# Privacy Preserving Proofs of Solvency

Konstantinos Chalkias
Kevin Lewi
Payman Mohassel
Valeria Nikolaenko

ZKProof Community Event
AMSTERDAM | OCTOBER 28 & 29, 2019

# Popularity of cryptocurrencies is expanding around the globe

***mainly via custodial wallets and exchanges***

## Reported statistics

- 40+M blockchain wallet users as of Oct 2019
- It was 10M four years ago
- 3-5% of Americans own Bitcoin
- 13M users for the most popular bitcoin wallet and exchange provider
- 45% of users are between 24-35

  30% between 35-44

  12% between 45-54

  Only 8% between 18-24

- Several accounts are no longer in use and many users occupy several wallets. But, Asian markets are not included.

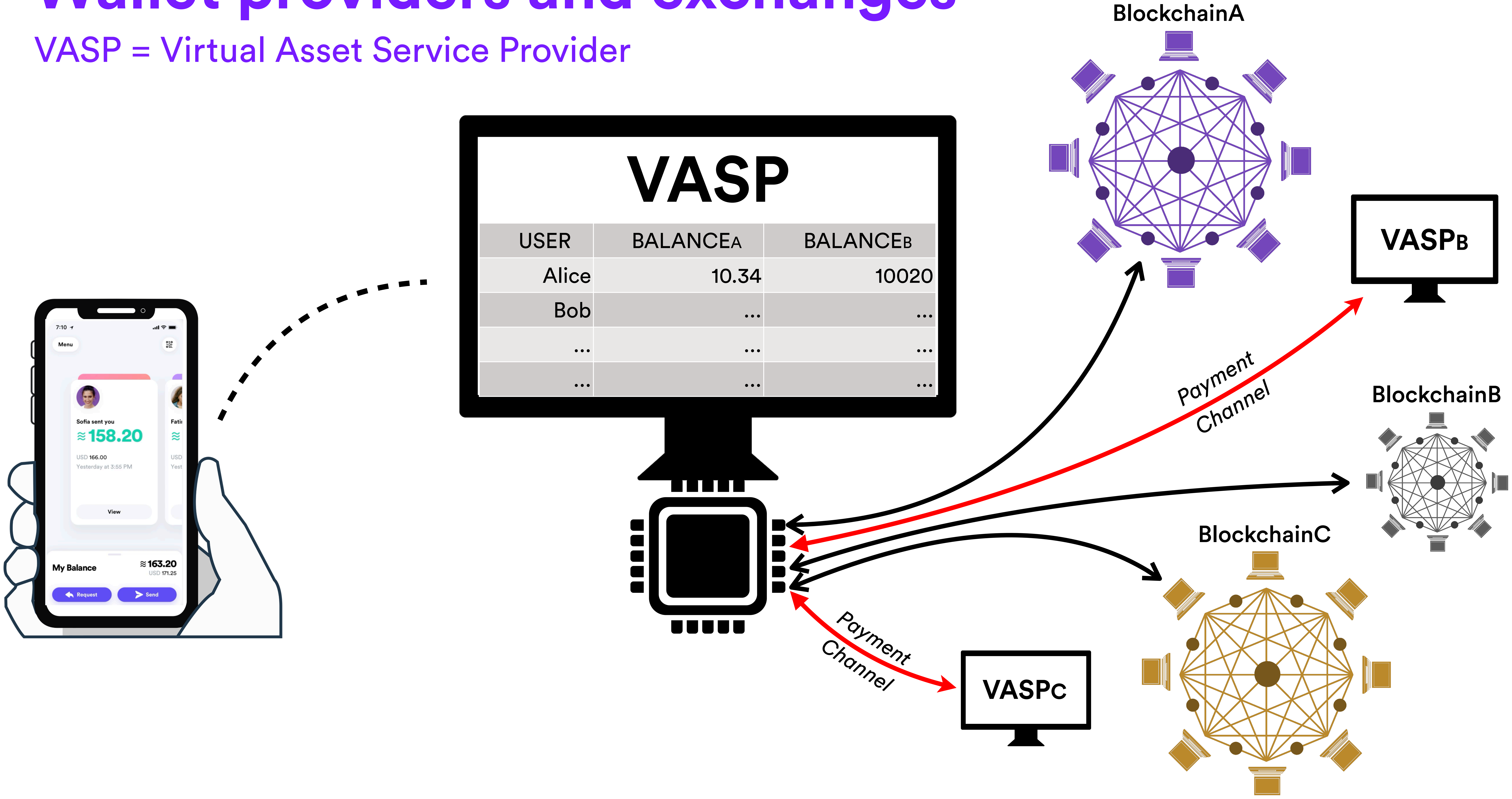
*Sources:*

*futurism.com - https://futurism.com/coinbase-users-surpasses-charles-schwab-brokerage-accounts*

*bitcoinmarketjournal.com - https://www.bitcoinmarketjournal.com/how-many-people-use-bitcoin*

# Wallet providers and exchanges

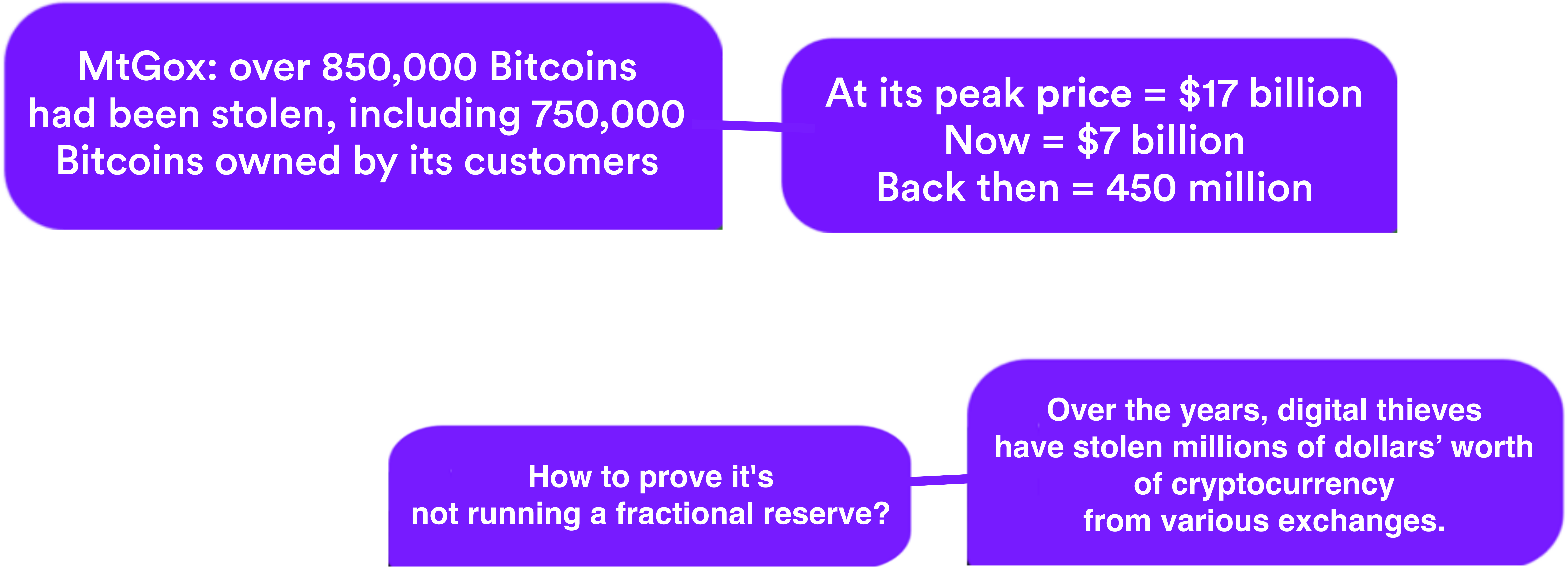
VASP = Virtual Asset Service Provider



VASP

| USER | BALANCEA | BALANCEB |
|---|---|---|
| Alice | 10.34 | 10020 |
| Bob | ... | ... |
| ... | ... | ... |
| ... | ... | ... |

# Is your VASP solvent?



MtGox: over 850,000 Bitcoins had been stolen, including 750,000 Bitcoins owned by its customers

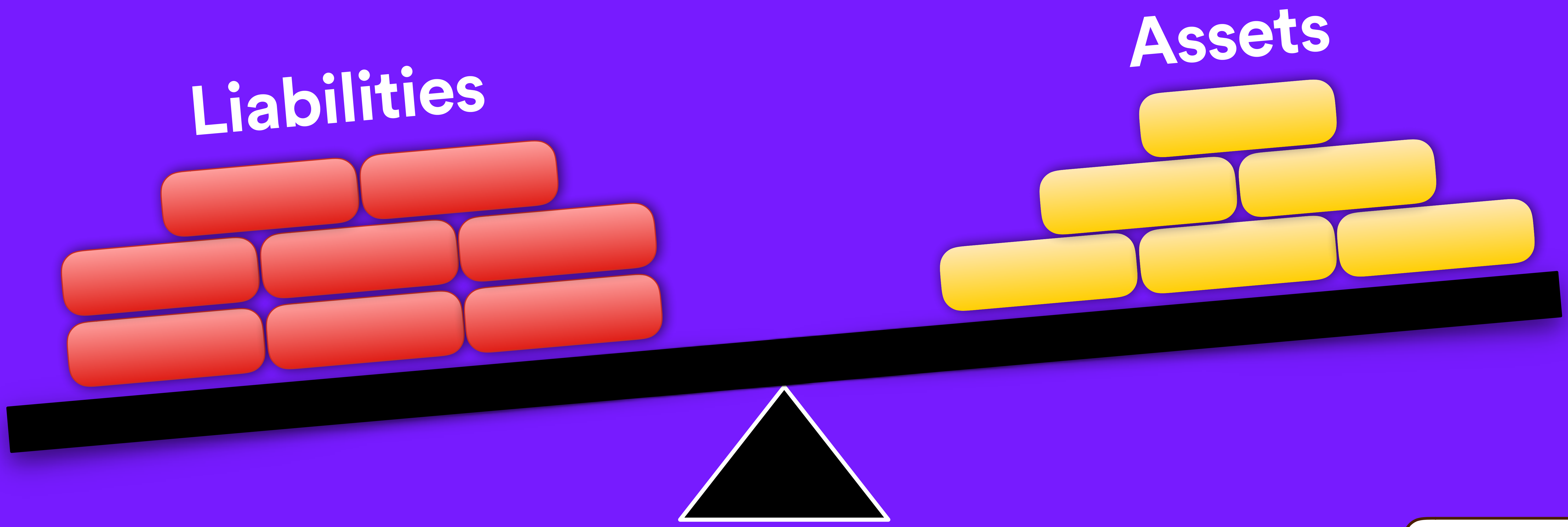
At its peak price = $17 billion
Now = $7 billion
Back then = 450 million

How to prove it's not running a fractional reserve?

Over the years, digital thieves have stolen millions of dollars’ worth of cryptocurrency from various exchanges.

