world state
(updated Rollup state root)

The two parts of Rollup

Rollup contract on layer-1 holds assets of all Rollup accounts (and Merkle state root)



coordinator (a server): Rollup state (L2)

Alice: **4 ETH, 1 DAI**

Bob: **3 ETH, 2 DAI**

…

a program on L1 chain

Alice: state

Bob: state

Rollup contract: **7 ETH, 3 DAI,** root

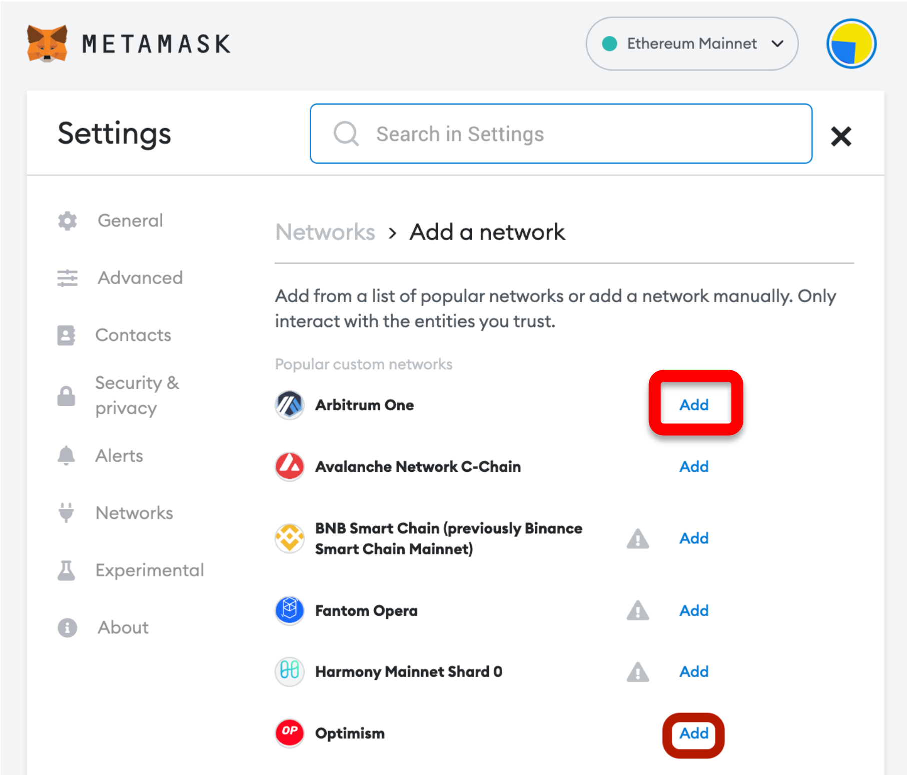
…

Layer-1 blockchain (L1)

How to send Tx to the coordinator

Enduser configures its wallet to send Tx to the RPC points of the selected Rollup.

(by default Metamask sends Tx to the Ethereum Mainnet RPC points)



Networks › Add a network

Add from a list of popular networks or add a network manually. Only interact with the entities you trust.

Popular custom networks

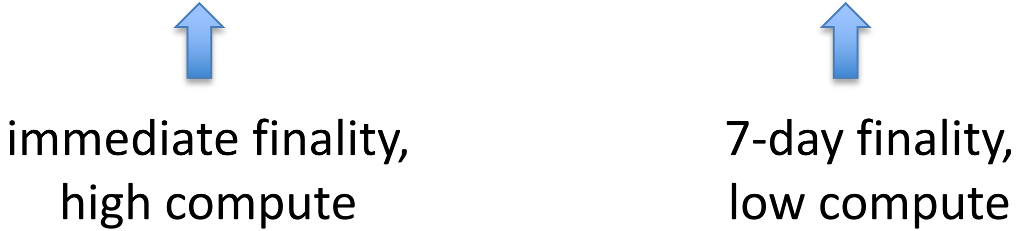
Arbitrum One	Add
Avalanche Network C-Chain	Add
BNB Smart Chain (previously Binance Smart Chain Mainnet)	Add
Fantom Opera	Add
Harmony Mainnet Shard 0	Add
Optimism	Add

Review: difficulties …

Problem 1: what if coordinator is dishonest?

- It could steal funds from the Rollup contract
- It could issue fake Tx on behalf of users

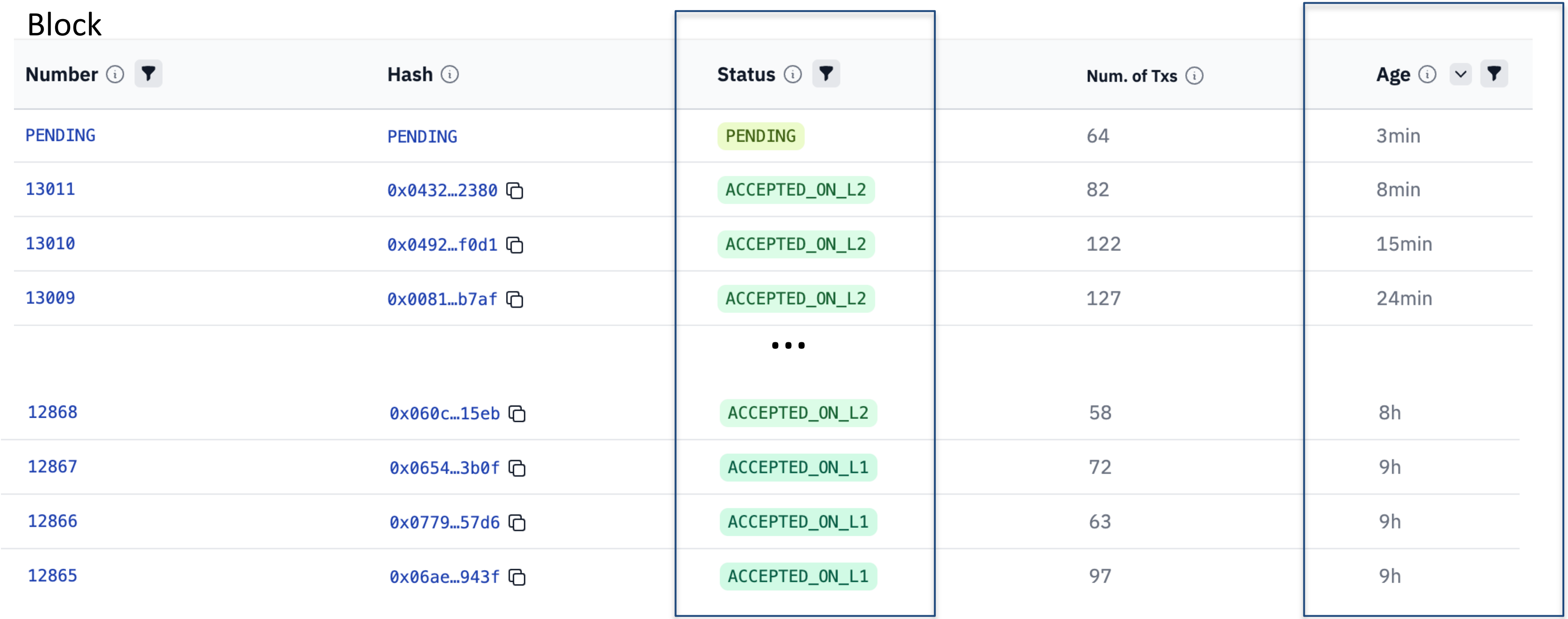
⇒ solution: validity proofs (zk-Rollup) or fraud proofs (opt. Rollup)



immediate finality,
high compute

7-day finality,
low compute

An example (StarkNet -- using validity proofs)



