CS251 Fall 2021

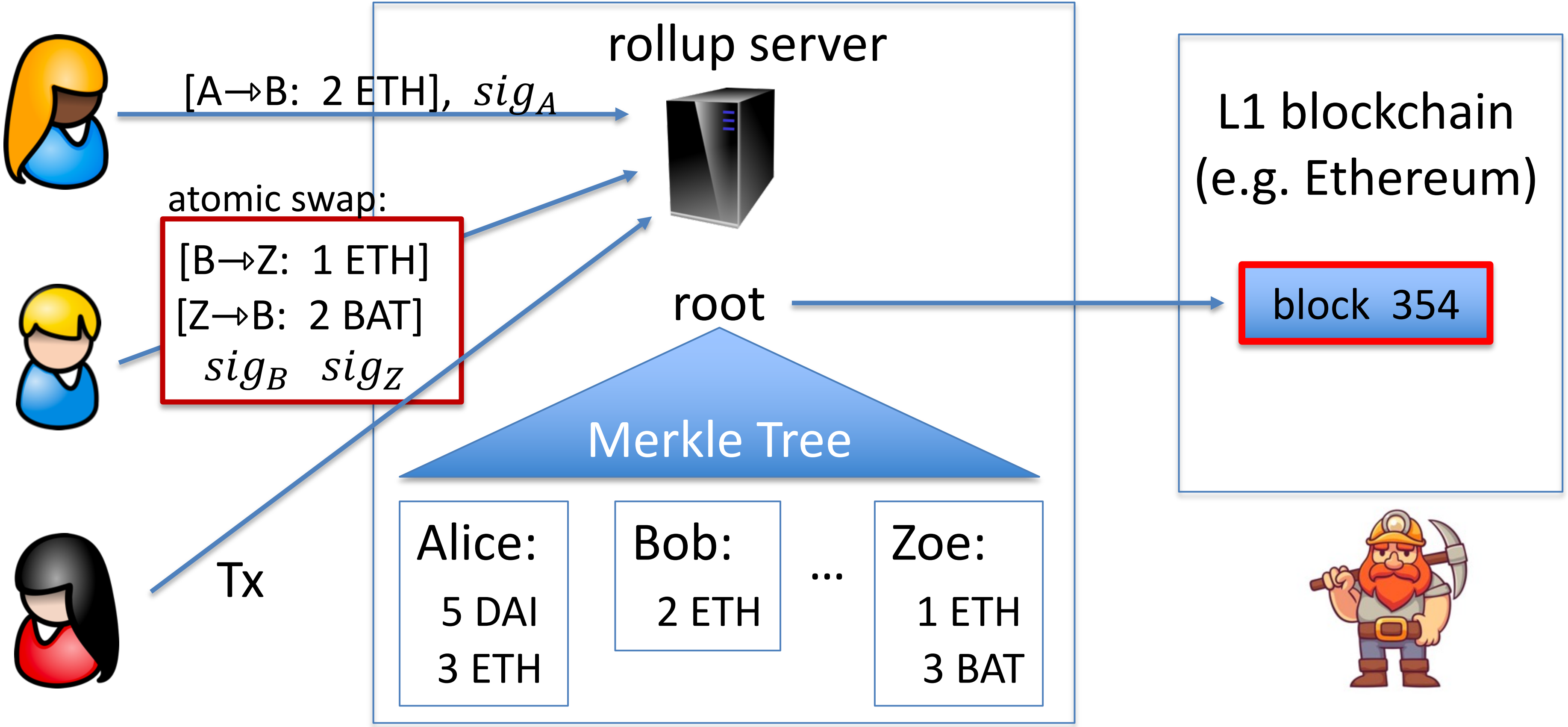
(cs251.stanford.edu)

Final Topics

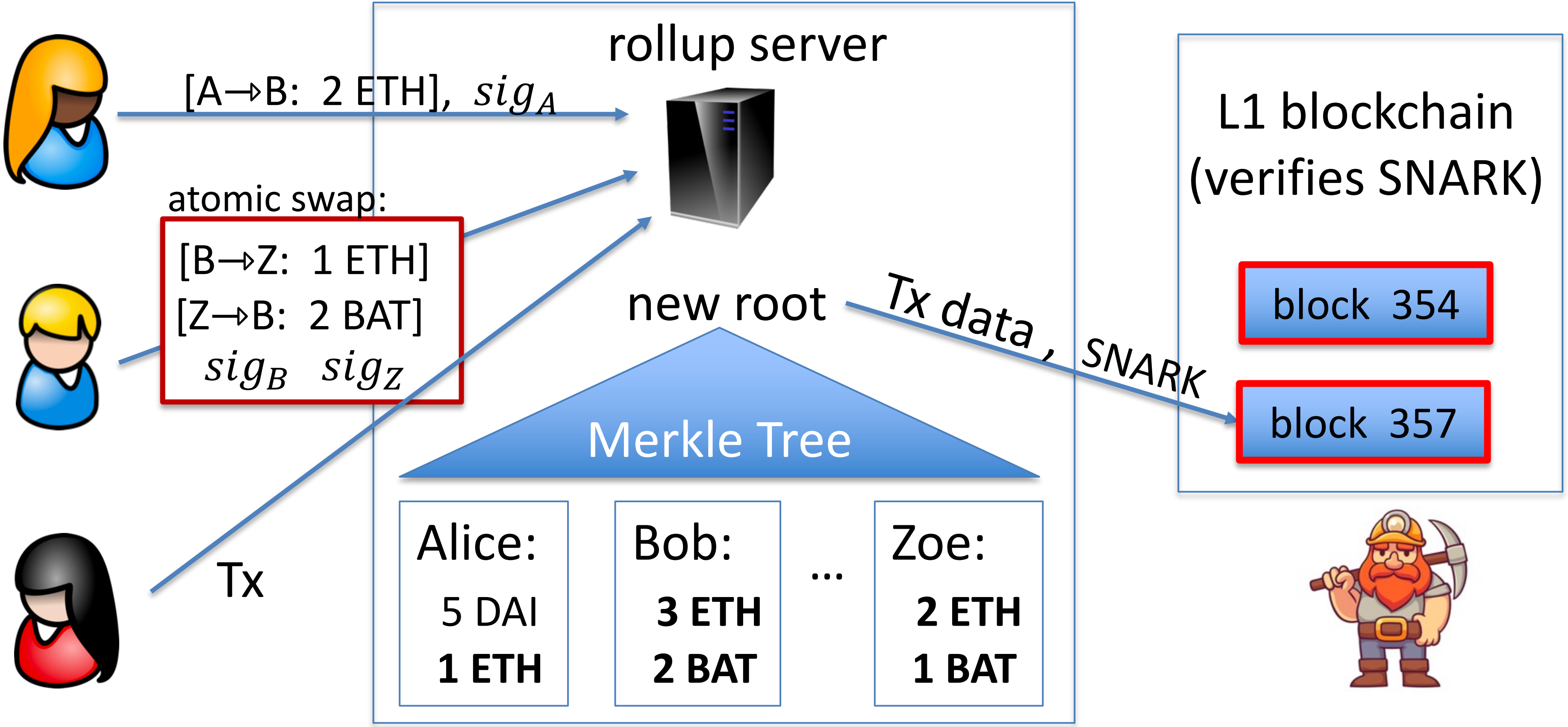
Dan Boneh

Invited talk final lecture.   Final exam will be released this week.