# Solvency Ratio

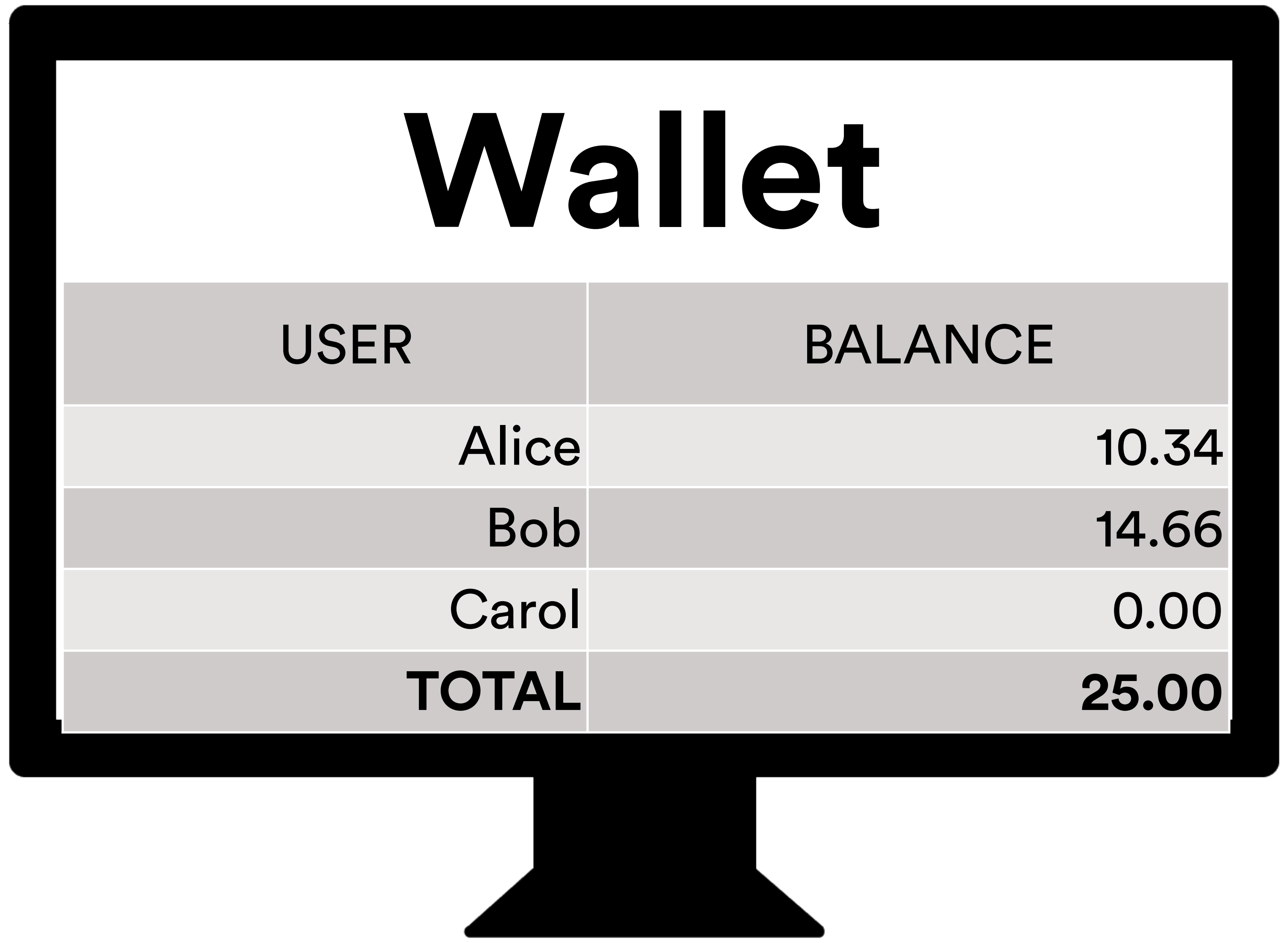
**Lower** the value of **Solvency Ratio** indicates a greater **probability of default** on the debt obligations



Ensure

**Liabilities <= Assets**

# Option A [Broadcast Everything]



## Wallet

| USER | BALANCE |
|---:|---:|
| Alice | 10.34 |
| Bob | 14.66 |
| Carol | 0.00 |
| **TOTAL** | **25.00** |

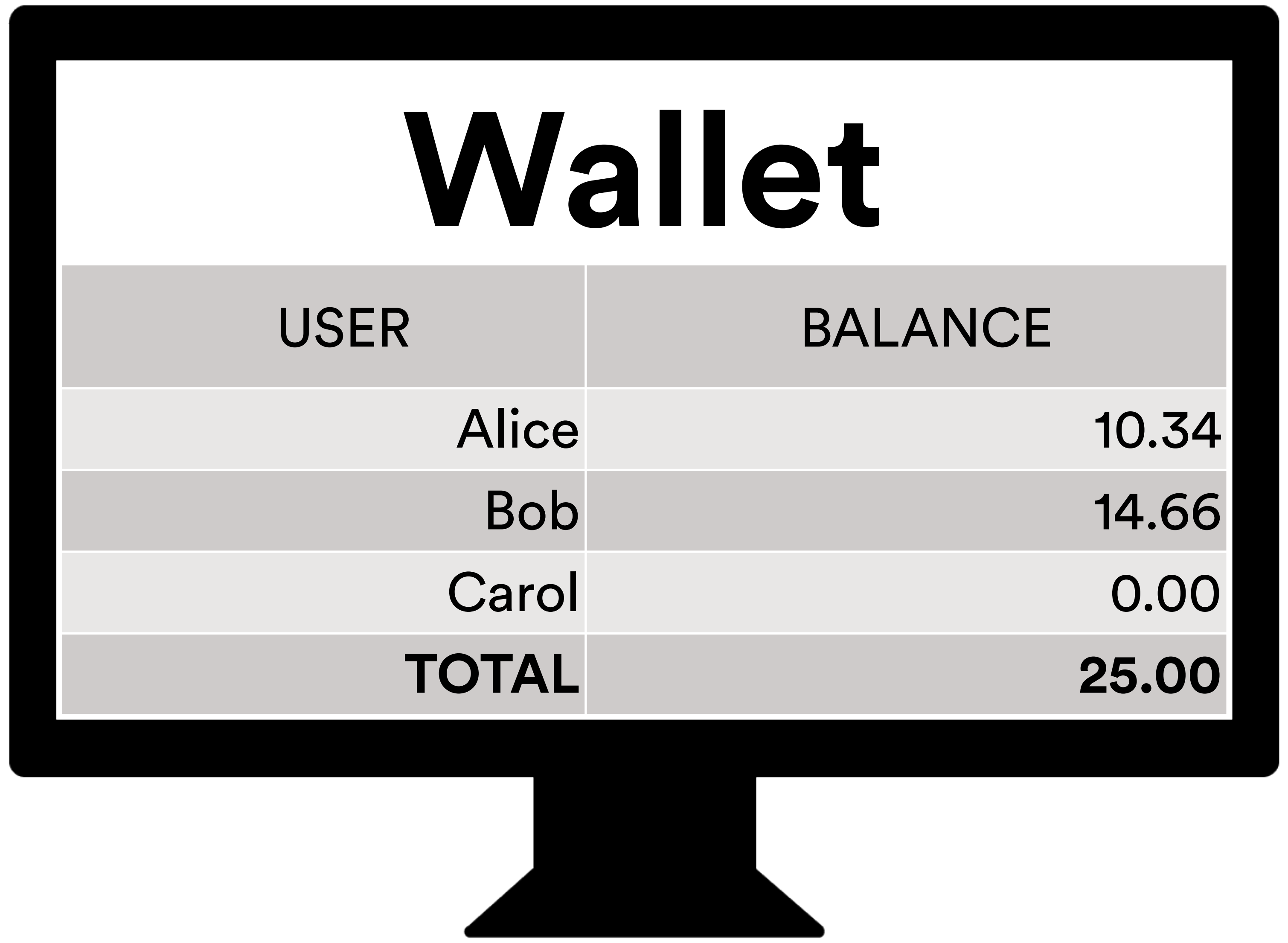
**Publicly expose**

- individual wallet balances
- wallet identities
- blockchain addresses
- blockchain balances
- wallet performance
- zero balance customers
- total liabilities (& assets)

## Blockchain

| ADDRESS | BALANCE |
|---:|---:|
| 0×434aaba2151 | 3.50 |
| 0×312323441aa | 0.20 |
| 0xbbafcddd1aa | 6.30 |
| ... | 10.00 |
| ... | 2.50 |
| ... | 2.50 |
| **TOTAL** | **25.00** |

# Option B [Publish to Auditor(s) only]



## Wallet

| USER | BALANCE |
|---|---|
| Alice | 10.34 |
| Bob | 14.66 |
| Carol | 0.00 |
| **TOTAL** | **25.00** |

**Expose to auditors**

- individual wallet balances
- wallet identities
- blockchain addresses
- blockchain balances
- wallet performance
- total liabilities (& assets)

**Wallet - Auditor collusion?**

## Blockchain

| ADDRESS | BALANCE |
|---|---|
| 0×434aaba2151 | 3.50 |
| 0×312323441aa | 0.20 |
| 0xbcafcddd1ca | 6.30 |
| ... | 10.00 |
| ... | 2.50 |
| ... | 2.50 |
| **TOTAL** | **25.00** |

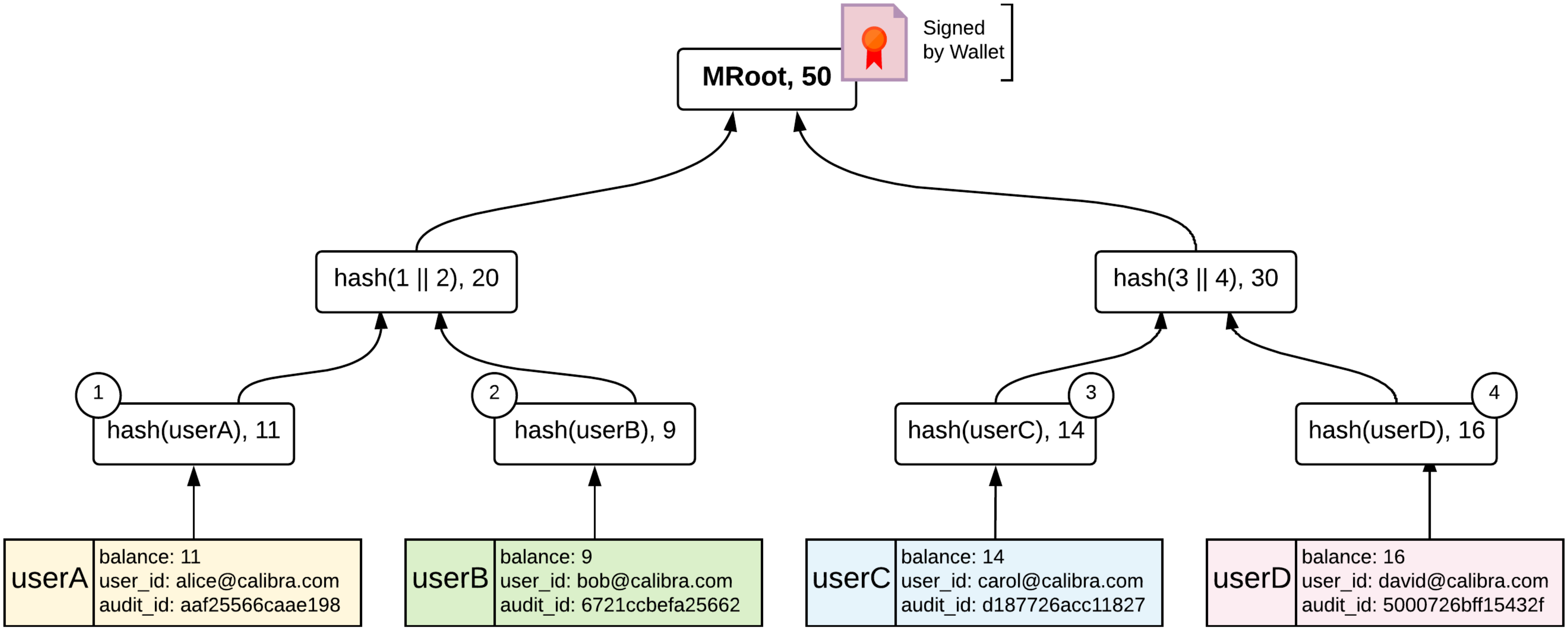
**2014 - Bitstamp proves its Bitcoin reserves to Mike H.**

*"To prove to me the size of the company's deposits, I was given direct MySQL access to their master database"*

**2014 - Bitfinex passes Stefan Thomas's PoSolv Audit**

*"Until we can implement fully zero-knowledge, cryptographically provable audits, you have to trust the auditor, i.e. me, to have done my job correctly"*