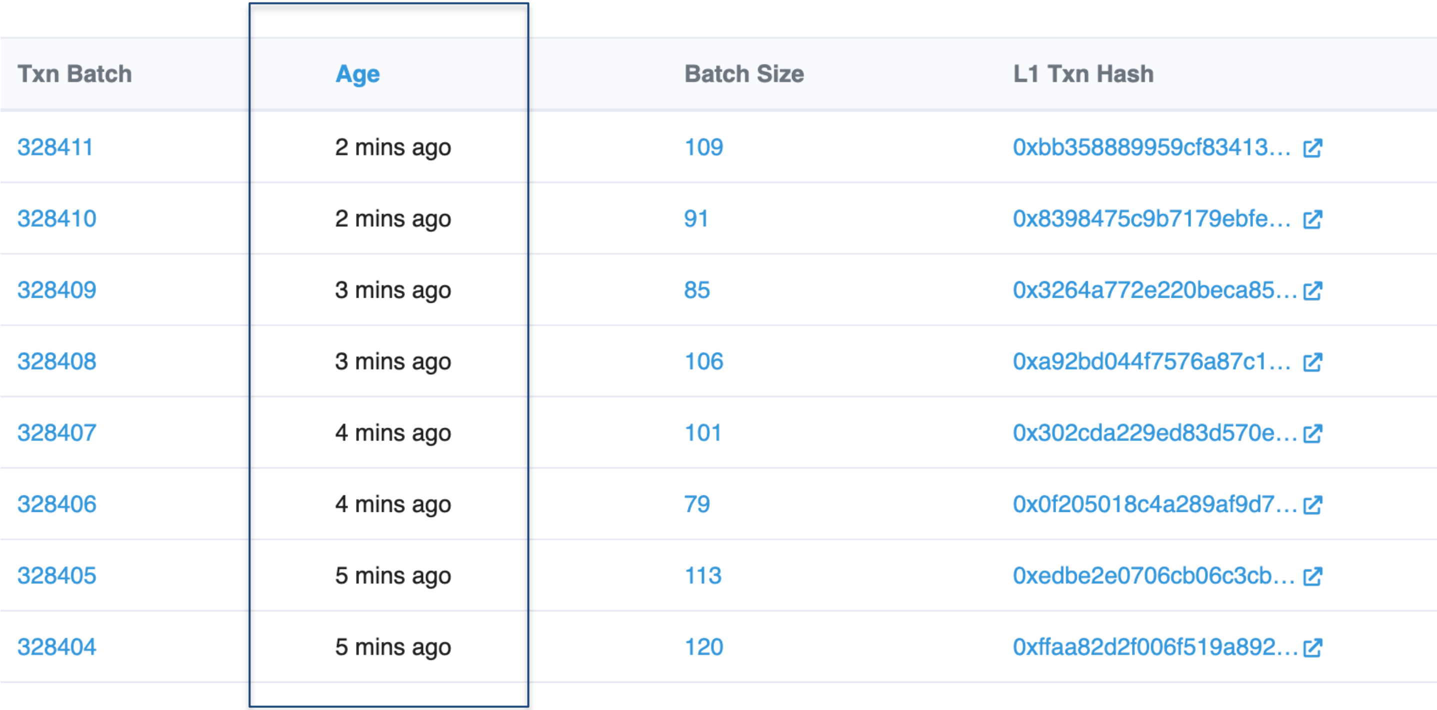
Block

Number	Hash	Status	Num. of Txs	Age
PENDING	PENDING	PENDING	64	3min
13011	0x0432…2380	ACCEPTED_ON_L2	82	8min
13010	0x0492…f0d1	ACCEPTED_ON_L2	122	15min
13009	0x0081…b7af	ACCEPTED_ON_L2	127	24min
		•••		
12868	0x060c…15eb	ACCEPTED_ON_L2	58	8h
12867	0x0654…3b0f	ACCEPTED_ON_L1	72	9h
12866	0x0779…57d6	ACCEPTED_ON_L1	63	9h
12865	0x06ae…943f	ACCEPTED_ON_L1	97	9h

Tx posted on L1 (Ethereum) about every eight hours

Source: starkscan.co

An example (Optimism -- using fraud proofs)



Txn Batch	Age	Batch Size	L1 Txn Hash
328411	2 mins ago	109	0xbb358889959cf83413…
328410	2 mins ago	91	0x8398475c9b7179ebfe…
328409	3 mins ago	85	0x3264a772e220beca85…
328408	3 mins ago	106	0xa92bd044f7576a87c1…
328407	4 mins ago	101	0x302cda229ed83d570e…
328406	4 mins ago	79	0x0f205018c4a289af9d7…
328405	5 mins ago	113	0xedbe2e0706cb06c3cb…
328404	5 mins ago	120	0xffaa82d2f006f519a892…

Shows batch posted on L1 (Ethereum)

Source: optimistic.etherscan.io

Review: difficulties …

Problem 1: what if coordinator is dishonest?

- It could steal funds from the Rollup contract
- It could issue fake Tx on behalf of users

⇒ solution: validity proofs (zk-Rollup) or fraud proofs (opt. Rollup)

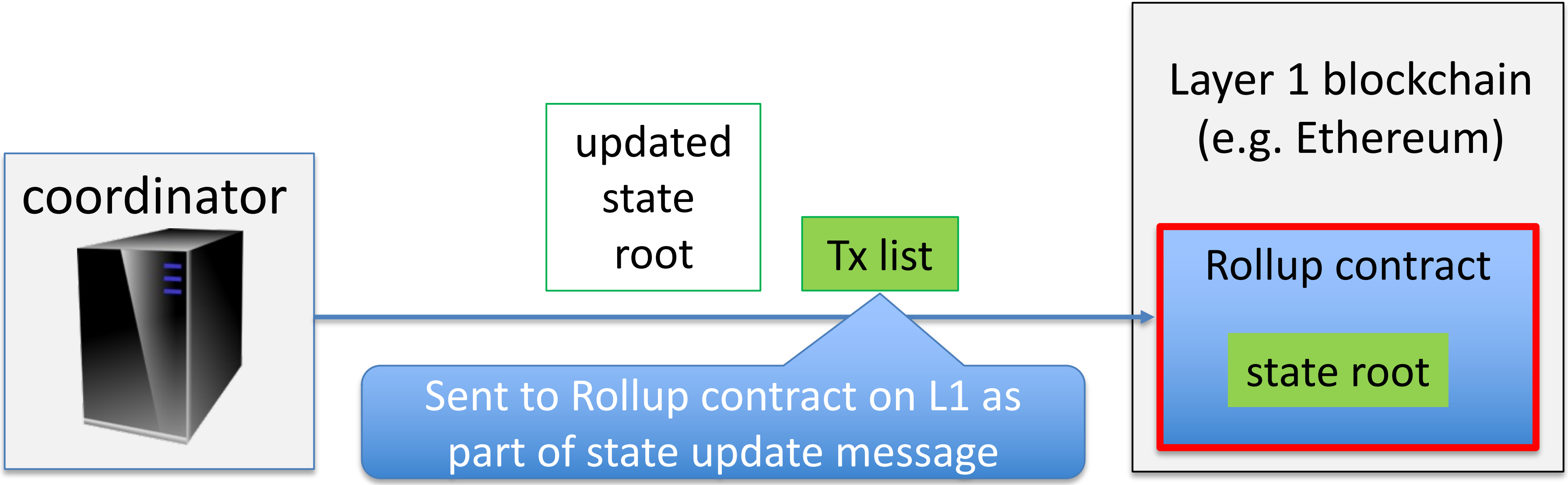
Problem 2: what if coordinator stops providing service?

- If Rollup state is lost, how can we initialize a new coordinator?

Ensuring Rollup state is always available

The definition of a Rollup:

Rollup state can always be reconstructed from data on the L1 chain



Sent to Rollup contract on L1 as part of state update message

Ensuring Rollup state is always available

To reconstruct current Rollup state:

- Read all Rollup update messages and re-execute Tx.

  ⇒ anyone can become a coordinator

- Rollups use L1 for data storage

… but note EIP-4444

What to store?

- For zk-Rollup: send Tx summary to L1, without signatures

  (SNARK proof proves validity of signatures)

- For optimistic: need to send Tx summary *and* signatures to L1

Ensuring Rollup state is always available

The downside: **expensive**

- Tx list is sent as calldata: 16 gas per non-zero byte

  (EIP-4488 aims to support Rollups by reducing to 3 gas/byte)

In practice:

- Optimistic Rollups fee/Tx: 3-8 times lower than Ethereum L1
- zk-Rollup fee/Tx: 40-100 times lower than Ethereum L1

Can we do even better?

Data Availability Committee (DAC)

To further reduce Tx fees:

- **Store L2 state root** (small) on the L1 chain
- **Store Tx data** (large) with a Data Availability Committee (**DAC**):
  - comprises a set of nodes trusted to keep the data available
  - cheaper than storage on L1
  - L1 accepts an update only if <u>all</u> DAC members sign it

    ⇒ ensures that all DAC members accepted Tx data

Setting up a new coordinator depends on availability of the DAC

Validium

Validium: an L2 using a DAC and validity proofs (SNARKs)

- Well suited for lower value assets.
- Potential privacy benefits … only DAC members see Tx data

An example: StarkEx uses a **five** member DAC

- Users can choose between Validium or Rollup modes

(Tx data off-L1-chain vs. Tx data on-L1-chain)

cheaper Tx fees,
but only secure as DAC

More expensive Tx,
but same as L1 security

Summary: types of L2

Scaling the blockchain: Payment channels and Rollups (L2 scaling)

security

availability

	SNARK validity proofs	**Fraud proofs**
Tx data on L1 chain	**zkRollup**	optimistic Rollup, 7-day finality
Tx data in a DAC	Validium (reduced fees, but higher risk)	”Plasma”

Volume of some L2 systems

	Tx Volume/day	average fee/tx	(on Nov. 15, 2022)
Ethereum:	1013K Tx	**2.71 USD/Tx**	
Arbitrum:	345K Tx	0.08 USD/Tx	(optimistic Rollup)
Optimism:	303K Tx	0.13 USD/Tx	(optimistic Rollup)
StarkNet:	14K Tx	0.22 USD/Tx	(zkRollup)

Can coordinator censor a Tx?

What if coordinators refuse to process my Tx?