Quick Recap: zkRollup



Quick Recap: zkRollup



Key points

The Rollup server stores all account balances

- L1 chain does not store explicit balances

Rollup: Tx data written to L1 chain (16 gas per byte)

Validium: Tx data written to off-chain staked servers (cheaper)

why store Tx data? … backup in case rollup server fails

Can we hide Tx data from the Rollup server and the public?

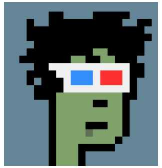
- Yes! Using $(zk)^2$-SNARKs

A brief discussion of NFTs

NFTs: managing digital assets

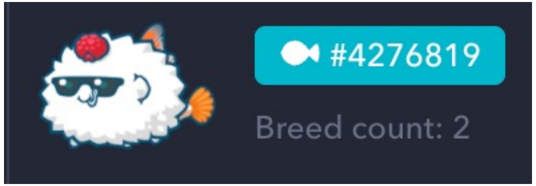
Example digital assets: (ERC-721)

- Digital art: opensea, foundation
- Collector items: NBA top shots
- Game items: horses (zed.run), axies, …
- Metaverse: ENS, plots in a virtual land



#8857

NBA

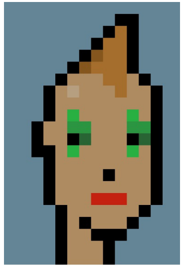


Why manage on a blockchain? Why not manage centrally?

- Blockchain ensures long-term ownership, until sale.
- Provides a trusted record of provenance (forgeries are evident)

Example: CryptoPunks

10,000 total CryptoPunks on Ethereum. Generated in 2017.



#7610

all offers and sales recorded on Ethereum (250 lines of Solidity)

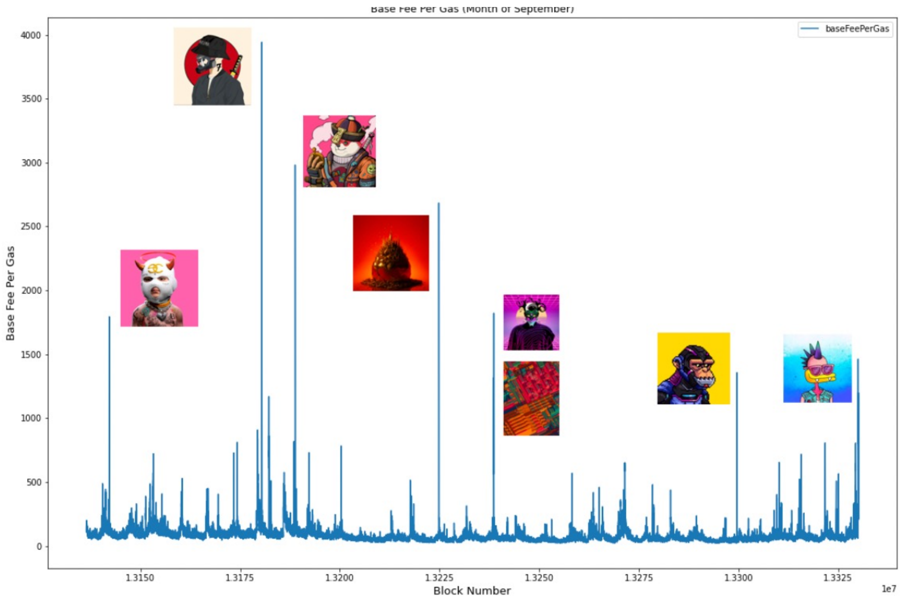
		visa			
Bid	beautifu...		150Ξ ($497,239)	Aug 24, 2021	
Sold	gmoney	0xa04e64	49.50Ξ ($149,939)	Aug 18, 2021	
Bid	0xa04e64		49.50Ξ ($149,024)	Aug 18, 2021	← buy offer
Sold	gr8wxl	0x84c920	21Ξ ($31,117)	Mar 05, 2021	
Offered			21Ξ ($31,117)	Mar 05, 2021	
Sold	0x02751f	gr8wxl	0.30Ξ ($67)	Aug 03, 2017	← sold!
Offered			0.30Ξ ($59)	Jul 30, 2017	← sell offer
Claimed		0x02751f		Jun 23, 2017	

https://www.larvalabs.com/cryptopunks/details/7610

The resulting gas wars

Gas prices spike around highly-anticipated NFT launches:

… maybe don't use first come first serve??



Base fee gas
Sep. 2021

https://www.paradigm.xyz/2021/10/a-guide-to-designing-effective-nft-launches/

digital assets: where is this going?

NFTs are about managing ownership of general digital assets

- Art
- Music
- Domain Names
- Virtual Worlds
- Trading Cards
- Collectibles
- Sports
- Utility

Growing list of categories on OpenSea

What does ownership mean:
- Where is item stored?
- Where can it be displayed?
- Who receives royalties on item: owner or creator?

digital assets: where is this going?

NFTs and DeFi: asset-based DeFi:

- Use NFT as collateral in loans (e.g., nftfi.com)
- Fractional ownership of NFT assets (e.g., fractional.art)
- NFT-based futures market

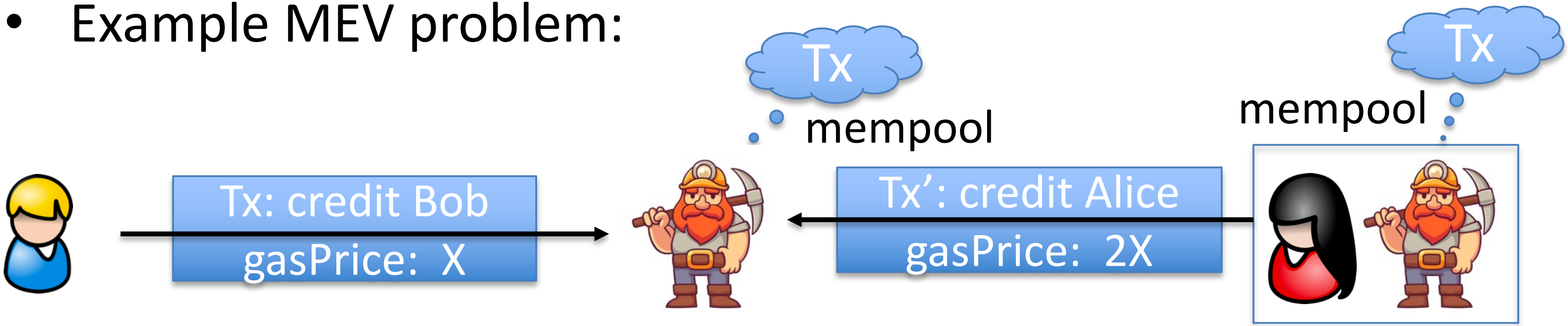
… all require a way to appraise an NFT (e.g., upshot.io)