# Option C [Summation Merkle Trees]



MRoot, 50

Signed by Wallet

hash(1 || 2), 20

hash(3 || 4), 30

1 hash(userA), 11

2 hash(userB), 9

3 hash(userC), 14

4 hash(userD), 16

userA
balance: 11
user_id: alice@calibra.com
audit_id: aaf25566caae198

userB
balance: 9
user_id: bob@calibra.com
audit_id: 6721ccbefa25662

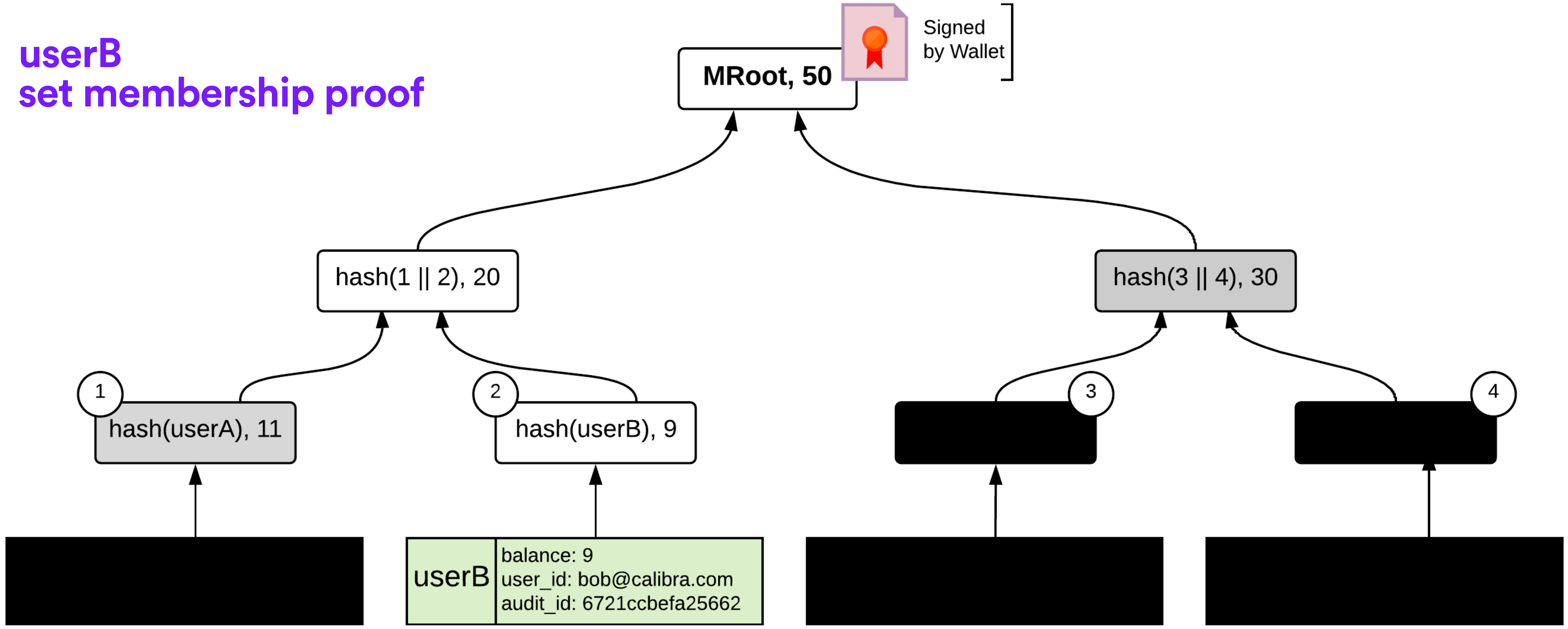
userC
balance: 14
user_id: carol@calibra.com
audit_id: d187726acc11827

userD
balance: 16
user_id: david@calibra.com
audit_id: 5000726bff15432f

# Option C [Summation Merkle Trees]

2/3

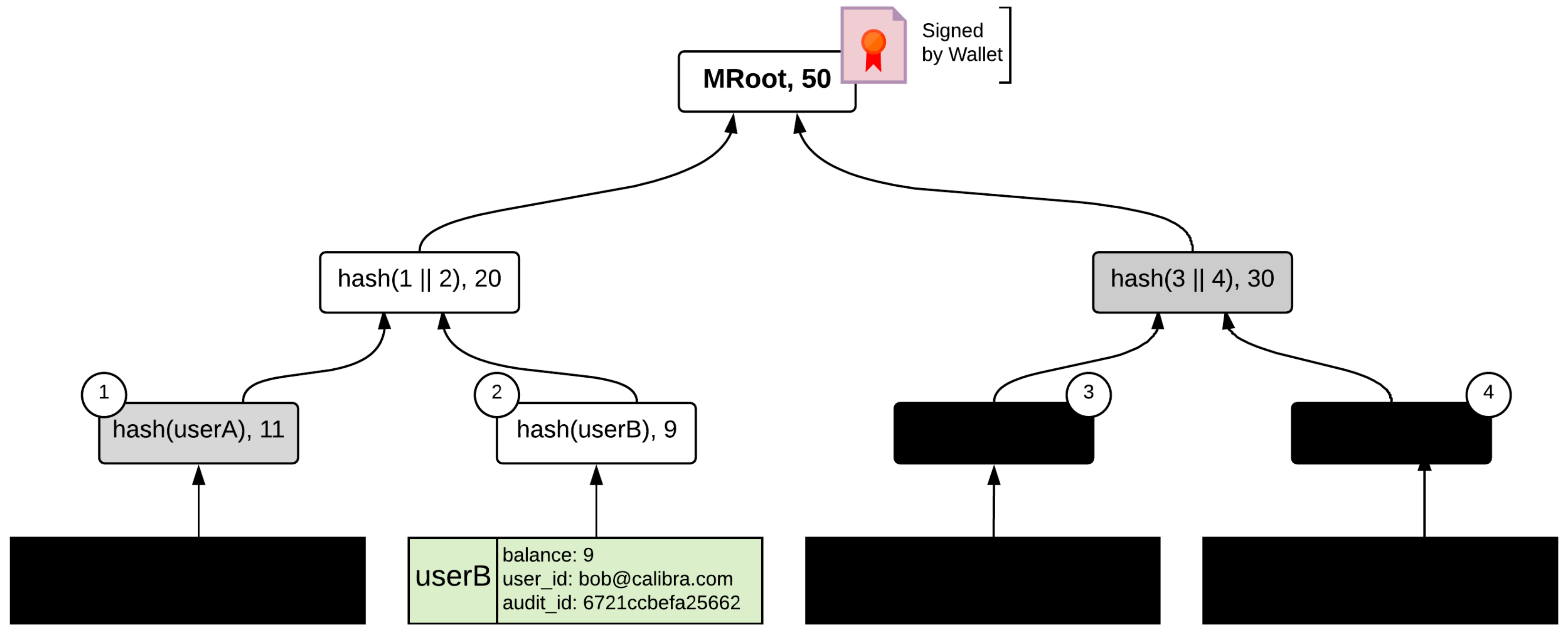
**userB**
**set membership proof**



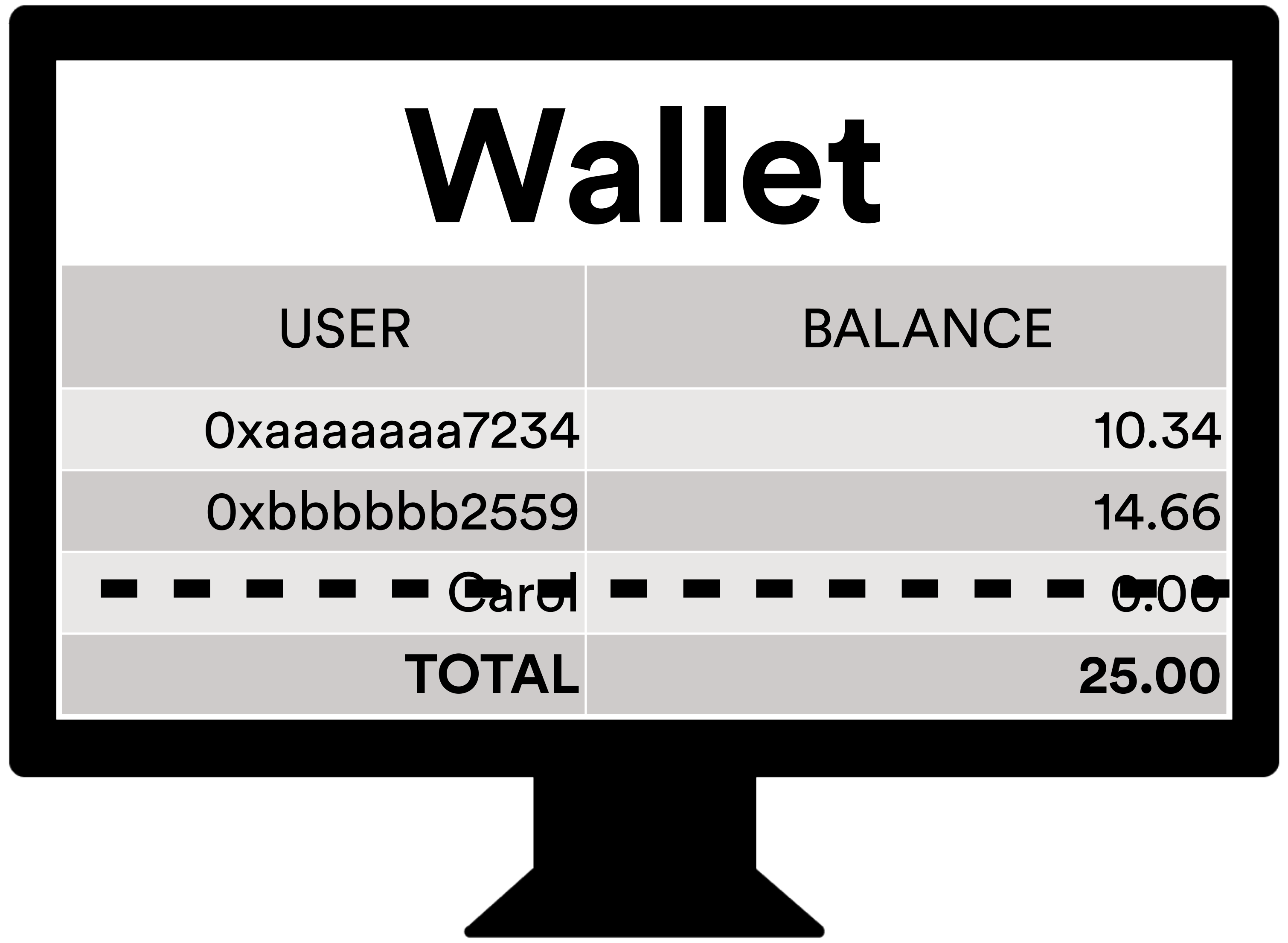
# Option C [Summation Merkle Trees]

3/3

## customer sees



## auditor sees



# Wallet

| USER | BALANCE |
|---|---|
| 0xaaaaaaa7234 | 10.34 |
| 0xbbbbbb2559 | 14.66 |
| ~~Carol~~ | ~~0.00~~ |
| **TOTAL** | **25.00** |

**Expose to auditors**

- individual wallet balances
- number of customers
- leak from multiple PoSolv
- total liabilities

**Expose to customers**

- Merkle path balances
- total liabilities
- number of customers (est)
- wallet performance

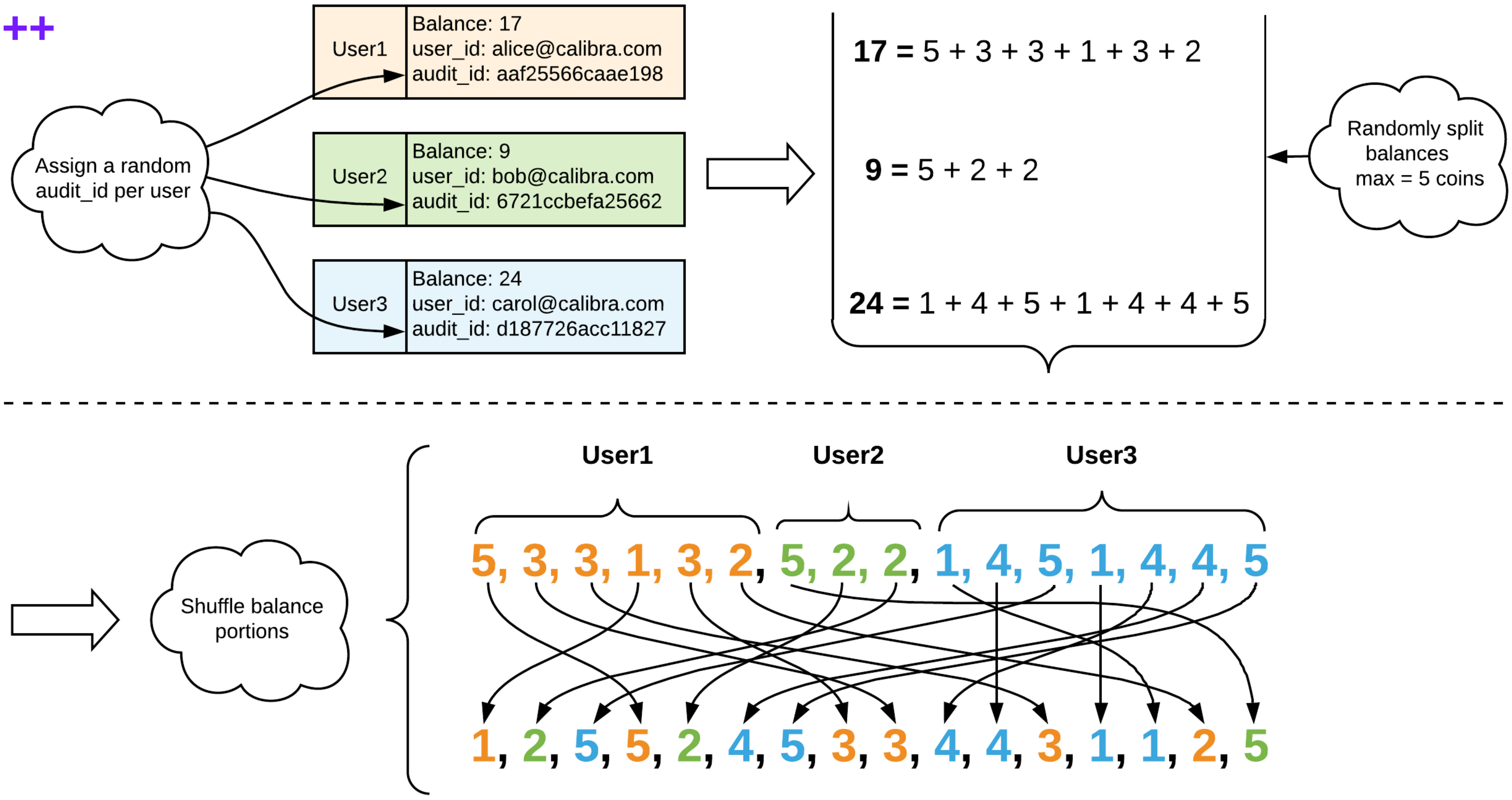
**2018 - ICONOMI is audited by Deloitte**