What to do? One option:

- enduser can post Tx directly to the L1 Rollup contract
- The L1 Rollup contract will then refuse to accept updates from a coordinator until an update includes that Tx

$\Rightarrow$ censorship will cause the entire Rollup to freeze

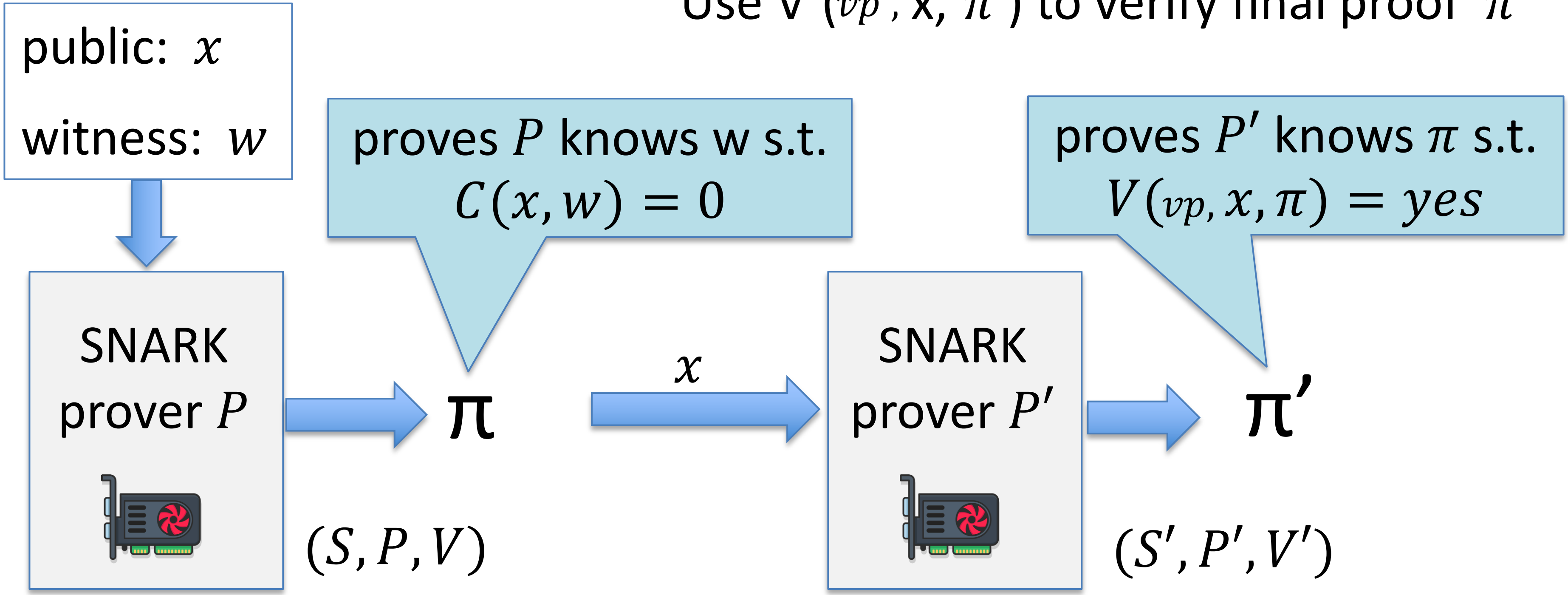
SNARK recursion

Layer 3 and beyond …

SNARK recursion

Two level recursion: **proving knowledge of a proof**

Use V’(vp', x, π’) to verify final proof π’



public: x

witness: w

SNARK prover P

π

proves P knows w s.t. $C(x, w) = 0$

(S, P, V)

x

SNARK prover P'

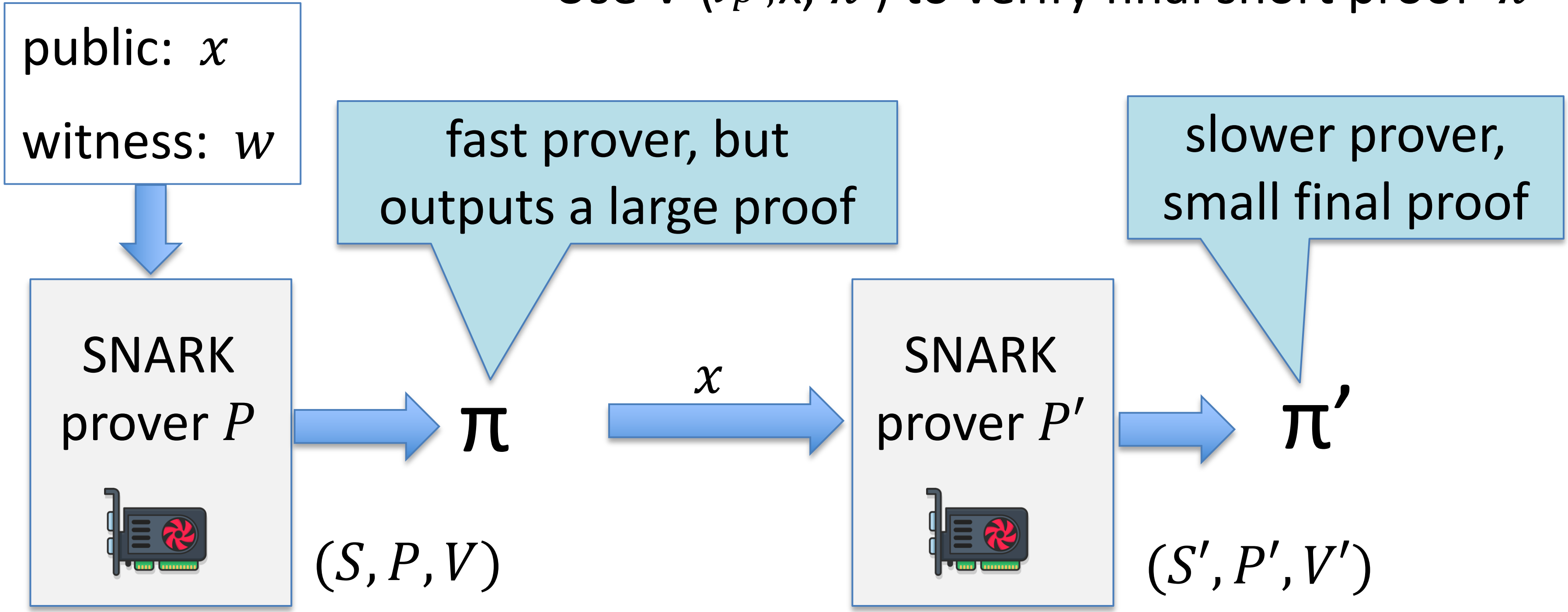
π’

proves P' knows π s.t. $V(vp, x, \pi) = yes$

(S', P', V')

Application 1: proof compression

Use V’(vp',x, π’) to verify final short proof π’



public: x

witness: w

SNARK prover P

fast prover, but outputs a large proof

π

x

SNARK prover P'

slower prover, small final proof

π’

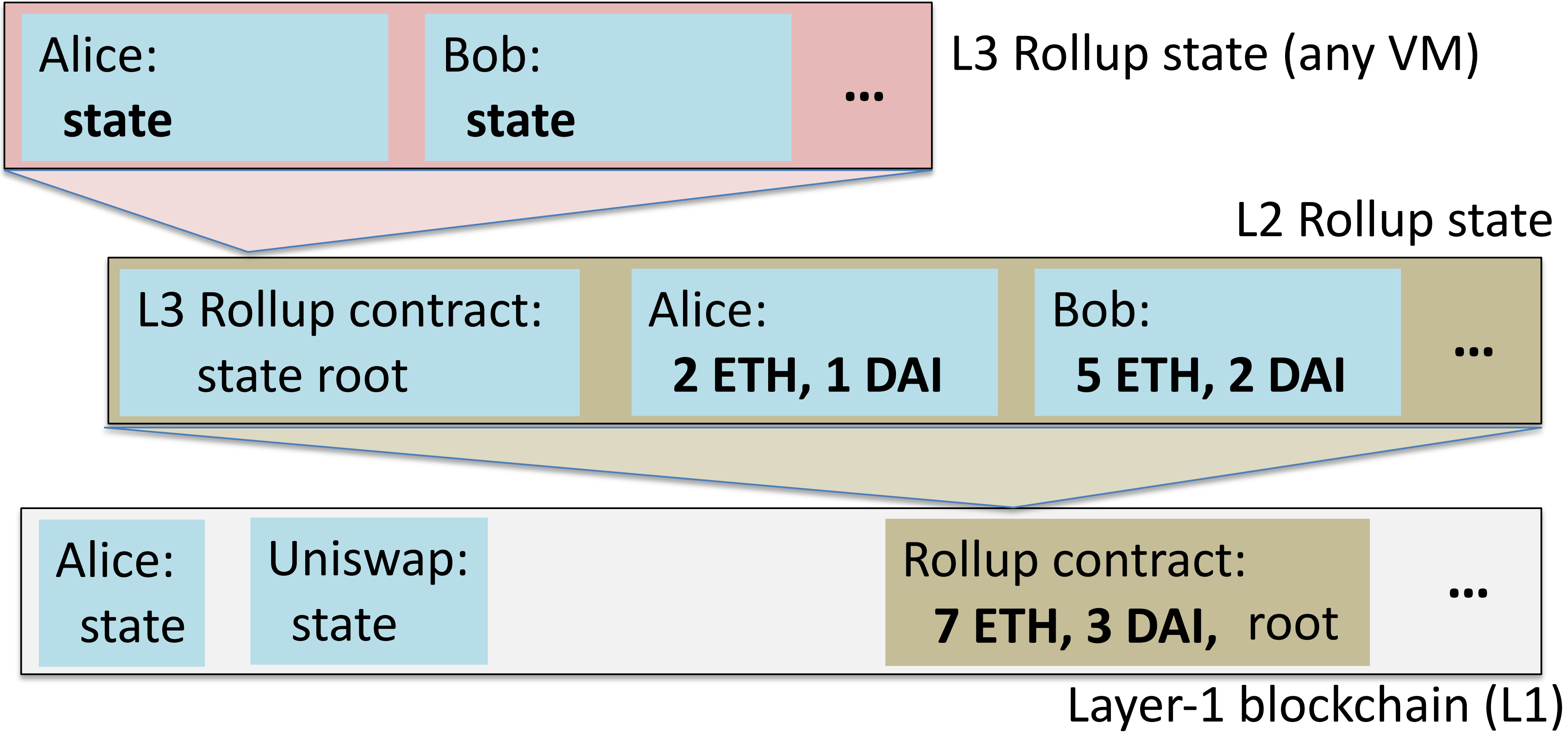
(S, P, V)

(S', P', V')

prove $C(x, w) = 0$

prove $V(vp, x, \pi) = yes$

Application 2: Layer three and beyond



Layer three and beyond

One L2 coordinator can support many L3s

- each L3 can run a custom VM with its own features
- L3 chains can communicate with each other through L2

Each L3 coordinator submits Tx list and SNARK proof to L2

- L2 coordinator: collects batch of proofs,
  - builds a proof π that it has a batch of valid proofs, and
  - submits the single proof π and updated root to L1 chain.