Many more topics to cover

Many more topics to cover …

(1) Maximal extractable value (MEV):

- Recall: Ethereum v1 ⟹ all Tx enter a <u>public</u> mempool
- Example MEV problem:



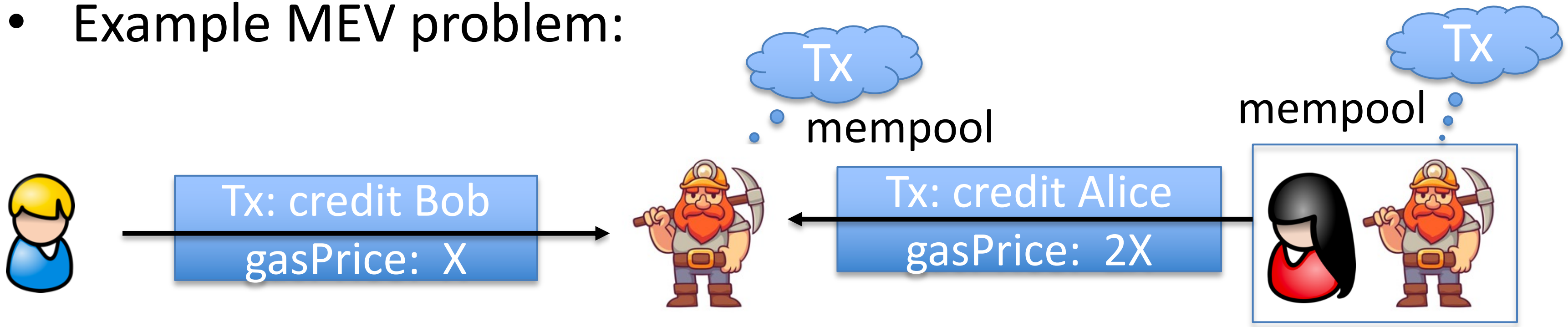
(i) Trader Bob finds a liquidation opportunity on Compound,
(ii) Alice scans mempool, finds Bob's Tx,
(iii) Alice issues Tx' with higher gasPrice, scheduled first, and takes Bob's profit

automated fontrunners ⟹ do this automatically

Many more topics to cover …

(1) Maximal extractable value (MEV):

- Recall: Ethereum v1 $\Longrightarrow$ all Tx enter a public mempool
- Example MEV problem:



Miner's revenues increase (MEV). Who gets hurt?

- Bob. Leads to high gas prices on Ethereum, and other bad effects

What to do? Several answers: see, e.g., **flashbots** (mev-geth)

Many more topics to cover …

(1) Maximal extractable value (MEV)

(2) On-chain Governance:

- How to decide on updates to Uniswap, Compound, … ???
- Current method:
  - Interested parties can buy governance tokens
  - One token one vote
- Better mechanisms?

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

Many more topics to cover …

(1) Maximal extractable value (MEV)

(2) Project governance:

- How to decide on updates to Uniswap, Compound, … ???

(3) Insurance: against bugs in Dapp code and other hacks

(4) Many more cute cryptography techniques (see slides at end)

(5) Interoperability between blockchains … discussed next

More topics …

- Where can I learn more?
  - **CS255** and **CS355**: Cryptography
  - **EE374**: Scaling blockchains with fast consensus
  - **Stanford blockchain conference** (SBC): Jan. 24-26, 2022.
  - **Stanford blockchain club**

Discussion: a career in blockchains? Where to start?

Bridging blockchains

Many L1 blockchains

Bitcoin: Bitcoin scripting language (with Taproot)

Ethereum: EVM. Currently: expensive Tx fees (better in Eth2)

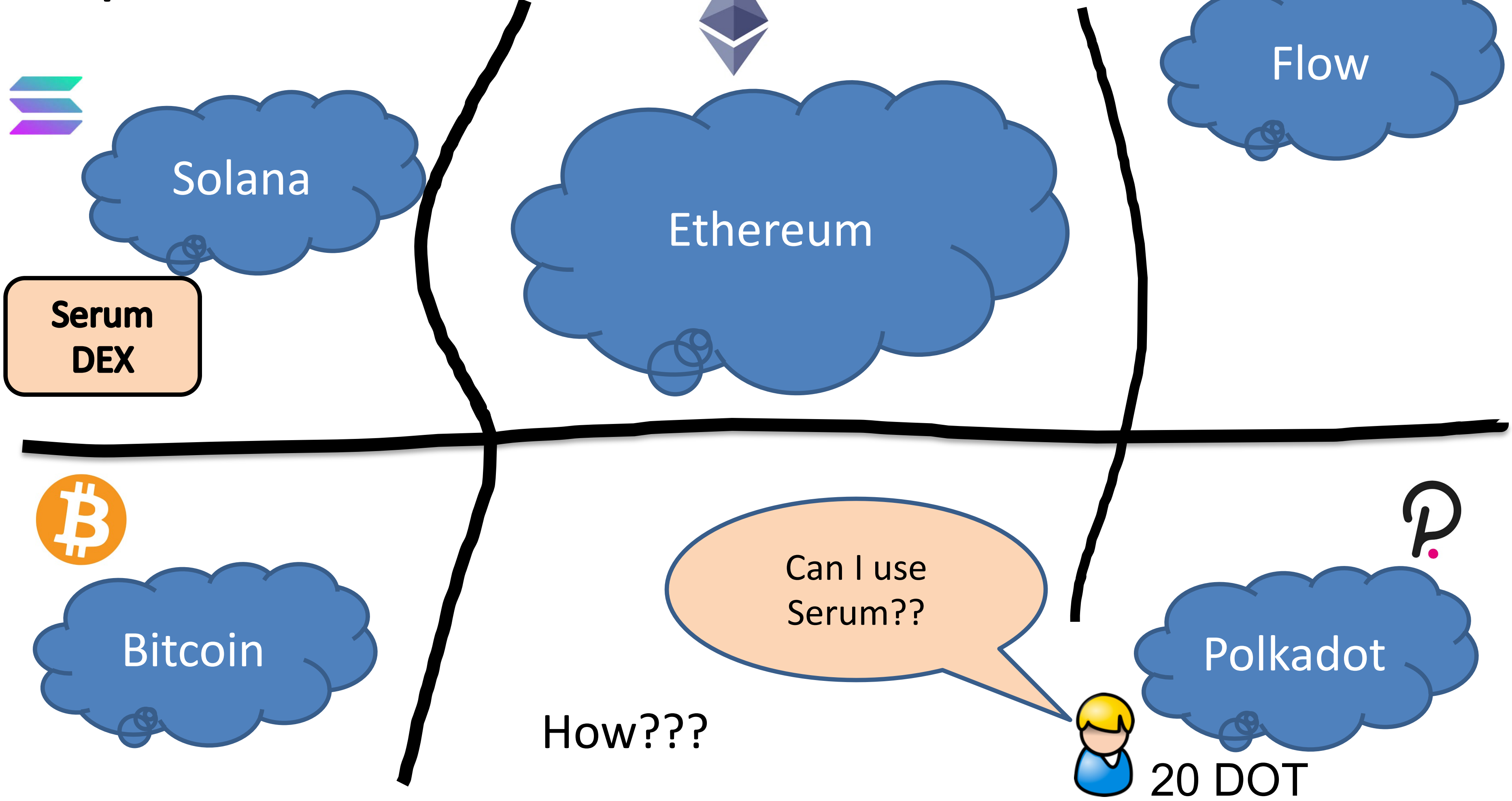
EVM compatible blockchains: **Celo, Avalanche, BSC**, …