*"Our goal for the blockchain audit was to prove our solvency and our digital asset holdings using best practices from the traditional financial industry merged with the transparency of the blockchain world"*

# Option D [Random Denomination Trees]

1/3

## Maxwell++



Assign a random audit_id per user

| | |
|---|---|
| User1 | Balance: 17<br>user_id: alice@calibra.com<br>audit_id: aaf25566caae198 |
| User2 | Balance: 9<br>user_id: bob@calibra.com<br>audit_id: 6721ccbefa25662 |
| User3 | Balance: 24<br>user_id: carol@calibra.com<br>audit_id: d187726acc11827 |

Randomly split balances max = 5 coins

**17 =** 5 + 3 + 3 + 1 + 3 + 2

**9 =** 5 + 2 + 2

**24 =** 1 + 4 + 5 + 1 + 4 + 4 + 5

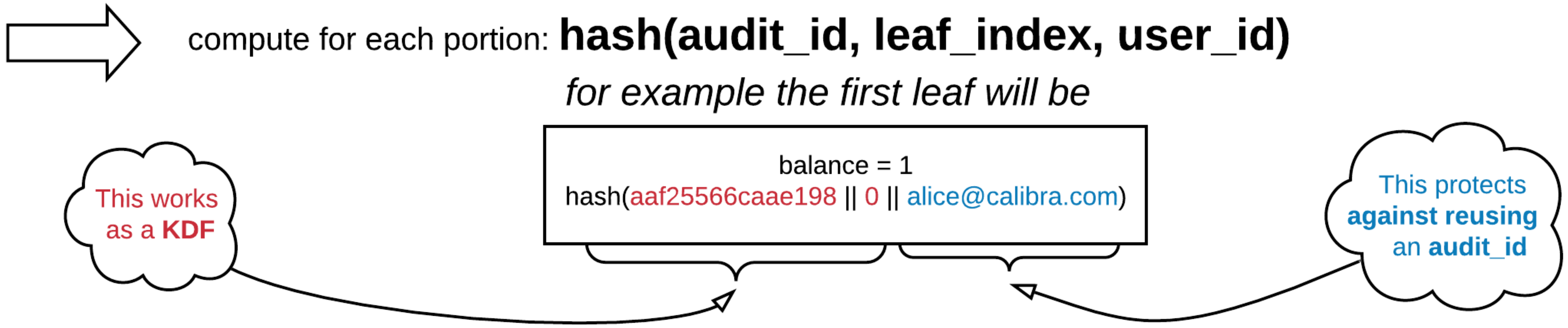
Shuffle balance portions

**User1** **User2** **User3**

5, 3, 3, 1, 3, 2, 5, 2, 2, 1, 4, 5, 1, 4, 4, 5

1, 2, 5, 5, 2, 4, 5, 3, 3, 4, 4, 3, 1, 1, 2, 5

# Option D [Random Denomination Trees]



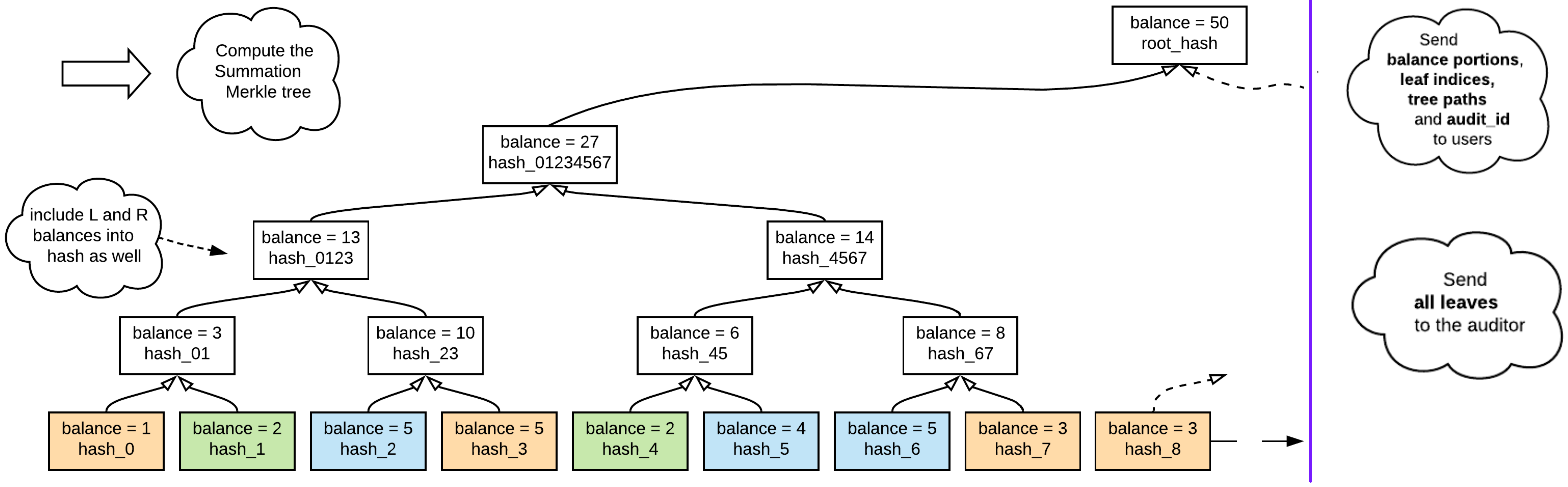
compute for each portion: **hash(audit_id, leaf_index, user_id)**

*for example the first leaf will be*

balance = 1
hash(aaf25566caae198 || 0 || alice@calibra.com)

This works as a **KDF**

This protects **against reusing** an **audit_id**



Compute the Summation Merkle tree

include L and R balances into hash as well

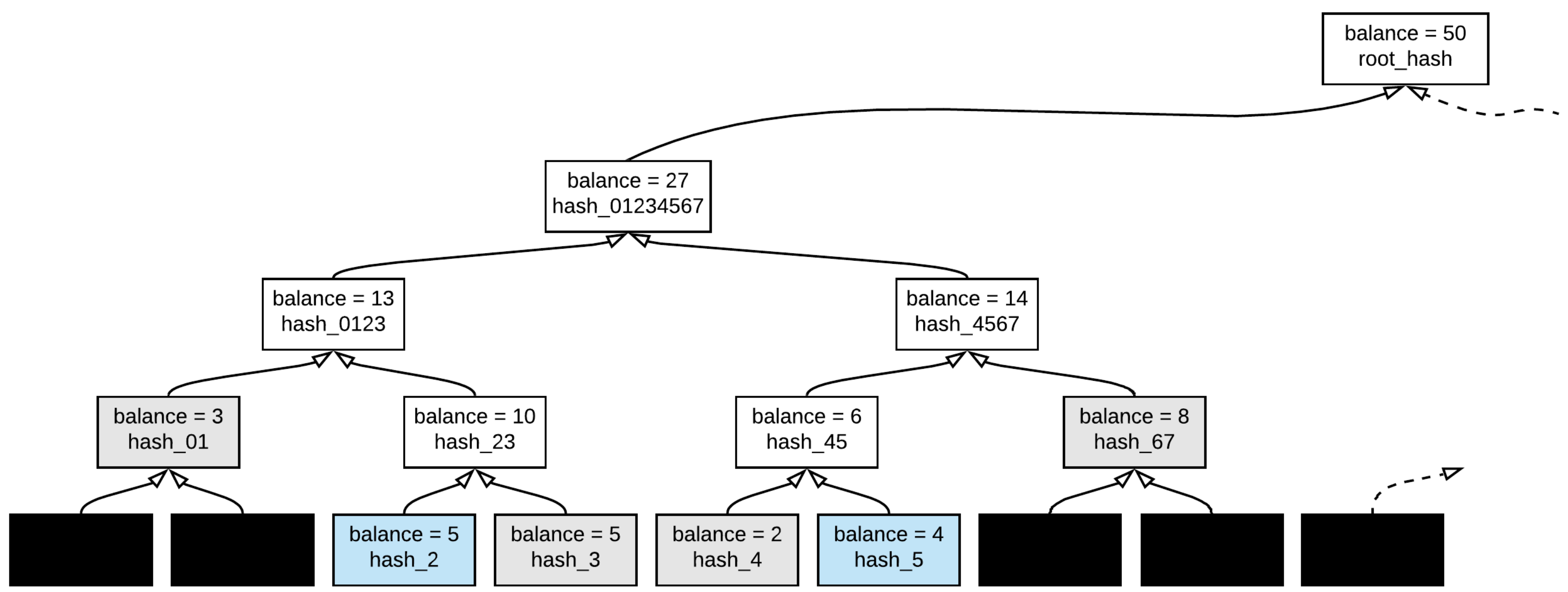
Send **balance portions, leaf indices, tree paths** and **audit_id** to users

Send **all leaves** to the auditor

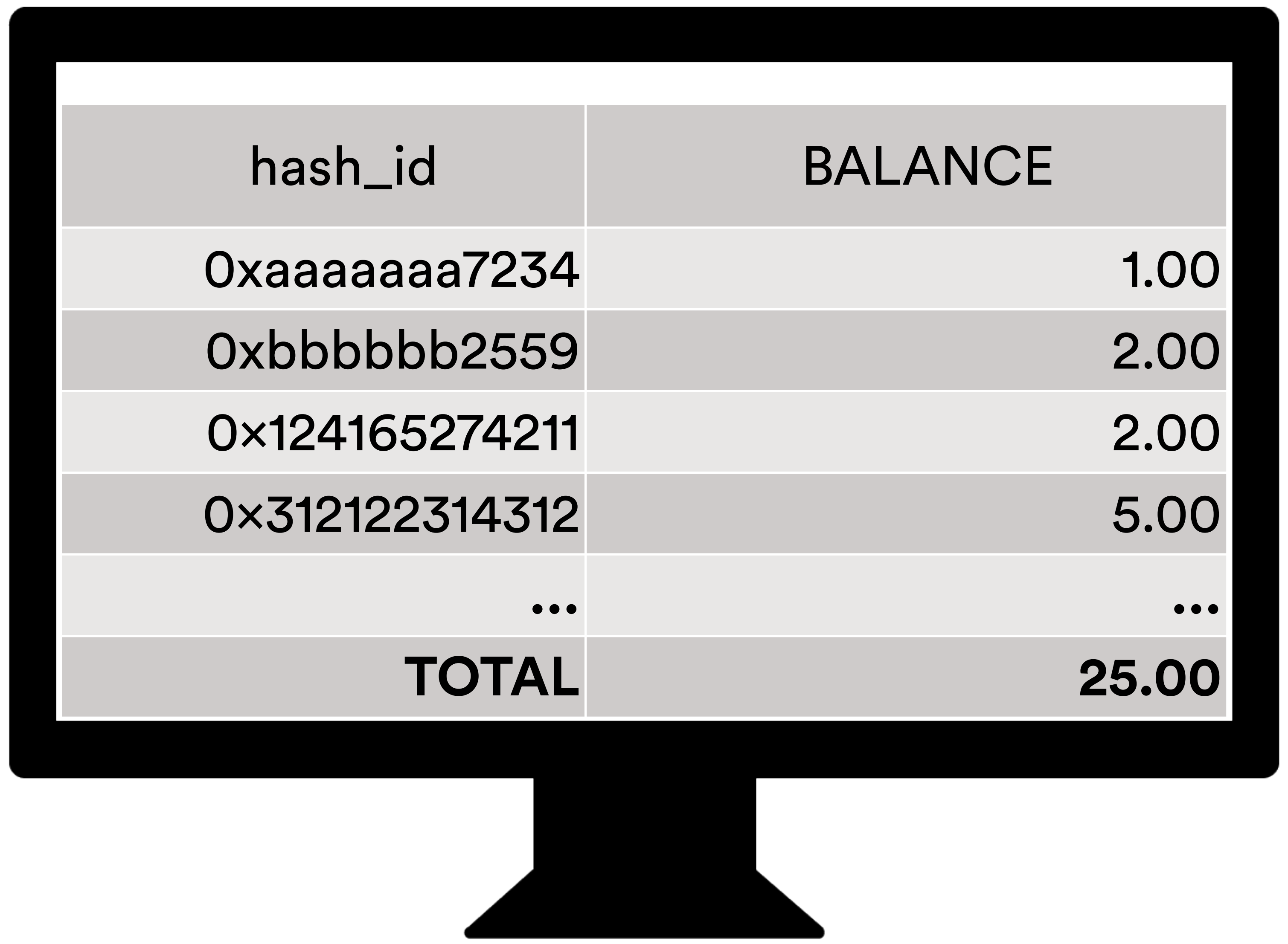
# Option D [Random Denomination Trees]