$\Rightarrow$ Scaling factor 100 × 100

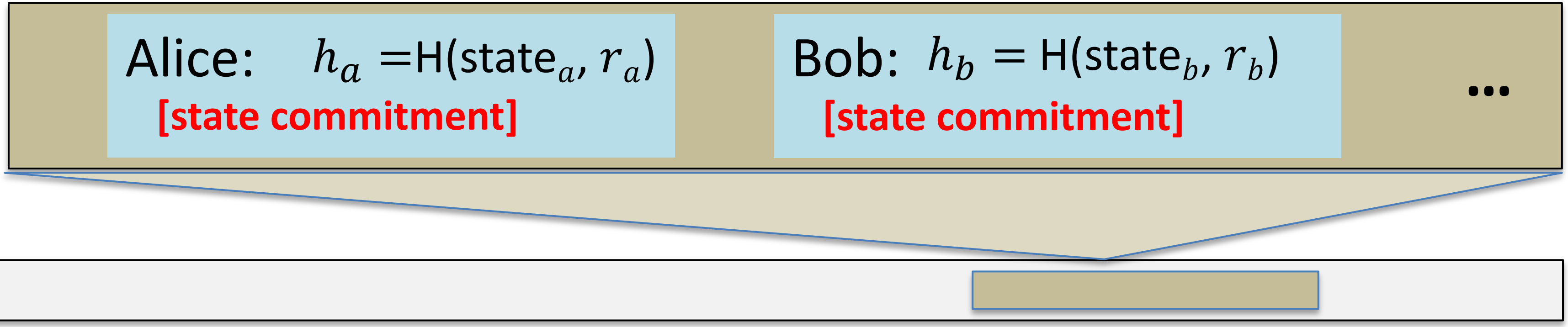
Application 3: L2 with private Tx (simplified)

Only Alice knows her own state$_a$ and r_a .

- Coordinator does not know account balances

  (only Alice knows her committed account balances)

L2 Rollup state: hidden balances



Alice: h_a =H(state$_a$, r_a)
[state commitment]

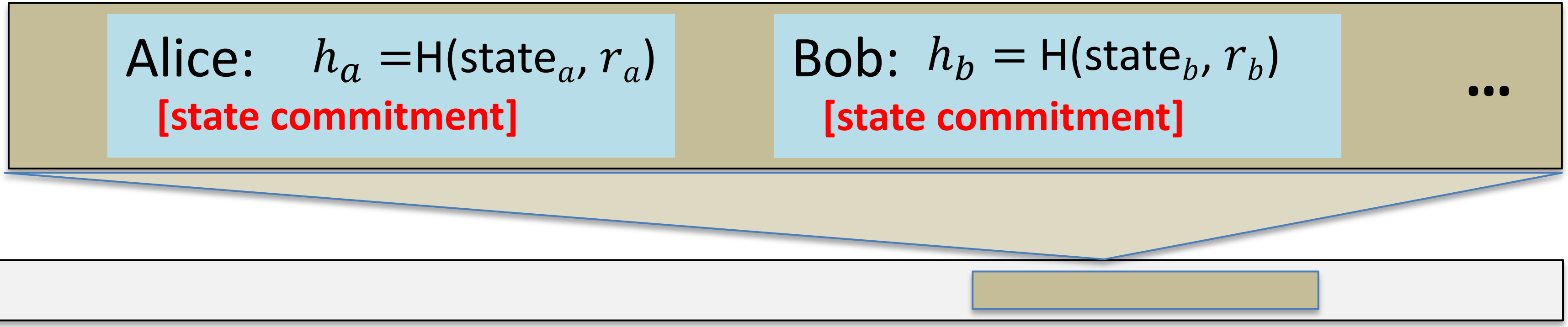
Bob: h_b = H(state$_b$, r_b)
[state commitment]

…

Application 3: L2 with private Tx (simplified)

Alice want to pay Bob 2 ETH: Tx: [A → B: 2 ETH, sig_A]

- compute updated state'$_a$ and send Tx to Bob (privately)
- choose random r'_a and set $h'_a \leftarrow$ H(state'$_a$, r'_a)
- build proof π_a that h'_a is a valid update to Alice's state
- Send (h'_a, π_a) to L2 coordinator



Alice: h_a =H(state$_a$, r_a)

[state commitment]

Bob: h_b = H(state$_b$, r_b)

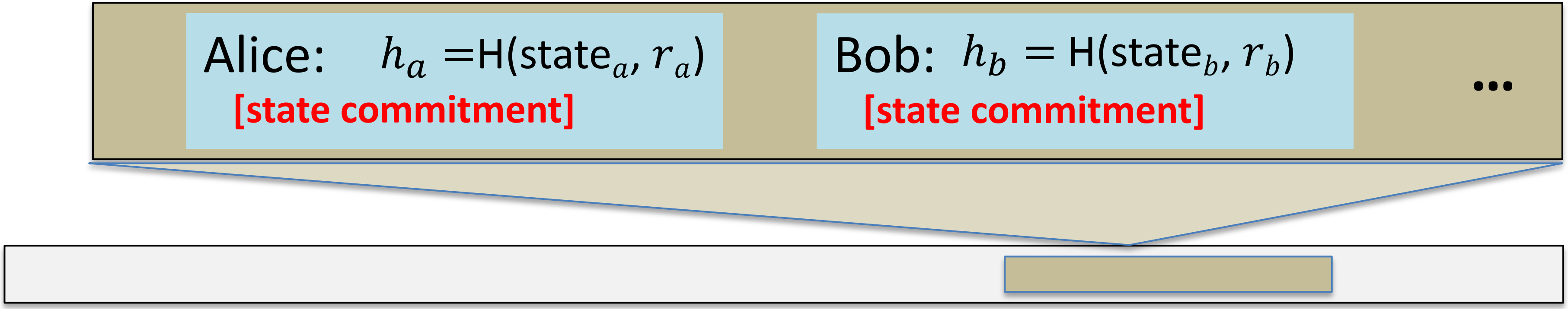
[state commitment]

…

Application 3: L2 with private Tx (simplified)

Bob receives $Tx =$ [A → B: 2 ETH, sig_A] from Alice

- compute updated state'$_{\mathrm{b}}$
- choose random r_b' and set $h_b' \leftarrow$ H(state'$_b$, r_b')
- build proof π_b that h_b' is a valid update to Bob's state
- Send (h_b', π_b) to L2 coordinator



Alice: h_a =H(state$_a$, r_a)

[state commitment]

Bob: h_b = H(state$_b$, r_b)

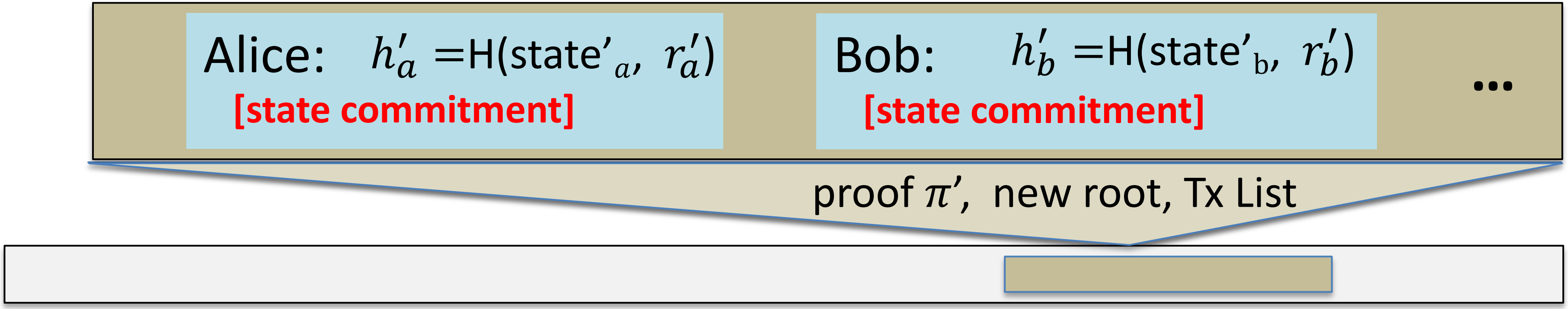
[state commitment]

…

Application 3: L2 with private Tx (simplified)

Collect a batch of transactions from users $\{(h_i', \pi_i)\}$:

- Update Merkle leaves to new committed states
- build a proof π’ that it has a batch of valid proofs for a consistent set of transactions, and
- submit a single proof π’ and updated root to L1 chain.