- Higher Tx rate $\Longrightarrow$ lower Tx fees
- EVM compatibility $\Longrightarrow$ easy project migration and user support

Other fast non-EVM blockchains: **Solana, Flow, Algorand**, …

- Higher Tx rate $\Longrightarrow$ lower Tx fees

The problem: siloes



Interoperability

Interoperability:

- User owns funds or assets (NFTs) on one blockchain system
  Goal: enable user to move assets to another chain

Composability:

- Enable a DAPP on one chain to call a DAPP on another

Both are easy if the entire world used Ethereum

- In reality: many blockchain systems that need to interoperate
- The solution: **bridges**

A first example: BTC in Ethereum

How to move BTC to Ethereum ??   Goal: enable BTC in DeFi.

$\Longrightarrow$ need new ERC20 on Ethereum pegged to BTC

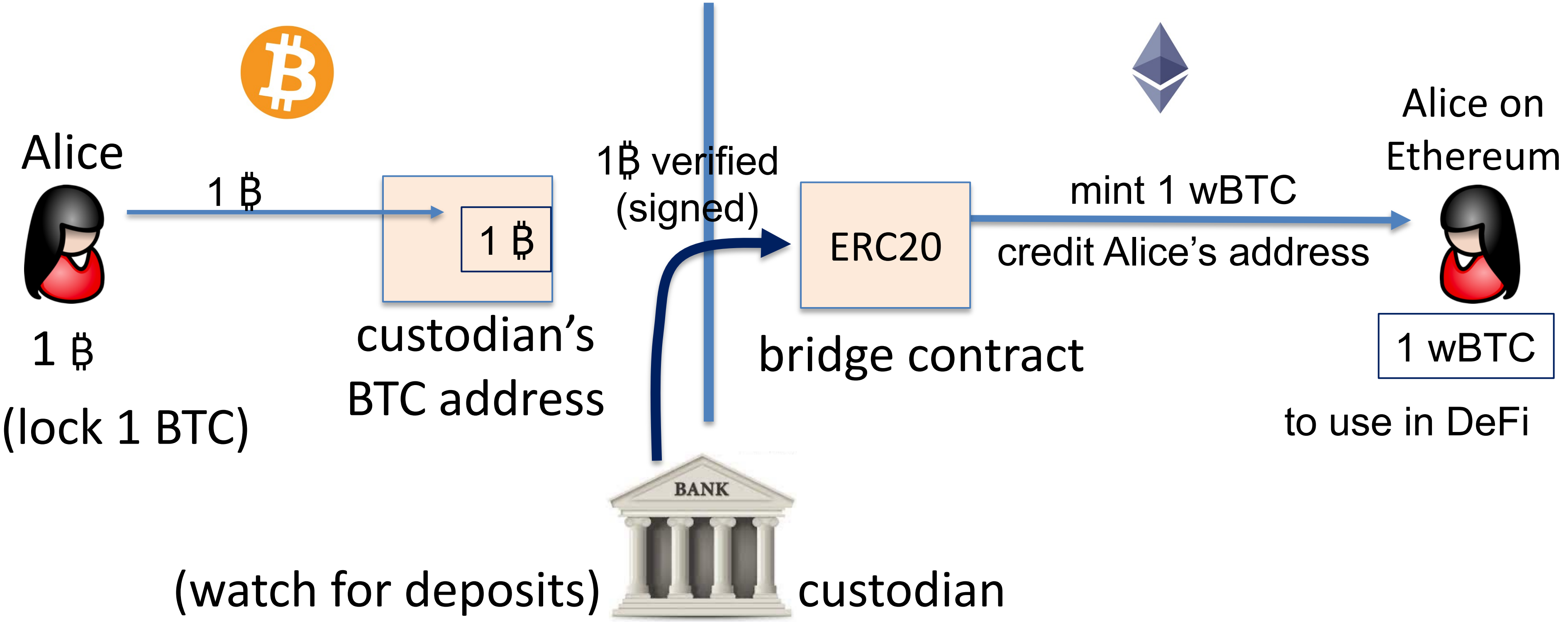
(e.g., use it for providing liquidity in DeFi projects)

The solution: **wrapped coins**

- Asset X on one chain appear as wrapped-X on another chain
- For BTC: several solutions   (e.g., wBTC, tBTC)

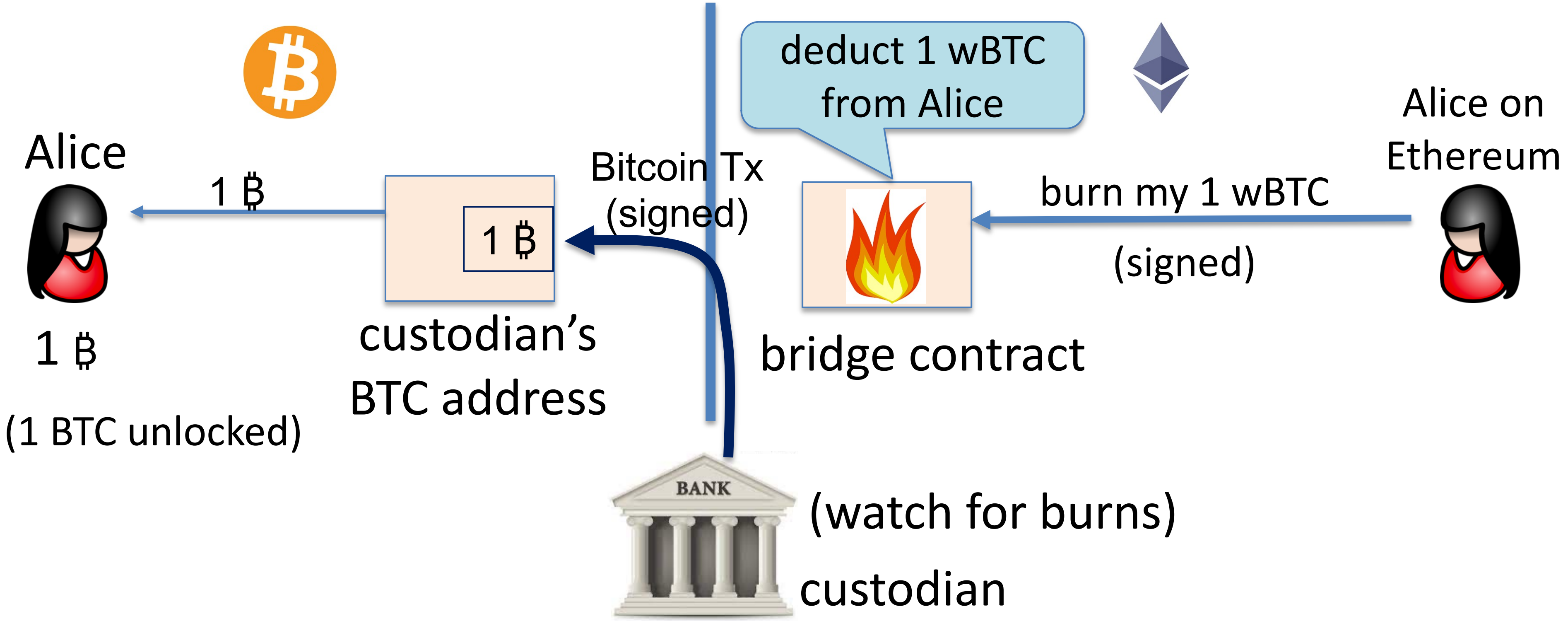
wBTC and tBTC: a lock-and-mint bridge

Let's start with wBTC: **moving 1 BTC to Ethereum**



Alice wants her 1 BTC back

Moving 1 wBTC back to the Bitcoin network:



wBTC

Example BTC → Ethereum:

Nov 26 2021 - 07:36	FUNDS SENT TO CUSTODIAN (Bitcoin Tx: ≈4,000 BTC) c605b4f2f0948e7deae0c5d7c27b3256b97120be760e2b81136eb95c819570f6	
Nov 26 2021 - 09:50	MINT COMPLETED BY CUSTODIAN (Ethereum Tx:) 0x70475eca8be89b67143f1b52df013fc1df7d254e836c836c8f368fc516aca76b	

Why two hours? … make sure no Bitcoin re-org

CUSTODY Nov. 2021

253,387.2485 BTC
($14,268,319,582.44 USD)

The problem: trusted custodian

Can we do better?

tBTC: no single point of trust

Alice requests to mint tBTC:

random three registered custodians are selected and
they generate P2PKH Bitcoin address for Alice

signing key is 3-out-of-3 secret shared among three

(all three must cooperate to sign a Tx)

Alice sends BTC to P2PKH address, and received tBTC.

Custodians must lock 1.5x ETH stake for the BTC they manage

- If locked BTC is lost, Alice can claim staked ETH on Ethereum.

Bridging smart chains (with Dapp support)

A very active area:

- Many super interesting ideas

Asset-specific	Chain-specific	Application-specific	Generalized
ever (AR)	Avalanche	ANY SWAP	AXELAR
INTERLAY (BTC)	BINANCE	Biconomy	Chainlink
tBTC	GRAVITY BRIDGE	CELER	ChainSafe
WBTC	Harmony	CHAINFLIP	composable
WRAPPED	(PoS Bridge)	Gateway	connext
	ETH NEAR Rainbow Bridge	liquality	deBridge
	Ronin	Qredo	IBC Inter-Blockchain Communication
	secret network	Ren	LayerZero.
	(SnowBridge)	Synapse	Movr
	Terra Shuttle	THORCHAIN	OPTICS
	TokenBridge	wanchain	Polymer
	WORMHOLE		PolyNetwork
	WRAP		orbit
	(XCMP)		router

1kx
@dberenzon

https://medium.com/1kxnetwork/blockchain-bridges-5db6afac44f8

Two types of bridges

Type 1: a lock-and-mint bridge

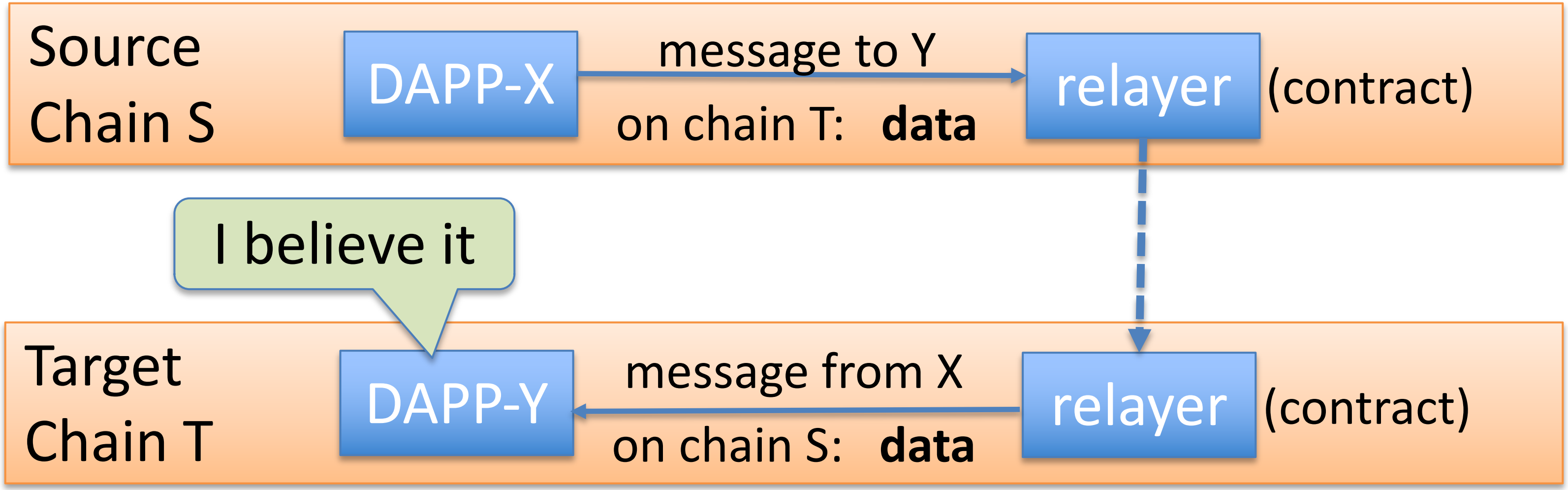
- SRC → DEST: user locks funds on SRC side,
  wrapped tokens are minted on the DEST side
- DEST → SRC: funds are burned on the DEST side,
  and released from lock on the SRC Side

Type 2: a liquidity pool bridge

- Liquidity providers provide liquidity on both sides
- SRC → DEST: user sends funds on SRC side,
  equivalent amount released from pool on DEST side

Bridging smart chains (with Dapp support)

Step 1 (hard): a secure cross-chain messaging system



Step 2 (easier): build a bridge using messaging system

Bridging smart chains (with Dapp support)

Step 1 (hard): a secure cross-chain messaging system

Step 2 (easier): build a bridge using messaging system

- DAPP-X → DAPP-Y: "I received 3 CELO, ok to mint 3 wCELO"
- DAPP-Y → DAPP-X: "I burned 3 wCELO, ok to release 3 CELO"

If messaging system is secure, no one can steal locked funds at S

Primarily two types of messaging systems

(1) **Externally verified**: external parties verify message on chain S

RelayerT dispatches only if all trustees signed

$\Longrightarrow$ <u>**if**</u> DAPP-Y trusts trustees, it knows DAPP-X sent message

Primarily two types of messaging systems

(1) **Externally verified**: external parties verify message on chain S

What if trustees sign and post a fake message to relayerT?