3/3

## customer sees



## auditor sees



| hash_id | BALANCE |
|---|---|
| 0xaaaaaaa7234 | 1.00 |
| 0xbbbbbb2559 | 2.00 |
| 0x124165274211 | 2.00 |
| 0x312122314312 | 5.00 |
| ... | ... |
| **TOTAL** | **25.00** |

**Expose to auditors**

- ~~individual wallet balances~~
- ~~number of customers~~
- ~~leak from multiple PoSolv~~
- total liabilities
- denominations distribution

**Expose to customers**

- ~~Merkle path balances~~
- total liabilities
- ~~number of customers~~
- ~~wallet performance~~ ???

# Option E [Remotely Attestable Secure Processors]

**Intel SGX, Apple SEP, Gradient, Keystone**

Use remote attestation to prove that a specific piece of code ran on a suitable secure enclave

**WALLET INPUTS**

- balance & hash **for non-zero in-wallet accounts**
- list of **all (or some)** active blockchain addresses & balances
- **proofs of key ownership**

**ENCLAVE LOGIC**

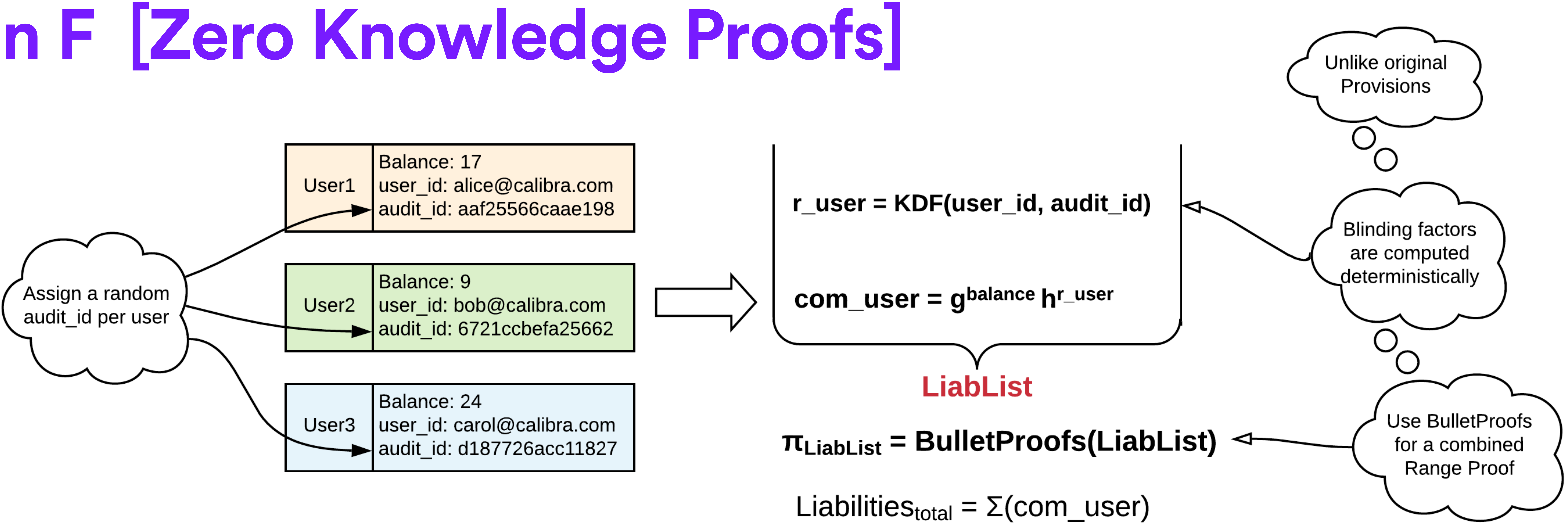
- compute Merkle roots
- check *all balances > 0*   &&   *liabilities <= assets*
- verify key(s) ownership
- sign(Liab_MRoot, Addresses_MRoot, result)

- alternative to ZKP using secure hardware
- normally, nothing is exposed
- customizable and fast
- need to add noise (i.e. zero balance accounts to hide number of customers and keys)

- stateful enclaves
- side channels
- trust hardware vendors
- decapping attacks

# Option F [Zero Knowledge Proofs]

1/4



Assign a random audit_id per user
Unlike original Provisions
Blinding factors are computed deterministically
Use BulletProofs for a combined Range Proof

| | |
|---|---|
| User1 | Balance: 17<br>user_id: alice@calibra.com<br>audit_id: aaf25566caae198 |
| User2 | Balance: 9<br>user_id: bob@calibra.com<br>audit_id: 6721ccbefa25662 |
| User3 | Balance: 24<br>user_id: carol@calibra.com<br>audit_id: d187726acc11827 |

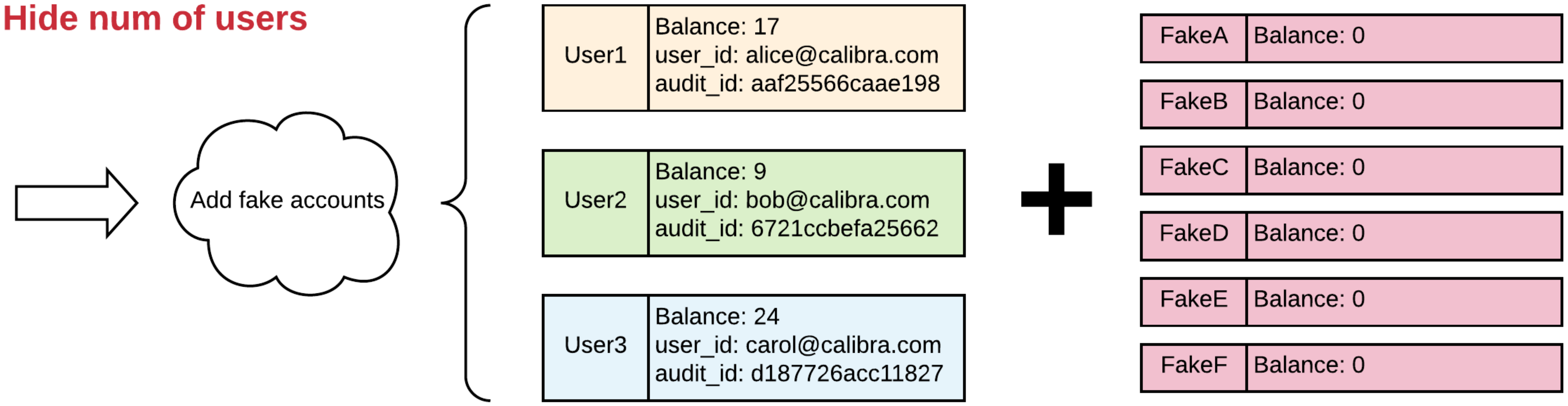
$$\underbrace{\begin{array}{l} \text{r\_user} = \text{KDF(user\_id, audit\_id)} \\ \text{com\_user} = g^{\text{balance}} h^{\text{r\_user}} \end{array}}_{\text{LiabList}}$$

$$\pi_{\text{LiabList}} = \text{BulletProofs(LiabList)}$$

$$\text{Liabilities}_{\text{total}} = \Sigma(\text{com\_user})$$

## EXTENSION A

### Hide num of users



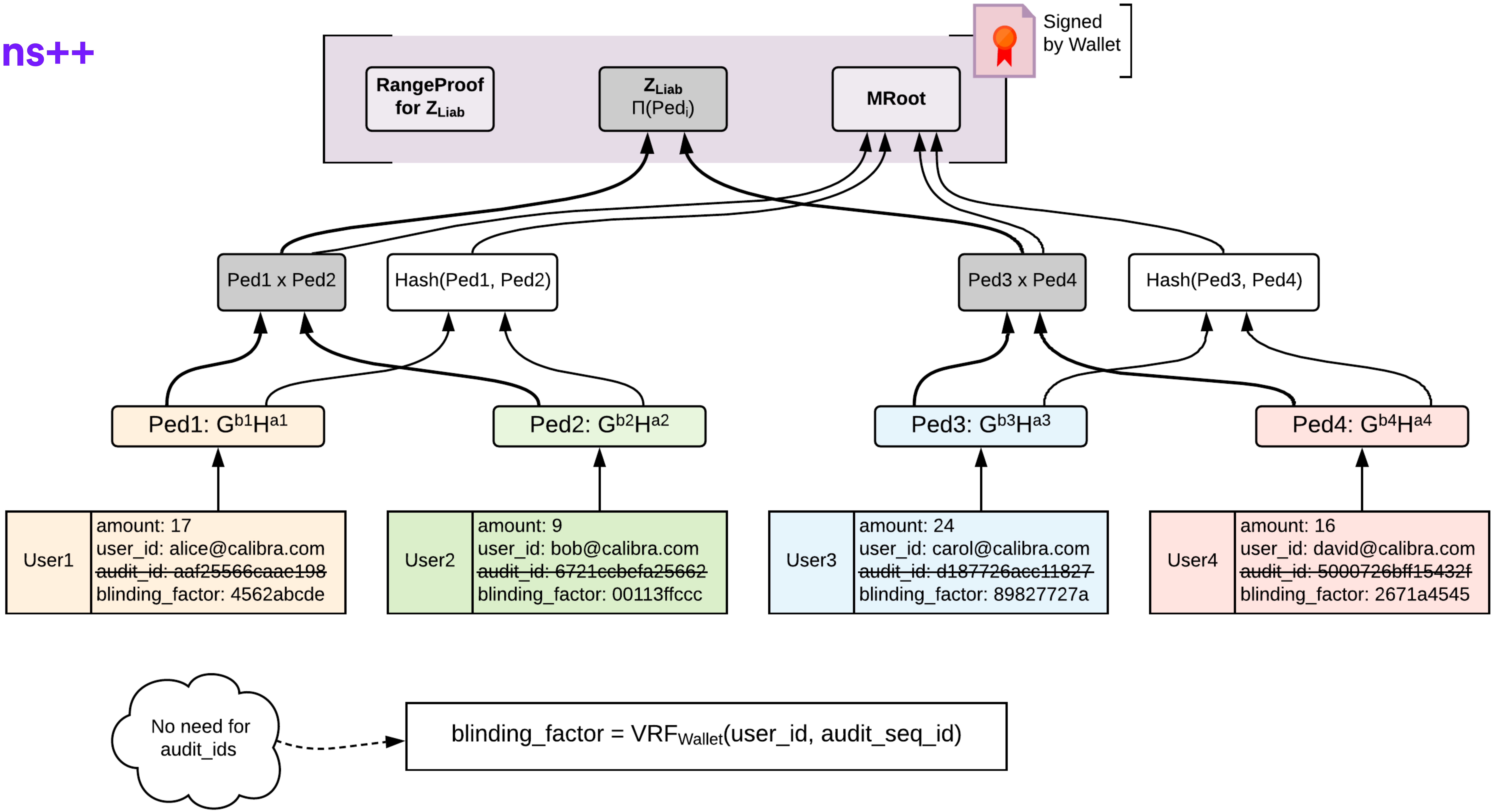
| | |
|---|---|
| User1 | Balance: 17<br>user_id: alice@calibra.com<br>audit_id: aaf25566caae198 |
| User2 | Balance: 9<br>user_id: bob@calibra.com<br>audit_id: 6721ccbefa25662 |
| User3 | Balance: 24<br>user_id: carol@calibra.com<br>audit_id: d187726acc11827 |