Alice: h_a' =H(state’$_a$, r_a')

[state commitment]

Bob: h_b' =H(state’$_b$, r_b')

[state commitment]

…

proof π', new root, Tx List

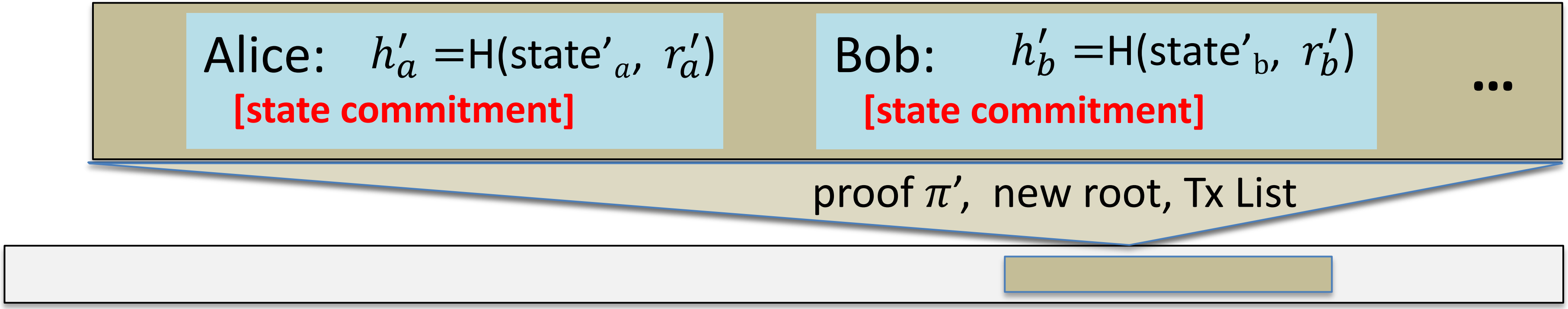
Application 3: main point

Only Alice knows her balance. Only Bob knows his balance.

... they can transact without revealing amouts

... also transact with a public contract (public code and state).

Note: as described, no privacy for Alice when withdrawing from L2



Alice: h'_a =H(state'$_a$, r'_a)
[state commitment]

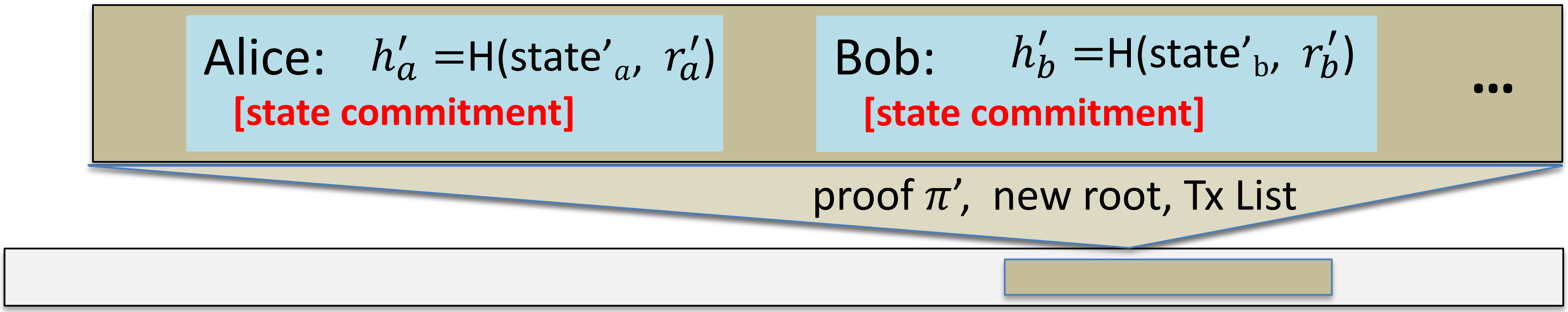
Bob: h'_b =H(state'$_b$, r'_b)
[state commitment]

...

proof π', new root, Tx List

Application 3: main point

Danger: if Alice loses here r_a , she loses access to her funds on L2



Alice: h'_a =H(state'$_a$, r'_a)

[state commitment]

Bob: h'_b =H(state'$_b$, r'_b)

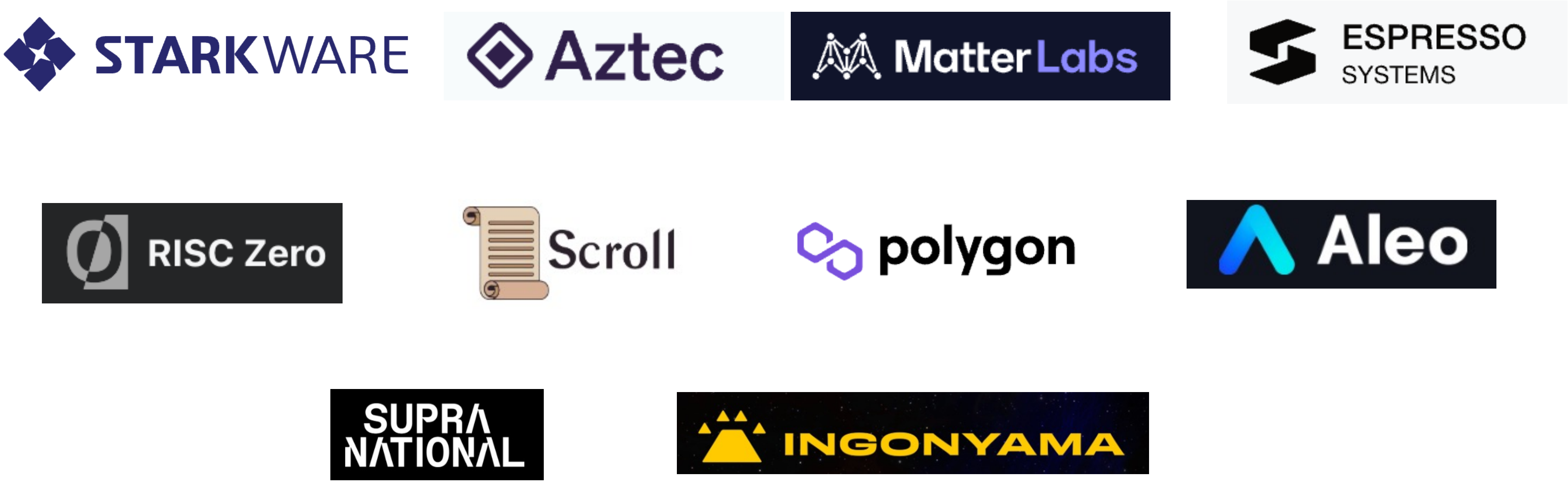
[state commitment]

…

proof π', new root, Tx List

Final ZK topics

Commercial interest in SNARKs



Many more building applications on top …

Why so much commercial interest?

Babai-Fortnow-Levin-Szegedy 1991:

In this setup, a single reliable PC can monitor the operation of a herd of supercomputers working with unreliable software.

“Checking Computations in Polylogarithmic Time”

Why so much commercial interest?

Babai-Fortnow-Levin-Szegedy 1991:

a slow and expensive computer

In this setup, ~~a single reliable PC~~ can monitor the operation of a herd of ~~supercomputers~~ working with unreliable software.

coordinators

“Checking Computations in Polylogarithmic Time”

Why so much commercial interest?

Babai-Fortnow-Levin-Szegedy 1991:

an L1 blockchain

In this setup, ~~a single reliable PC~~ can monitor the operation of a herd of ~~supercomputers~~ working with unreliable software.

coordinators

“Checking Computations in Polylogarithmic Time”