- off-chain party can send trustee's signature to relayerS $\Longrightarrow$ trustee slashed

Primarily two types of messaging systems

(2) **On-chain verified**: chain T verifies block header of chain S

receive msgs

Source Chain S

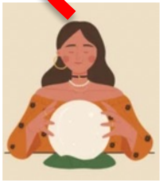
relayerS

send messages D[] to relayerT, along with finalized block header on chain S, and Merkle proofs

verify and dispatch

relayerT

Target Chain T

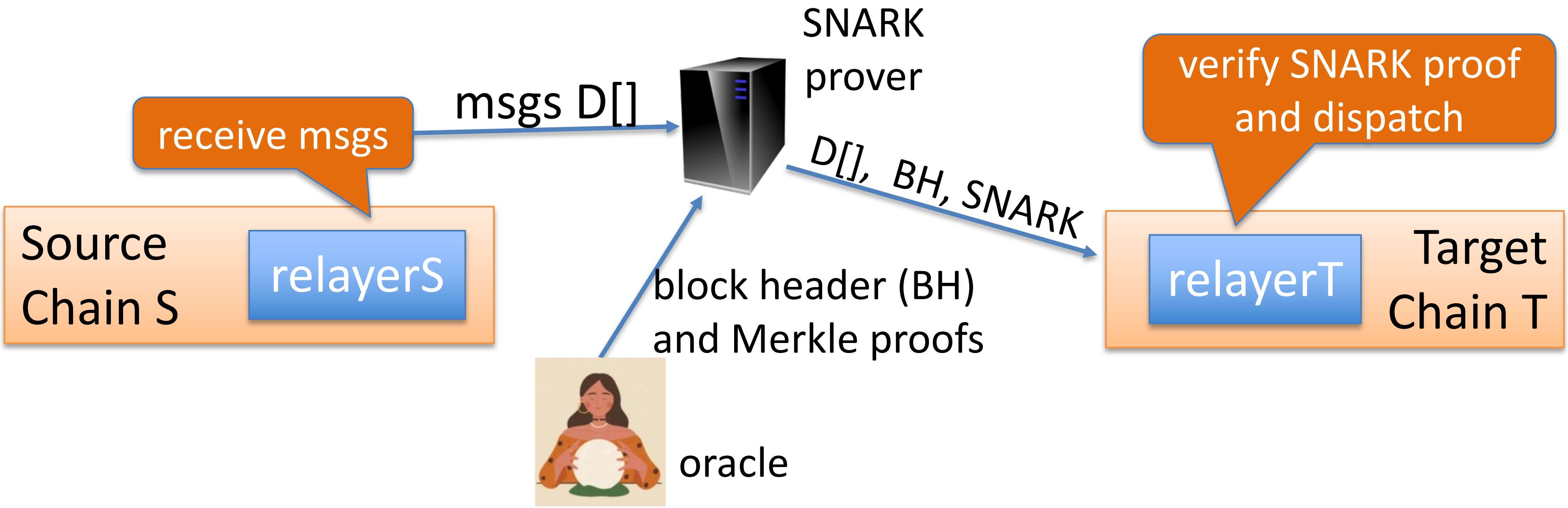


oracle

relayerT runs a (light) client for chain S to verify that relayerS received messages D[]

no trustees

Primarily two types of messaging systems



Problem: high gas costs on chain T to verify state of source chain.
Solution: use SNARKs $\Longrightarrow$ little work for relayerT

Bridging: the future vision

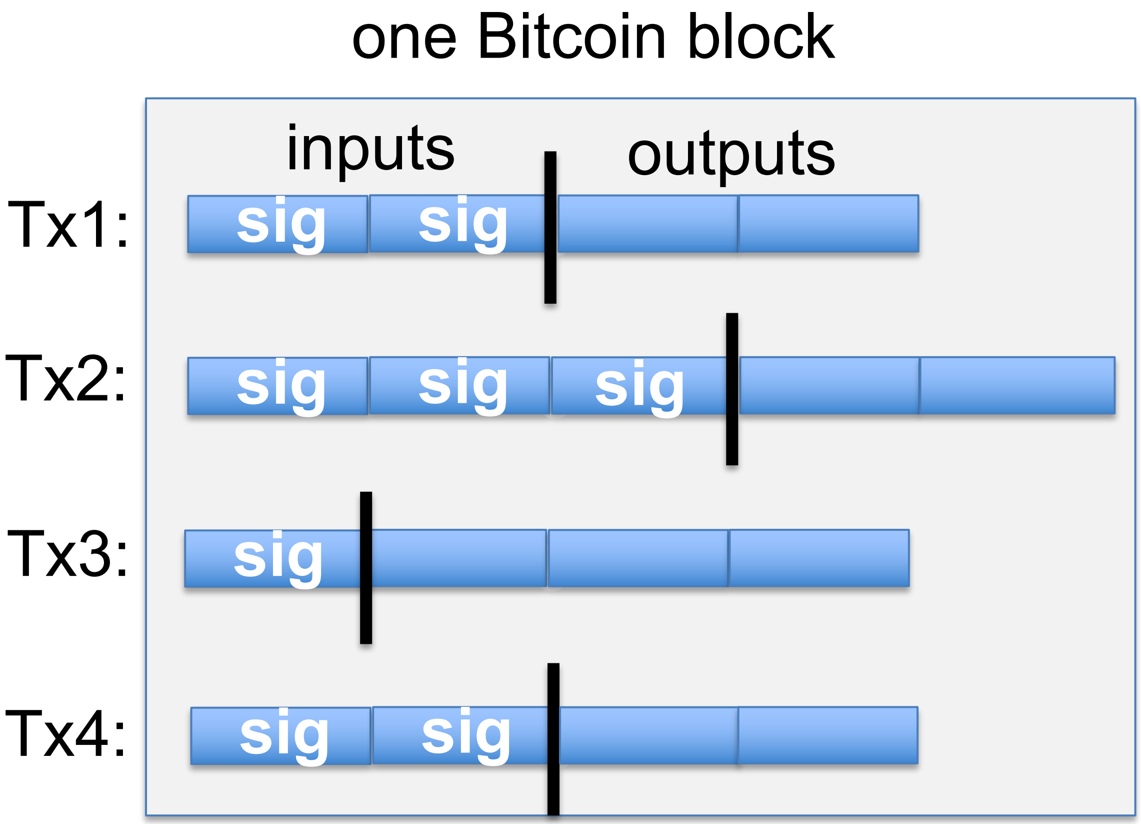
User can hold assets on any chain

- Assets move cheaply and quickly from chain to chain
- A project's liquidity is available on all chains
- Users and projects choose the chain that is best suited for their application and asset type

We are not there yet …

Fun crypto tricks

BLS signatures



Signatures make up most of Tx data.

Can we compress signatures?

- Yes: aggregation!
- not possible for ECDSA

BLS Signatures

Used in modern blockchains: Ehtereum 2.0, Dfinity, Chia, etc.