+

| | |
|---|---|
| FakeA | Balance: 0 |
| FakeB | Balance: 0 |
| FakeC | Balance: 0 |
| FakeD | Balance: 0 |
| FakeE | Balance: 0 |
| FakeF | Balance: 0 |

# Option F [Zero Knowledge Proofs]

## Provisions++



User1
amount: 17
user_id: alice@calibra.com
~~audit_id: aaf25566caac198~~
blinding_factor: 4562abcde

User2
amount: 9
user_id: bob@calibra.com
~~audit_id: 6721ccbcfa25662~~
blinding_factor: 00113ffccc

User3
amount: 24
user_id: carol@calibra.com
~~audit_id: d187726acc11827~~
blinding_factor: 89827727a

User4
amount: 16
user_id: david@calibra.com
~~audit_id: 5000726bff15432f~~
blinding_factor: 2671a4545

$\text{blinding\_factor} = \text{VRF}_{\text{Wallet}}(\text{user\_id}, \text{audit\_seq\_id})$

# Option F [Zero Knowledge Proofs]

3/4

I know $\text{key}_i$
and $p_i$ is commitment to $b_i$
OR
$p_i$ is commit(0)

## Provisions Proofs of Assets

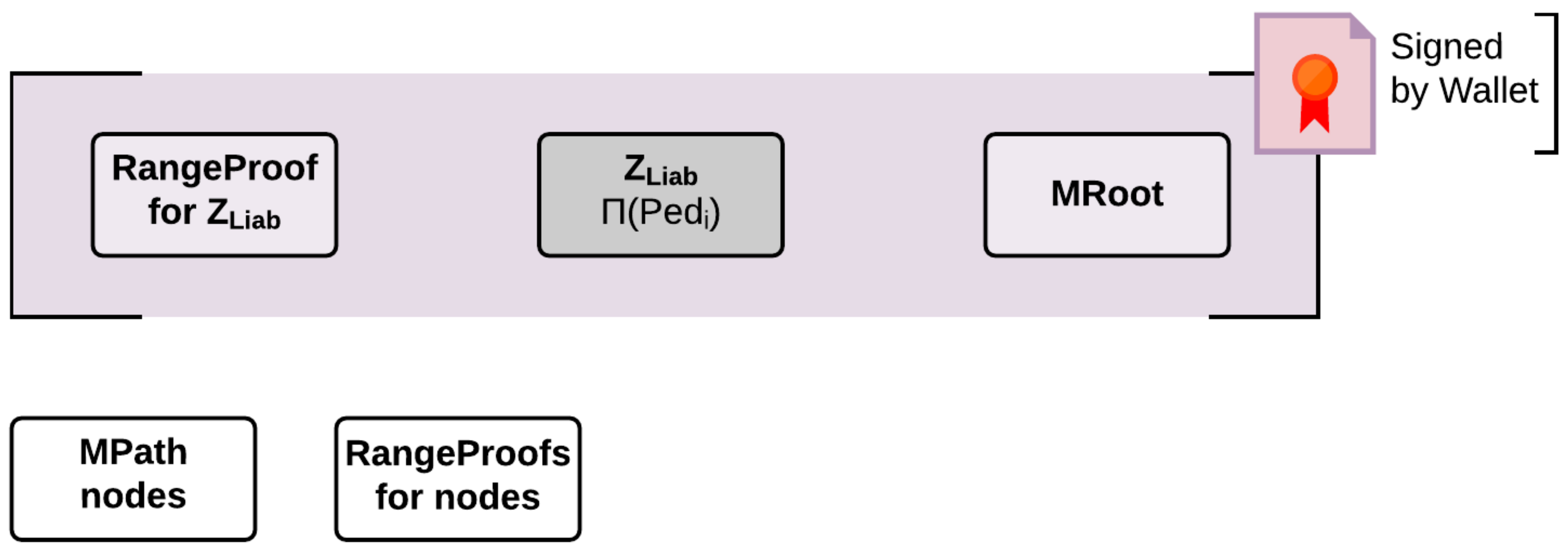
| address | private key | public balance | Pedersen commitment | proof |
|---|---|---|---|---|
| **PK1** | **key1** | **b1 = 20** | **p1 = commit(20)** | |
| PK2 | key2 | b2 = 30 | p2 = commit(0) | |
| PK3 | key3 | b3 = 30 | p3 = commit(0) | |
| **PK4** | **key4** | **b4 = 10** | **p4 = commit(10)** | |
| PK5 | key5 | b5 = 10 | p5 = commit(0) | |

# Option F [Zero Knowledge Proofs]

4/4

## customer sees

blinding_factor = $\text{VRF}_{\text{Wallet}}$(user_id, audit_seq_id)



RangeProof for $Z_{\text{Liab}}$

$Z_{\text{Liab}}$ $\Pi(\text{Ped}_i)$

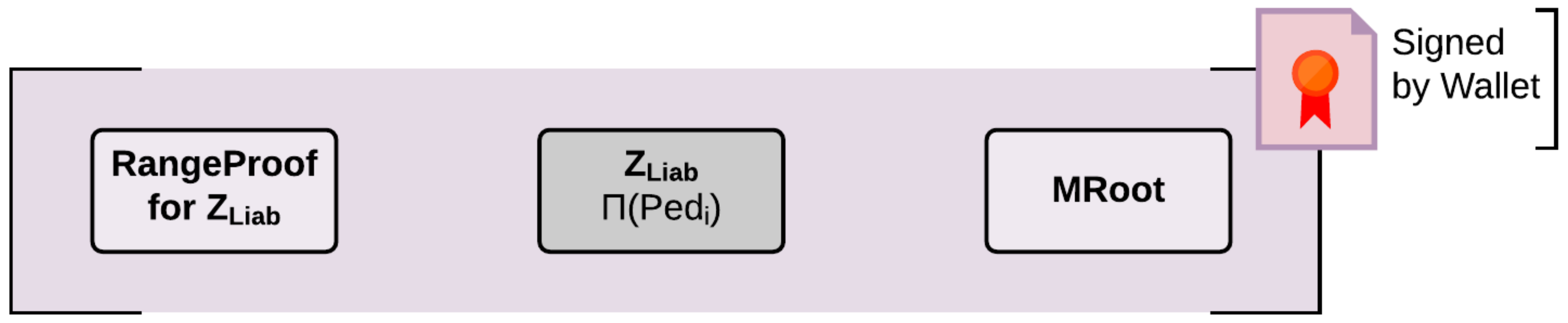
MRoot

Signed by Wallet

MPath nodes

RangeProofs for nodes

## auditor sees



RangeProof for $Z_{\text{Liab}}$

$Z_{\text{Liab}}$ $\Pi(\text{Ped}_i)$

MRoot

Signed by Wallet

**Expose to auditors**

- upper bound for number of customers

**Expose to customers**

- upper bound for number of customers

# Option G [Differential Privacy Guarantees]

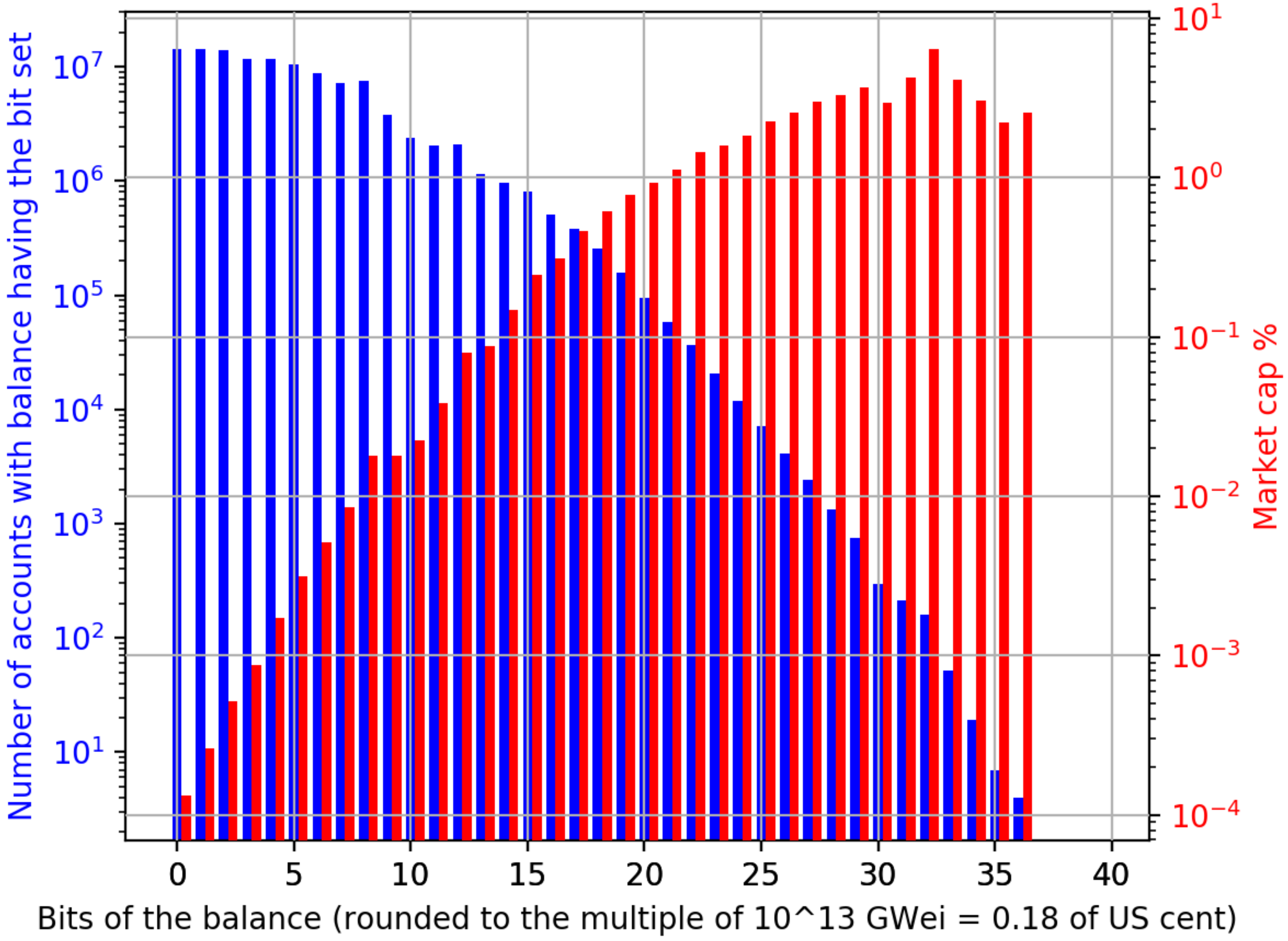
### Formally reason about the privacy gain

Guarantee that for any user in the universe, the auditor cannot tell whether their account was part of the proof of liabilities or not except with some a-priori probability.