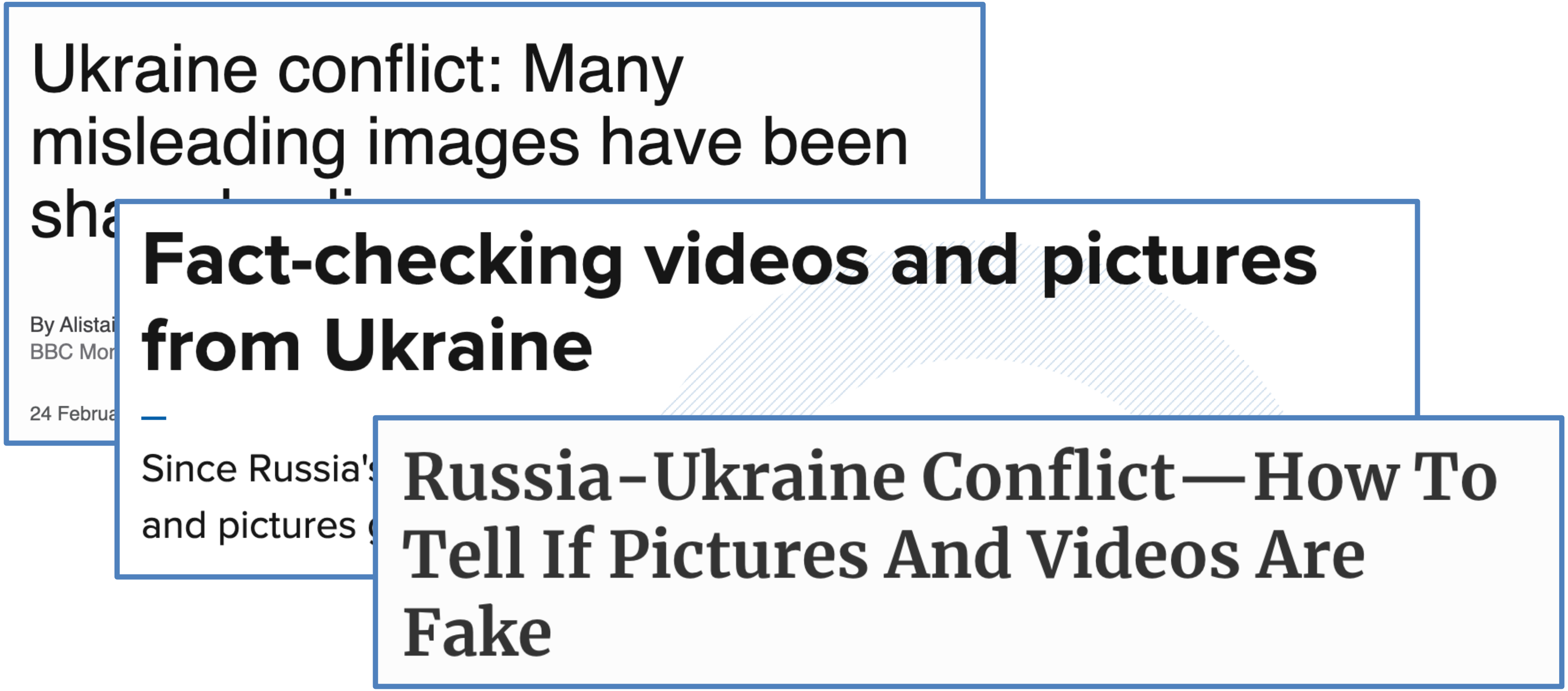
We are going to the moon …

Blockchains drive the development of SNARKs:

zkRollup,   zkBridge, zkCreditScore, zkTaxes, …

… but **many** non-blockchain applications

Using ZK to fight disinformation



Ukraine conflict: Many misleading images have been sha

By Alistai
BBC Mon

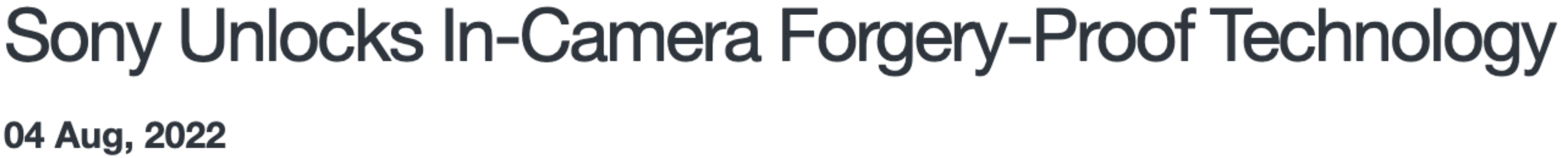
24 Februa

Fact-checking videos and pictures from Ukraine

Since Russia's
and pictures

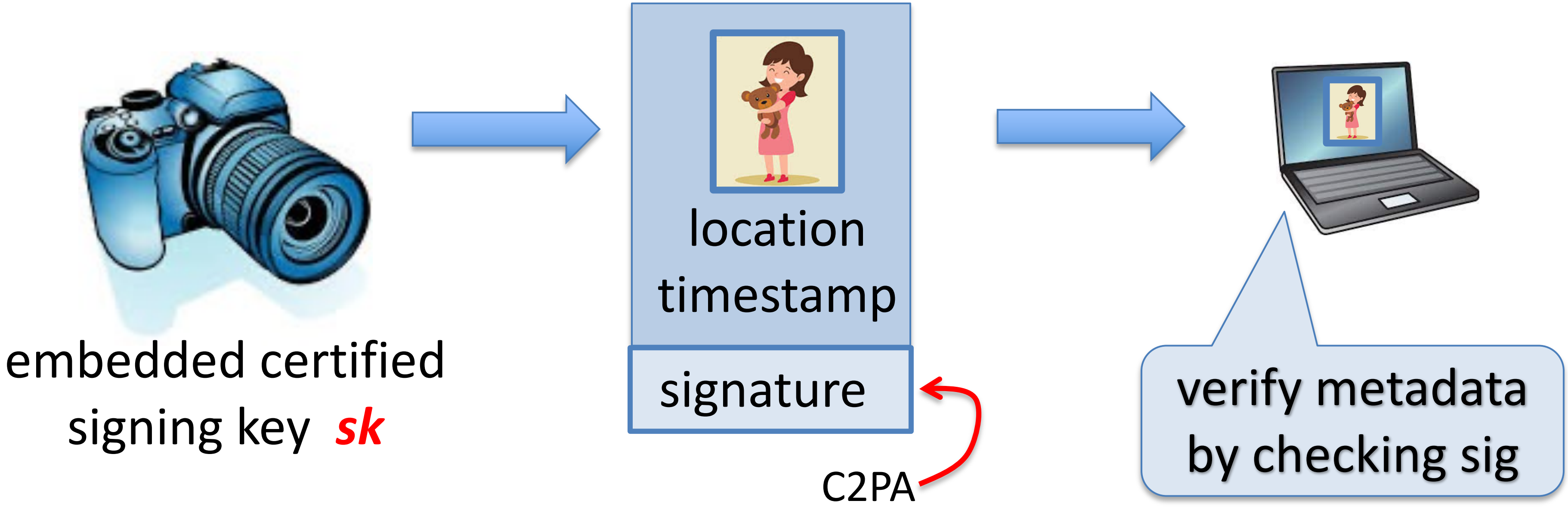
Russia-Ukraine Conflict—How To Tell If Pictures And Videos Are Fake

C2PA: a standard for content provenance



Sony Unlocks In-Camera Forgery-Proof Technology

04 Aug, 2022



embedded certified signing key *sk*

location

timestamp

signature

C2PA

verify metadata by checking sig

A problem: post-processing

Newspapers often process the photos before publishing:

- Resize (1500 × 1000), Crop, Grayscale (AP lists allowed ops)

The problem: laptop cannot verify signature on processed photo

C2PA "solution":
editing software will sign
processed photo to certify edits

A solution using ZK proofs (SNARKs)

(with T. Datta)

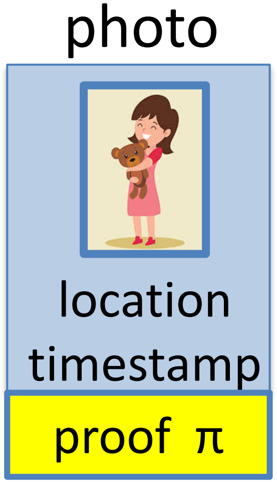
Editing software attaches a proof π to photo that:

I know a triple (***Orig***, ***Ops***, ***Sig***) such that

1. ***Sig*** is a valid C2PA signature on ***Orig***

2. photo is the result of applying ***Ops*** to ***Orig***

3. metadata(photo) = metadata(***Orig***)



⇒ Laptop verifies π and shows metadata to user

Performance

Proof size: 200-400 bytes. Verification time: 2 ms.
(in browser)

Proof generation time by newspaper:

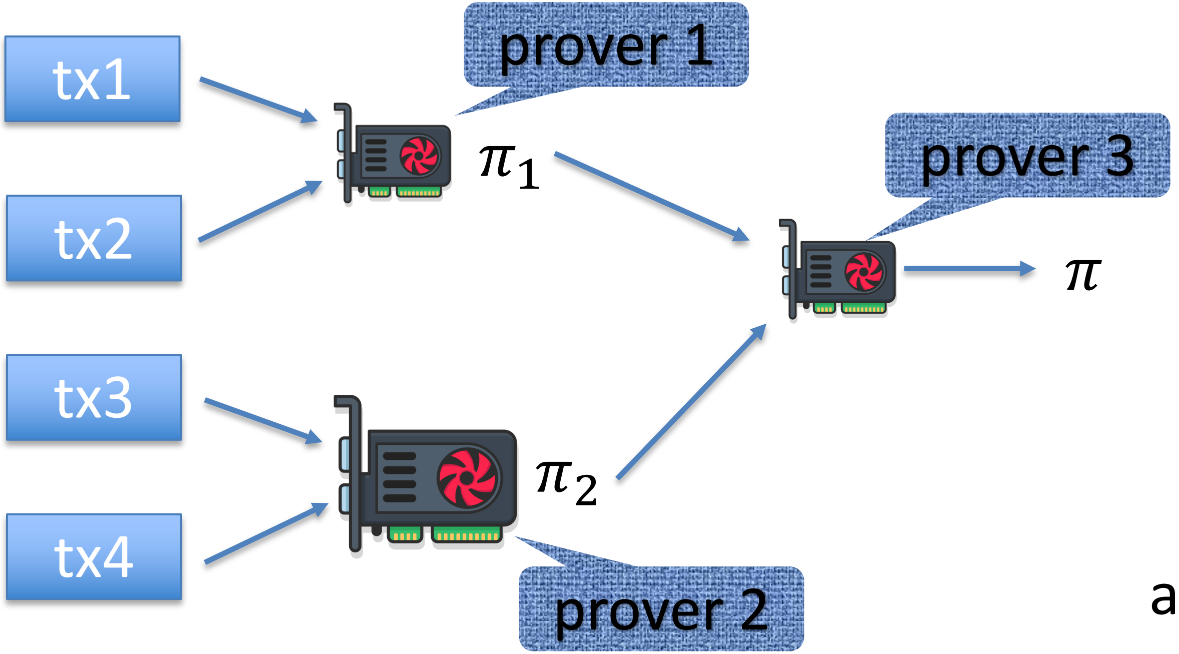
- Resize (3000 × 3000 → 1500 × 1500): 84 sec.
- Crop (3000 × 3000 → 1500 × 1500): 60 sec.
- Grayscale (2.25M pixels): 25 sec.