The setup:

- $G = \{1, g, \ldots, g^{q-1}\}$ a cyclic group of prime order q
- $H: M \times G \to G$ a hash function (e.g., based on SHA256)

BLS Signatures

<u>KeyGen</u>(): choose random α in $\{1, \ldots, q\}$

output $\boxed{\text{sk} = \alpha \;, \quad \text{pk} = g^{\alpha} \;\; \in \text{G}}$

<u>Sign</u>(sk, m): output $\boxed{\text{sig} = H(m, \text{pk})^{\alpha} \;\; \in \text{G}}$

<u>Verify</u>(pk, m, sig): output accept if $\log_g(\text{pk}) = \log_{H(m,\text{pk})}(\text{sig})$

Note: signature on m is unique! (no malleability)

How does verify work?

A pairing: an efficiently computable function $e: G \times G \to G'$

such that $\boxed{e(g^{\alpha}, g^{\beta}) = e(g,g)^{\alpha\beta}}$ for all $\alpha, \beta \in \{1, \ldots q\}$

and is not degenerate: $e(g,g) \neq 1$

verify test

Observe: $\log_g(pk) = \log_{H(m,pk)}(sig)$

if and only if $\boxed{e(g, sig) = e(pk, H(m,pk))}$

$$\begin{array}{ccc} e(g, sig) & = & e(pk, H(m,pk)) \\ \| & & \| \\ e(g, H(m,pk)^{\alpha}) & = & e(g^{\alpha}, H(m,pk)) \end{array}$$

Properties: signature aggregation [BGLS'03]

<u>Anyone</u> can compress n signatures into <u>one</u>

pk_1 , m_1 $\longrightarrow$ σ_1

$\vdots$

pk_n , m_n $\longrightarrow$ σ_n

aggregate $\rightarrow \sigma^*$

single short signature

Verify($\overline{\mathbf{pk}}$, $\overline{\mathbf{m}}$, σ^*) = "accept"

convinces verifier that

for i=1,…,n:

user i signed msg m_i

Aggregation: how

user 1: $pk_1 = g^{\alpha 1}$, m_1 $\longrightarrow$ $\sigma_1 = H(m_1, pk_1)^{\alpha_1}$

$\vdots$

user n: $pk_n = g^{\alpha n}$, m_n $\longrightarrow$ $\sigma_n = H(m_n, pk_n)^{\alpha_n}$

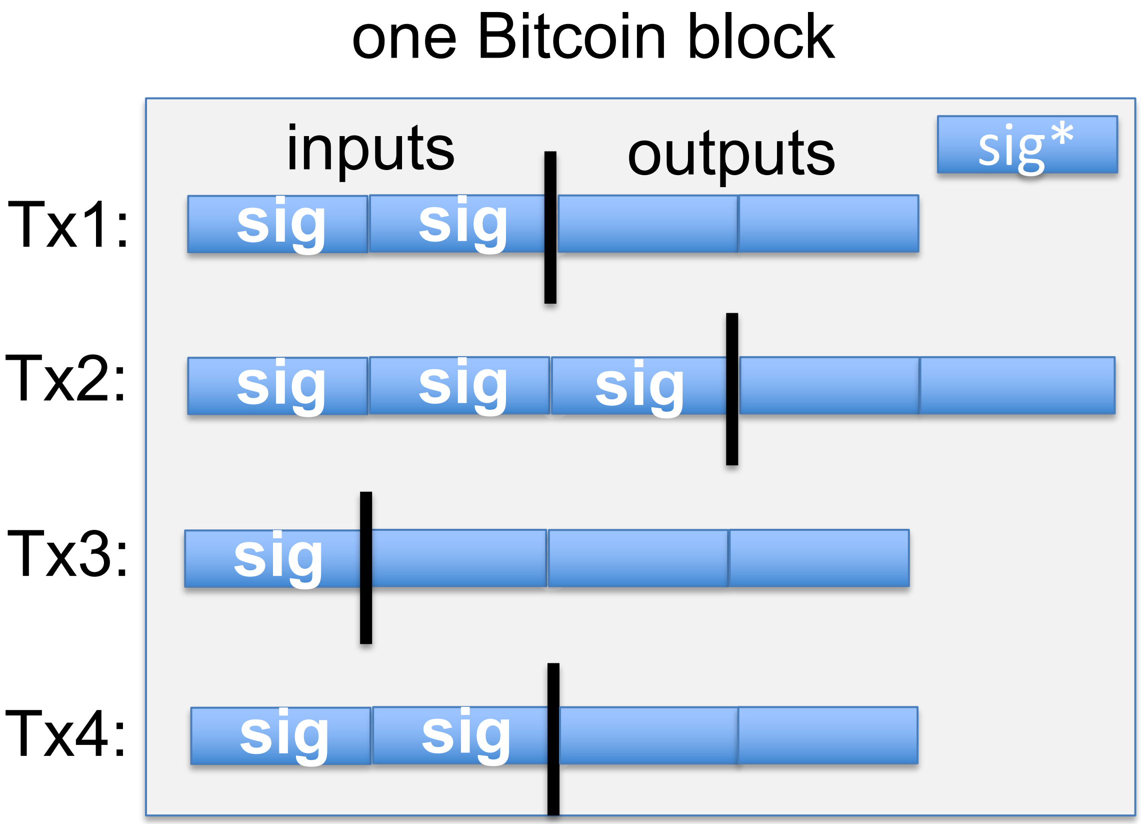
$\sigma \longleftarrow \sigma_1 \cdots \sigma_n$

Verifying an aggregate signature: (incomplete)

$$\prod_{i=1}^{n} e(H(mi,pki), g^{\alpha_i}) \stackrel{?}{=} e(\sigma, g)$$

$$\Pi_{i=1}\, e(H(m_i,pk_i)^{\alpha_i}, g) = e(\Pi_{i=1} H(m_i,pk_i)^{\alpha_i}, g)$$

Compressing the blockchain with BLS



if needed:

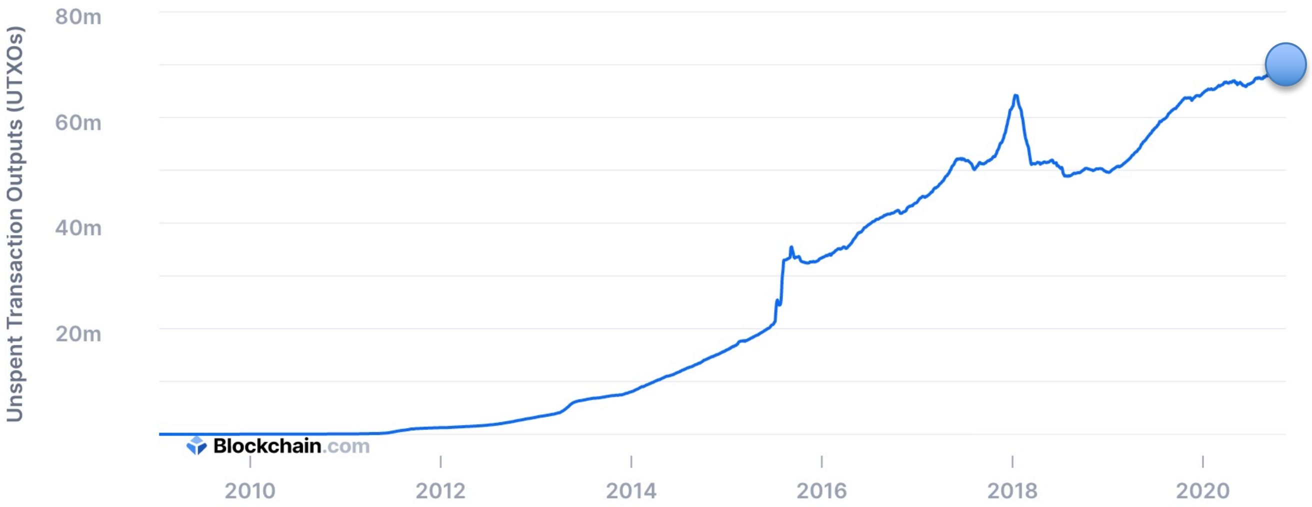
compress all signatures in a block into a single aggregate signatures