**Accumulator-based**

- decompose balances (i.e., powers of two)
- maybe set a cap (i.e., up to *2^20*)
- add **positive** private **noise** only (*Laplace* or *Gaussian*)

### Ethereum account balances distribution



# Hybrid Solutions?
## for performance

### A. Provisions++ for large balances only

- reduce the amount of DP noise (extra money)
- limit the use of expensive range-proofs for a much smaller set of account balances

### B. Provisions++ for DP noise

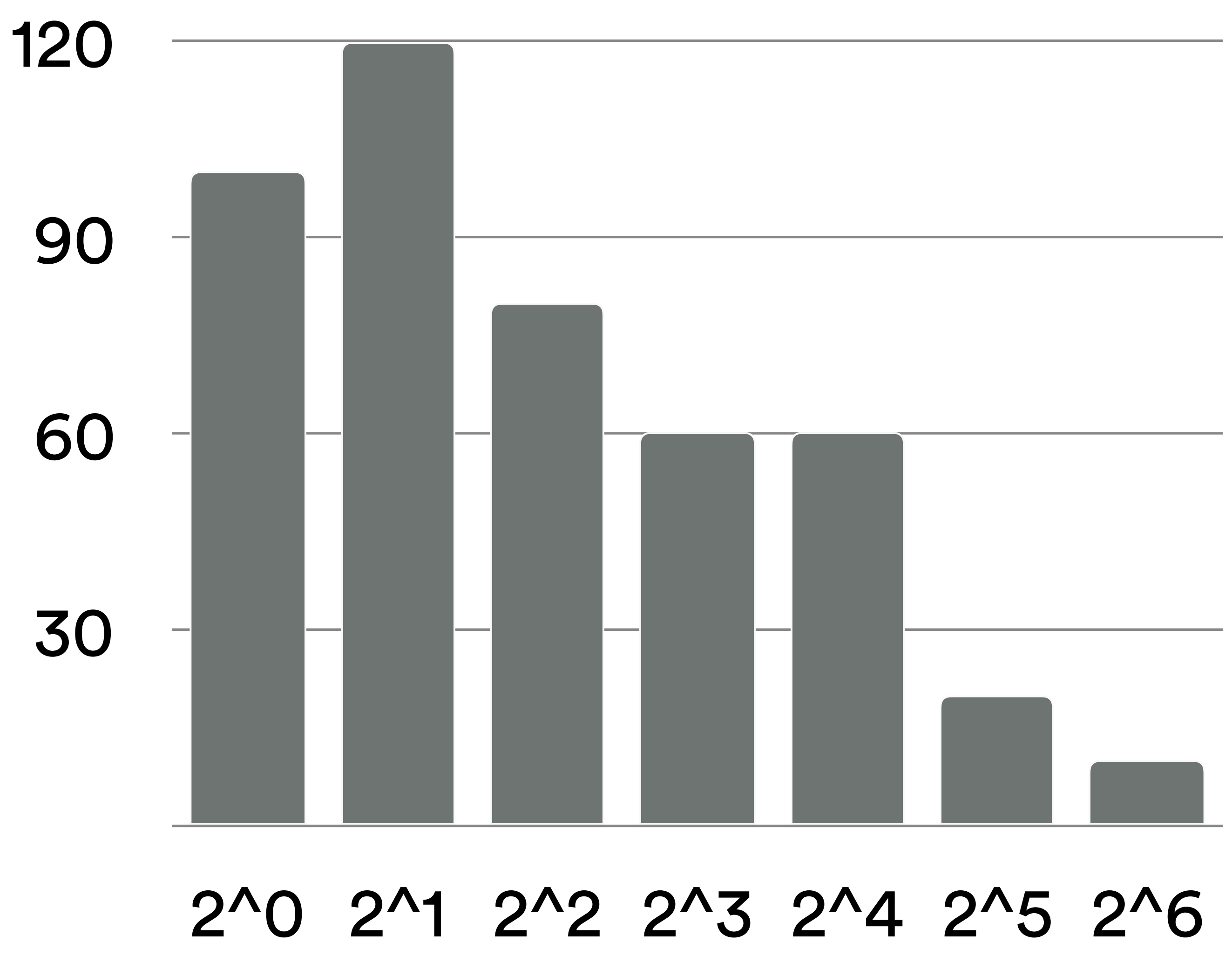
- keep negative noise
- but, move negative chunks to range proofs
- less extra money (even zero)
- positive noise can be accommodated by ZKP large balances

### C. flat distribution?

- instead of running DP directly, move chunks from larger to smaller denominations until we get a flat distribution
- special case: put everything into the 1st bucket (size issues)