What about video??

See also: PhotoProof by Naveh & Tromer (2016)

The future: a market for ZK provers

Anyone with a GPU will be paid to create ZK proofs



market

selects provers
and distributes rewards

ZK: final thoughts

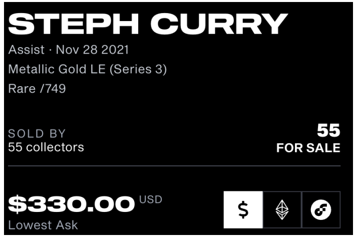
- **Lots** more to work on:
  - **Better provers**: faster, lower memory footprint, shorter proofs, quantum resistant, no trusted setup, distributed witness.
  - **New applications** for SNARKs and zk-SNARKs

DAOs

Recap: current application areas

1. **<u>Finance</u>** (DeFi):
   - new financial instruments, exchanges, lending, …

2. Managing **<u>digital assets</u>** (NFTs)
   - Assured provenance



STEPH CURRY
Assist · Nov 28 2021
Metallic Gold LE (Series 3)
Rare /749

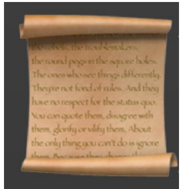
SOLD BY
55 collectors

55
FOR SALE

$330.00 USD
Lowest Ask

ENS

3. **Decentralized organizations** (DAOs):
   - DAOs for investment, for donations, for collecting art, etc.
   - Governance: group decision making



the rebels, the troublemakers,
the round pegs in the square holes.
The ones who see things differently.
They're not fond of rules. And they
have no respect for the status quo.
You can quote them, disagree with
them, glorify or vilify them. About
the only thing you can't do is ignore

Decentralized orgs (DAO)

What is a DAO?

- A Dapp deployed on-chain at a specific address
- Anyone (globally) can send funds to DAO treasury
- Anyone can submit a proposal to DAO

  ⟹ participants vote

  ⟹ approved → proposal executes

snapshot.org

(SafeSnap: trustless on-chain execution of off-chain votes)

Examples of DAOs

There are currently about 6500 DAOs managed on Snapshot

- **Collector DAOs**: PleasrDAO, flamingoDAO, ConstitutionDAO, …

  (see art collection at https://gallery.so/pleasrdao)

PleasrDAO: 103 members.

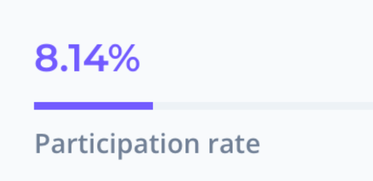
- Manages a treasury, has full time employees.
- Deliberations over what to acquire over telegram.

Examples of DAOs

There are currently about 6500 DAOs managed on Snapshot

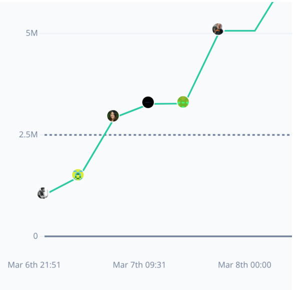
- Collector DAOs: PleasrDAO, flamingoDAO, ConstitutionDAO, …
- **Charity DAO**: gitcoin (42K members), …

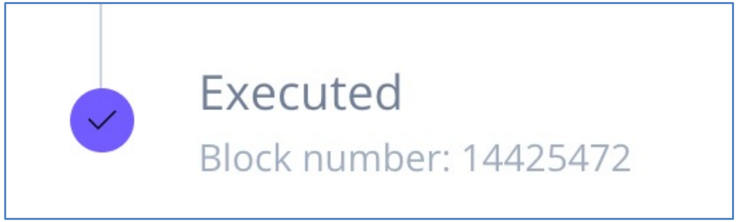
Proposal ID 21: This proposal looks to ratify the allocation of 30,000 GTC from the Community Treasury to the MMM workstream.



8.14%

Participation rate





Executed

Block number: 14425472

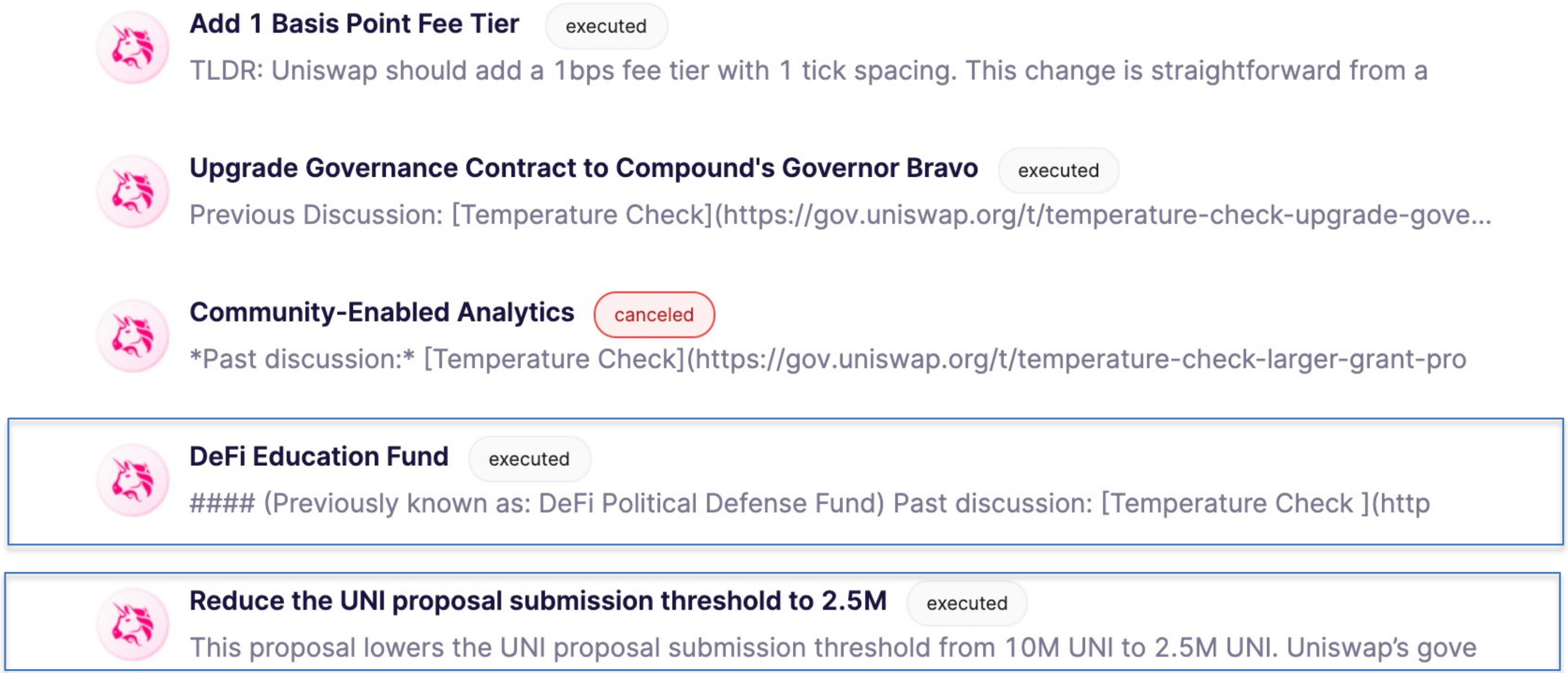
(tally.com)

Examples of DAOs

There are currently about 6500 DAOs managed on Snapshot

- Collector DAOs: PleasrDAO, flamingoDAO, ConstitutionDAO, …
- Charity DAO: gitcoin, …
- **Protocol DAO**: manages operation of a specific protocol
  Uniswap DAO (29K members), Compound (4K members), …
- **Social DAO**: FWB, …
- **Investment DAO**: many

Example: Uniswap proposals



Add 1 Basis Point Fee Tier executed

TLDR: Uniswap should add a 1bps fee tier with 1 tick spacing. This change is straightforward from a

Upgrade Governance Contract to Compound's Governor Bravo executed

Previous Discussion: [Temperature Check](https://gov.uniswap.org/t/temperature-check-upgrade-gove...

Community-Enabled Analytics canceled

Past discussion: [Temperature Check](https://gov.uniswap.org/t/temperature-check-larger-grant-pro

DeFi Education Fund executed

(Previously known as: DeFi Political Defense Fund) Past discussion: [Temperature Check](http

Reduce the UNI proposal submission threshold to 2.5M executed

This proposal lowers the UNI proposal submission threshold from 10M UNI to 2.5M UNI. Uniswap's gove

How to build a DAO

Three key decisions:

- What is the community for the DAO?

- How is membership managed?