⇒ shrink block

or: aggregate in smaller batches

Reducing Miner State

UTXO set size



≈70M UTXOs

Miners need to keep all UTXOs in memory to validate Txs

Can we do better?

Recall: polynomial commitments

- *commit*(*pp*, f, r) → ***com$_f$*** commitment to f $\in \mathbb{F}_p^{(\leq d)}[X]$

- *eval*: goal: for a given ***com$_f$*** and x, y $\in \mathbb{F}_p$,

  construct a SNARK to prove that f(x) = y.

Homomorphic polynomial commitment

A polynomial commitment is **homomorphic** if

there are efficient algorithms such that:

- <u>*commit*</u>(*pp*, f_1, r_1) → $\boldsymbol{com_{f1}}$ <u>*commit*</u>(*pp*, f_2, r_2) → $\boldsymbol{com_{f2}}$

Then:

(i) for all $a, b \in \mathbb{F}_p$: $\boldsymbol{com_{f1}}$, $\boldsymbol{com_{f2}}$ → $\boldsymbol{com_{a*f1+b*f2}}$

(ii) $\boldsymbol{com_{f1}}$ → $\boldsymbol{com_{X*f1}}$

Committing to a set (of UTXOs)

(accumulator)

Let $S = \{U_1, \ldots, Un\} \in \mathbb{F}_p$ be a set of UTXOs

Define: $f(X) = (X - U_1) \cdots (X - Un) \quad \in \mathbb{F}_p^{(\leq n)}[X]$

Set: $\boldsymbol{com_f}$ = commit(pp, f, r) ⟵ short commitment to S

For $U \in \mathbb{F}_p$: $U \in S$ if and only if $f(U) = 0$

To add U to S: $\boldsymbol{com_f} \to \boldsymbol{com_{X*f-U*f}}$ ⟵ short commitment to $S \cup \{U\}$

How does this help?

Miners maintain two commitments:

(i) commitment to set T of all UTXOs

(ii) commitment to set S of spent TXOs

$\Big\}$ ≤ 1KB

$com_T, \; com_S$

Tx format:

- every input U includes a proof $(U \in T \;\;\&\&\;\; \mathrm{U} \notin S)$

  Two eval proofs: $T(U) = 0 \;\;\&\&\;\; S(U) \neq 0$ (short)

Tx processing: miners check eval proofs, and if valid,
add inputs to set S and outputs to set T. That’s it!

Does this work ??

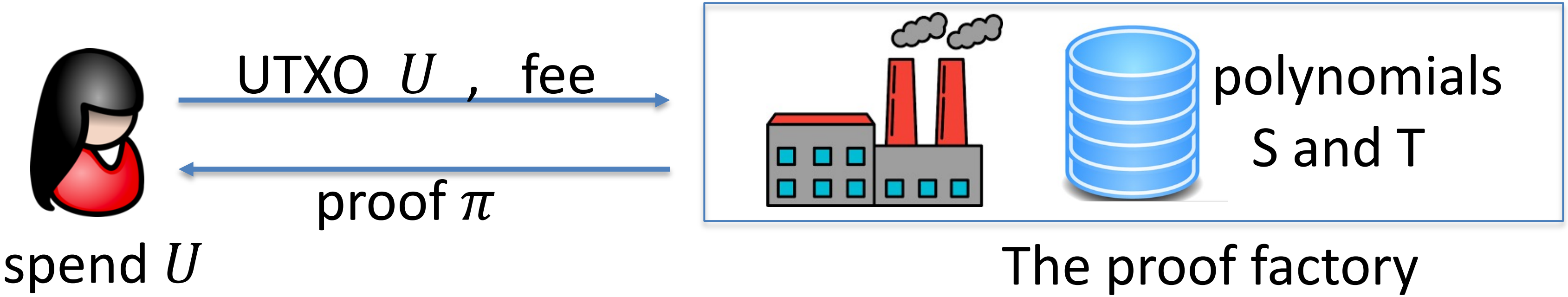
Problem: how does a user prove that her UTXO U satisfies

$$T(U) = 0 \quad \&\& \quad S(U) \neq 0 \qquad ???$$

This requires knowledge of the entire blockchain

$\Rightarrow$ user needs large memory and compute time

$\Rightarrow$ … can be outsourced to an untrusted 3rd party



Is this practical?

Not quite …

- Problem: the factory's work per proof is <u>linear</u> in the number of UTXOs ever created

- <u>Many</u> variations on this design:
  - can reduce factory's work to $\log_2$(# current UTXOs) per proof
  - Factory's memory is linear in (# current UTXOs)

End result: outsource memory requirements to a small number of 3rd party service providers

Taproot: semi-private scripts in Bitcoin

Taproot is here …

Bitcoin's long-anticipated Taproot upgrade is activated

November 14, 2021, 12:49AM EST · 1 min read

Script privacy

Currently: Bitcoin scripts must be fully revealed in spending Tx

Can we keep the script secret?

Answer: Yes, easily! when all goes well …

How?

ECDSA and Schnorr public keys:

- **<u>KeyGen</u>**(): sk = α , pk = g^{α} ∈ G for α in $\{1, \ldots, q\}$

Suppose $sk_A = \alpha$, $sk_B = \beta$.

- Alice and Bob can sign with respect to pk $= pk_A \cdot pk_B = g^{\alpha+\beta}$

  ⇒ an interactive protocol between Alice and Bob

  (note: much simpler with BLS)

  ⇒ Alice & Bob can imply consent to Tx by signing with pk = $g^{\alpha+\beta}$

How?

S: Bitcoin script that must be satisfied to spend a UTXO U

S involves only Alice and Bob. Let $pk_{AB} = pk_A \cdot pk_B$

Goal: keep S secret when possible.

How: modify S so that a signature with respect to

$$\text{pk} = pk_{AB} \cdot g^{H(pk_{AB}\,,S)}$$

is sufficient to spend UTXO, without revealing S !!

The main point

- If parties agree to spend UTXO,

  $\Rightarrow$ sign with respect to pk_{AB} and spend while keeping S secret

- If disagreement, Alice can reveal S

  and spend UTXO by proving that she can satisfy S.

Taproot pk compactly supports both ways to spend the UTXO

END OF LECTURE

Next lecture: super cool final guest lecture