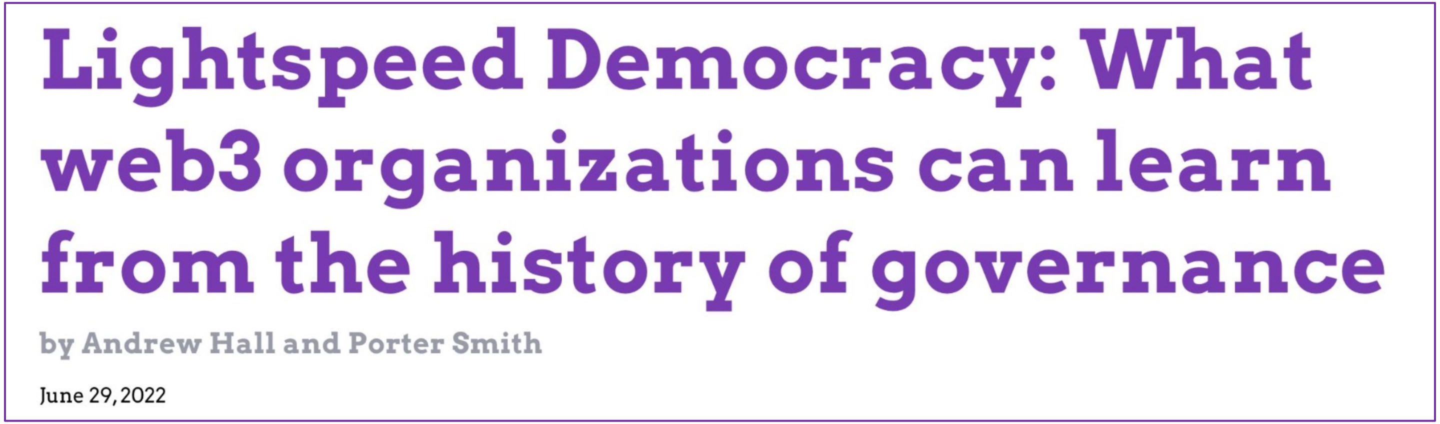
  Many available tools: Syndicate, Juicebox, Colony, …

  can anyone join, or does the community vote?

- How to do governance? What is controlled by governance?

Many DAO governance experiments

Who can vote? How to vote? What voting mechanism?



Lightspeed Democracy: What web3 organizations can learn from the history of governance

by Andrew Hall and Porter Smith

June 29, 2022

DAOs: a platform for experimenting with governance mechanisms

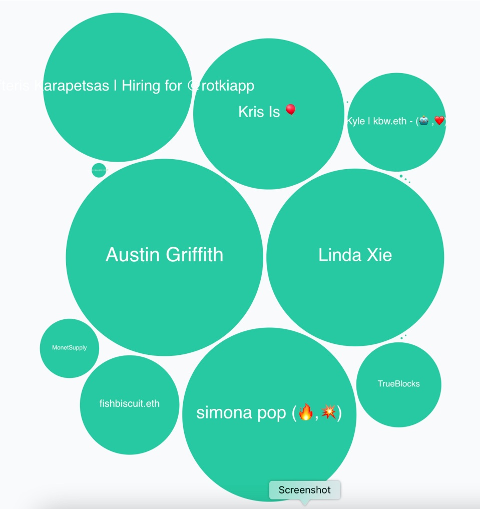
Governance methods

One token one vote: (most common)

- Members receive tokens based on their contribution.
- Everyone can vote.

Frequently implemented using one of
OpenZeppelin’s Governor contracts (Solidity code)

_castVote(**proposalID**, **voter**, **support**, **reason**);



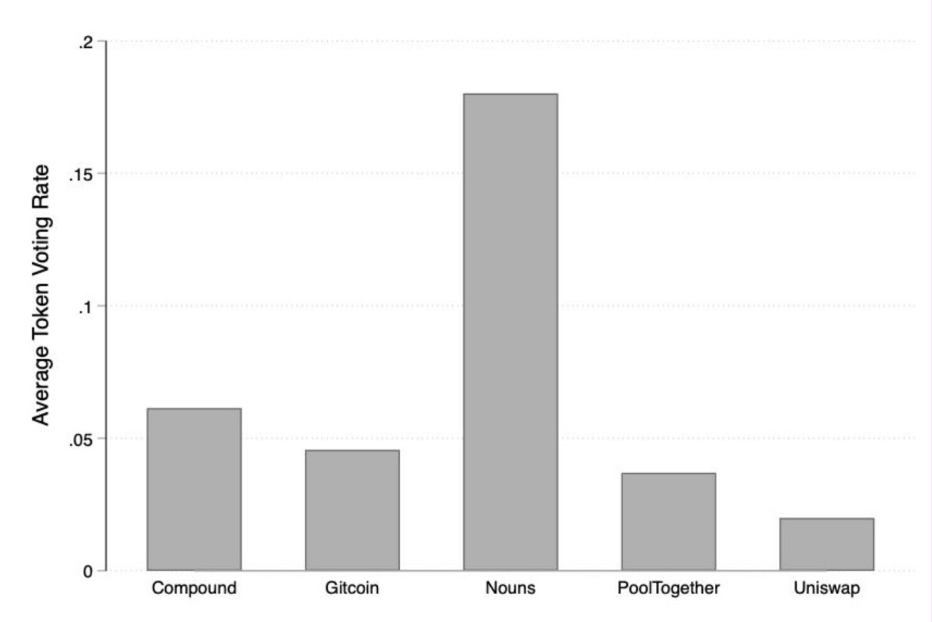
proposal 21 (tally.com)

Problem: direct democracy does not scale.

Poor participation rate

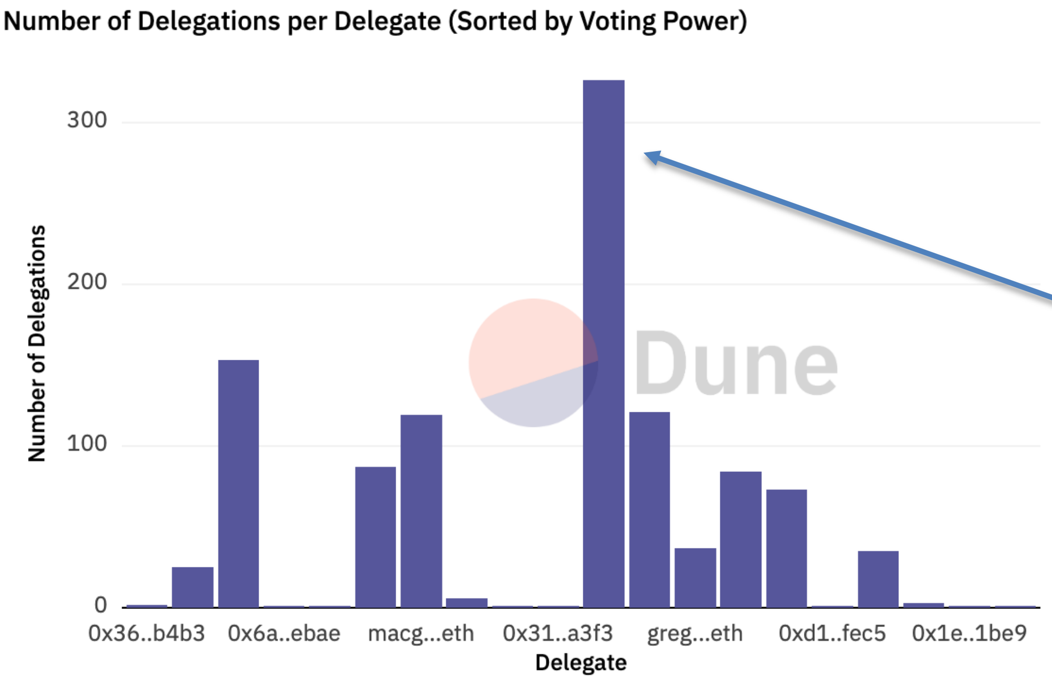
For all but one project:
participation rate < 5%

What to do? **delegation**
Supported in Governor contract



Voting rate = # Tokens voted / Total tokens in existence
These 5 DAOs sampled for convenience
Source: Boneh and Hall (super preliminary ongoing research)

Delegation example: element



≈300 addresses delegated tokens to this address

Private DAO treasury

2021: an auction for a physical copy of the constitution.

(Sotheby's auction house)

ConstitutionDAO:

- Formed in Nov. 2021 to participate in auction.
- Raised $46.3M from about 20K participants worldwide
- Lost to another bidder who bid $43M

bidder knew that ConstitutionDAO could not outbid it

How to participate in an auction when everyone knows your treasury??

Private DAO treasury

[Dunaif, Boneh, 2021]

The design:

One DAO platform manages many DAOs:

a single Ethereum contract (e.g., JuiceBox)

DAO manager: sets up a DAO by publishing a DAO public key (pk)

Contributor: sends funds to platform with a "blinded DAO-pk"

Contract records contribution

⇒ an observer learns nothing about which DAO received the funds

⇒ only learns total amount stored on the platform as a whole

DAO manager can later use its secret key to claim funds sent to its DAO

medium.com/@boneh

Many other DAO privacy questions …

- **Private DAO participation**: keep participant list private
- **Private voting**: keep who voted how on each proposal private
- **Private delegations**

… while complying with all relevant laws.

Some of these questions are solved by general privacy platforms such as **Aztec**, **Aleo**, and others.

END OF LECTURE

Next lecture: MEV and bridging