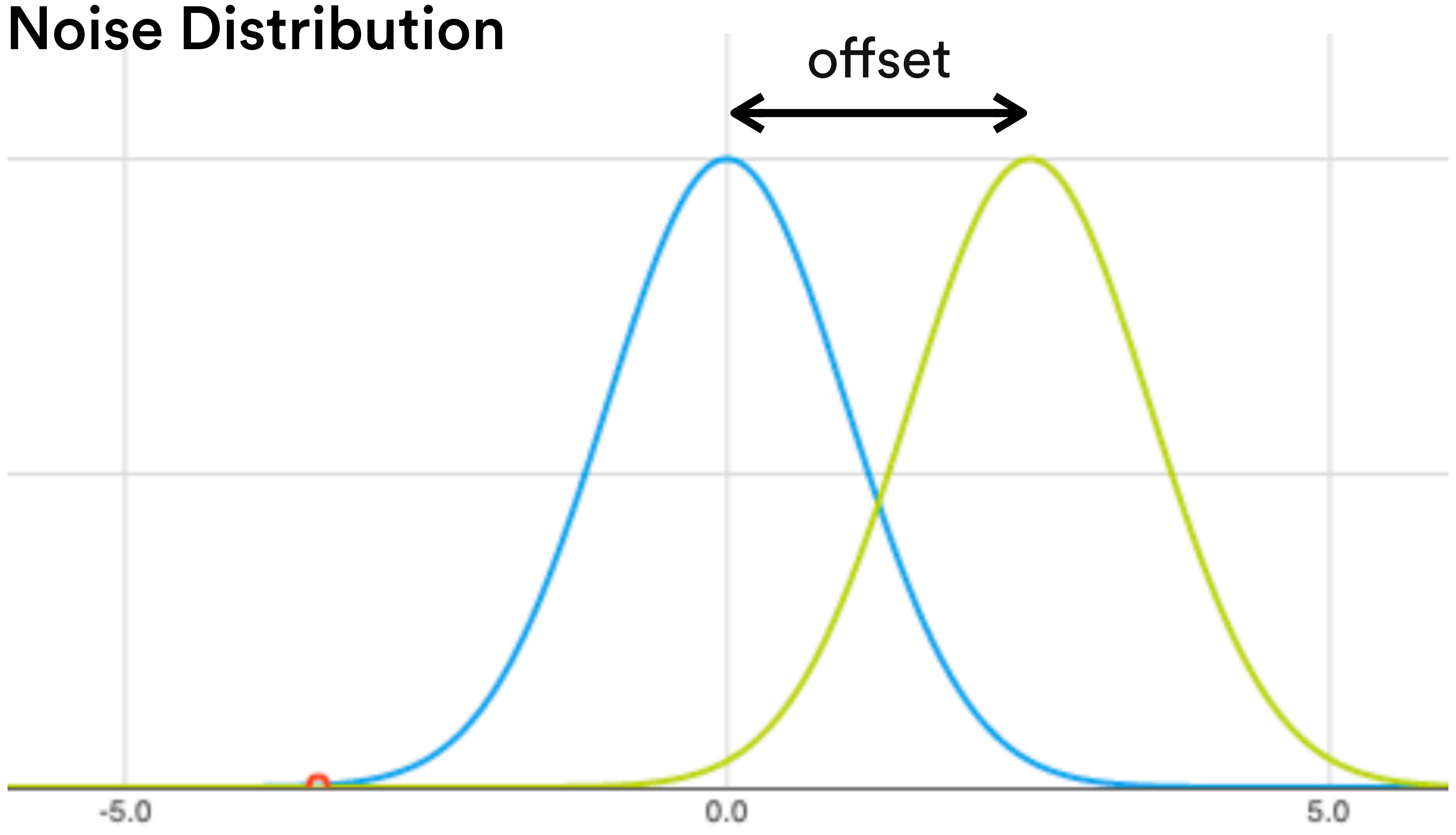
# Accumulator-based PoSolv

Original Balance Distribution

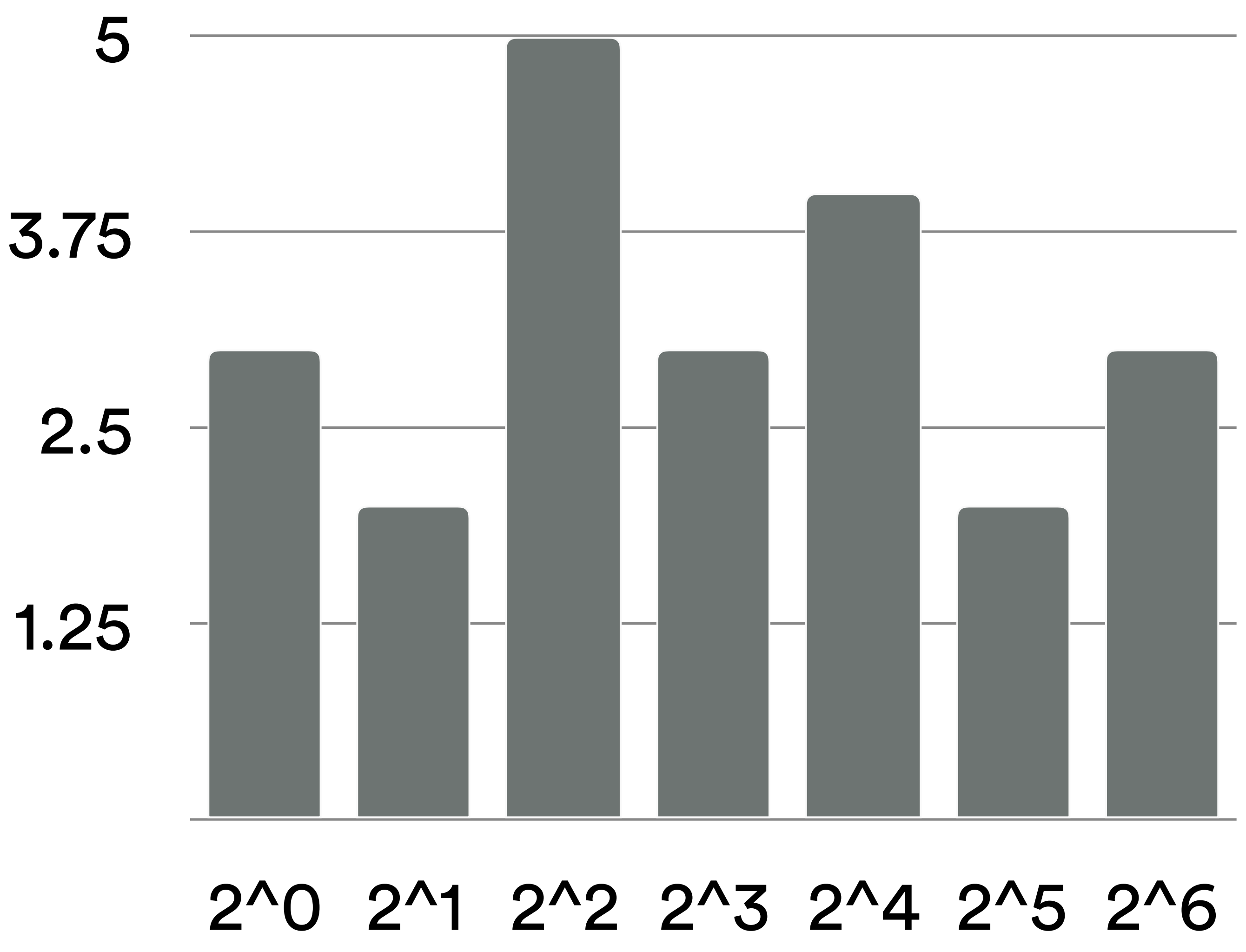


sample positive noise

Noise Distribution

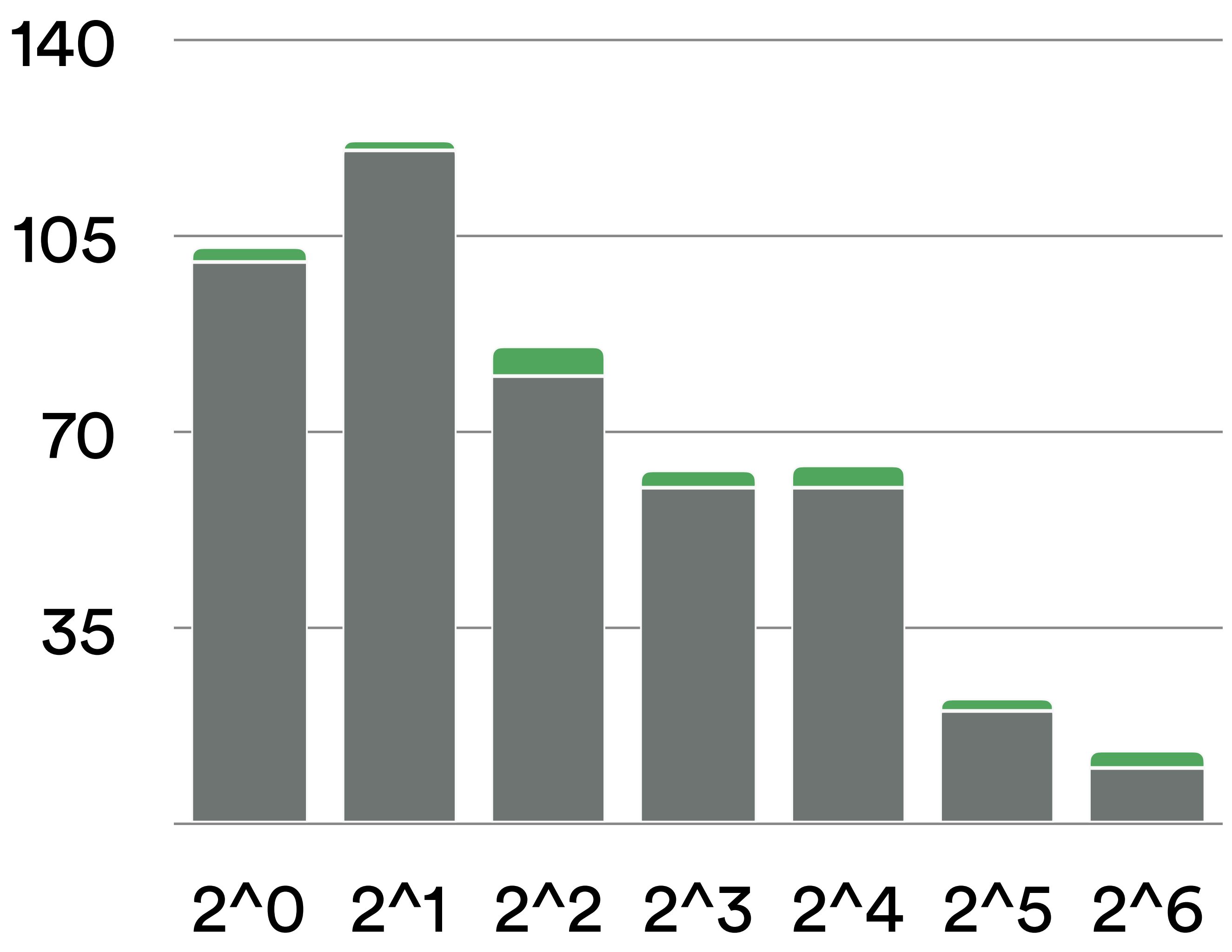


DP noise



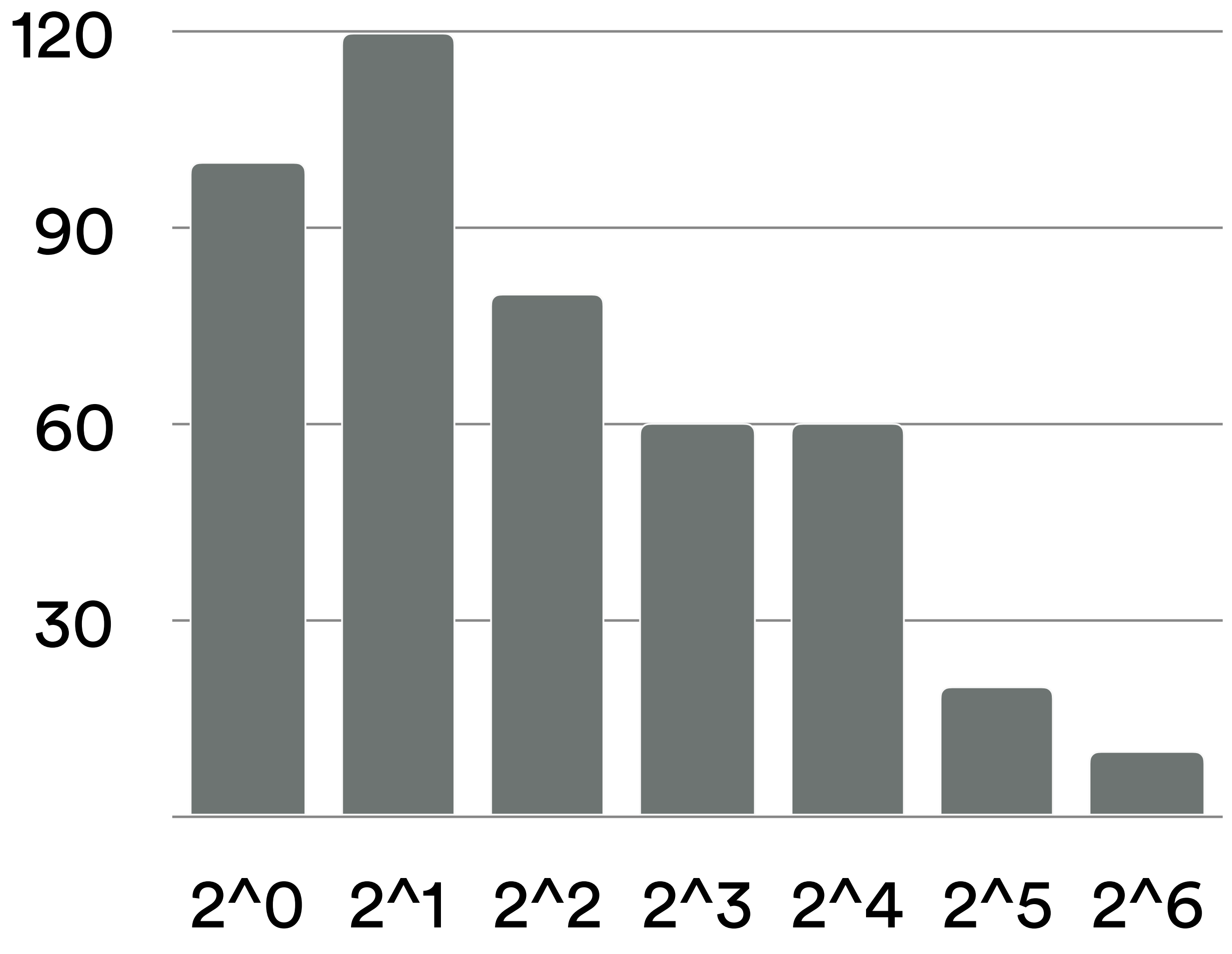
apply noise

Distribution after DP



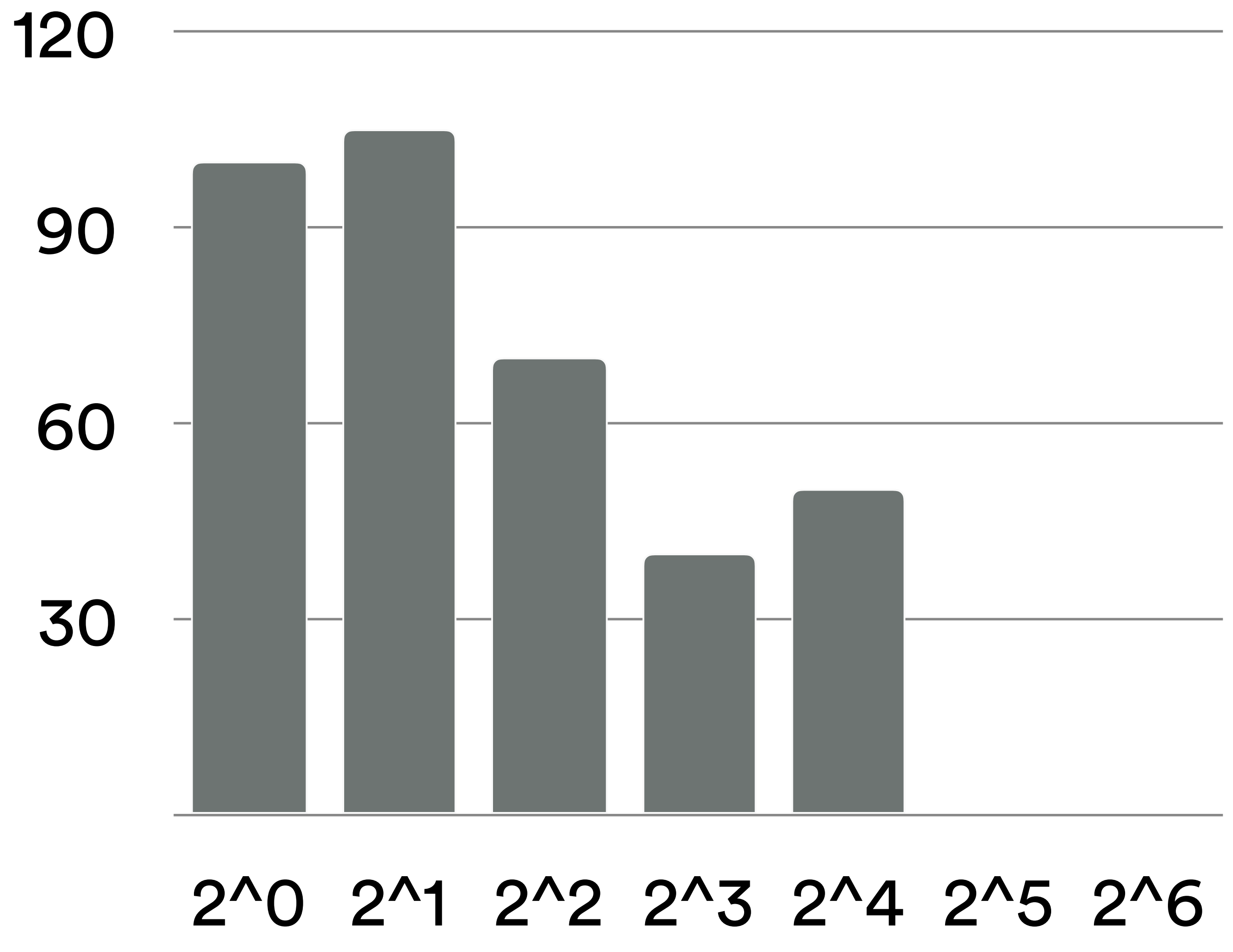
# ZKP as a tool for DP

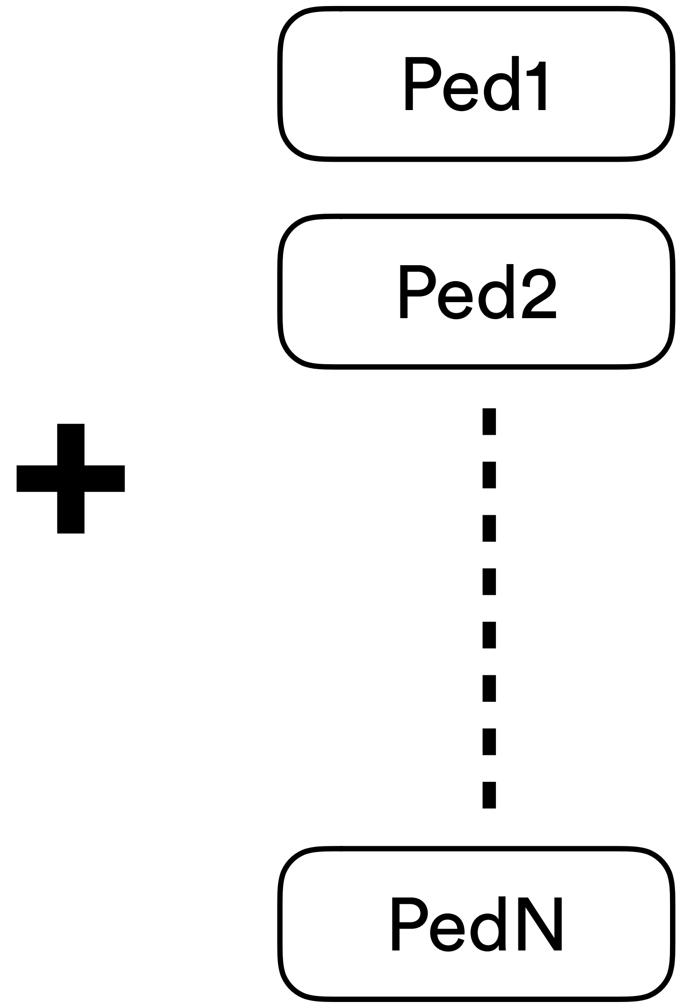
Original Balance Distribution



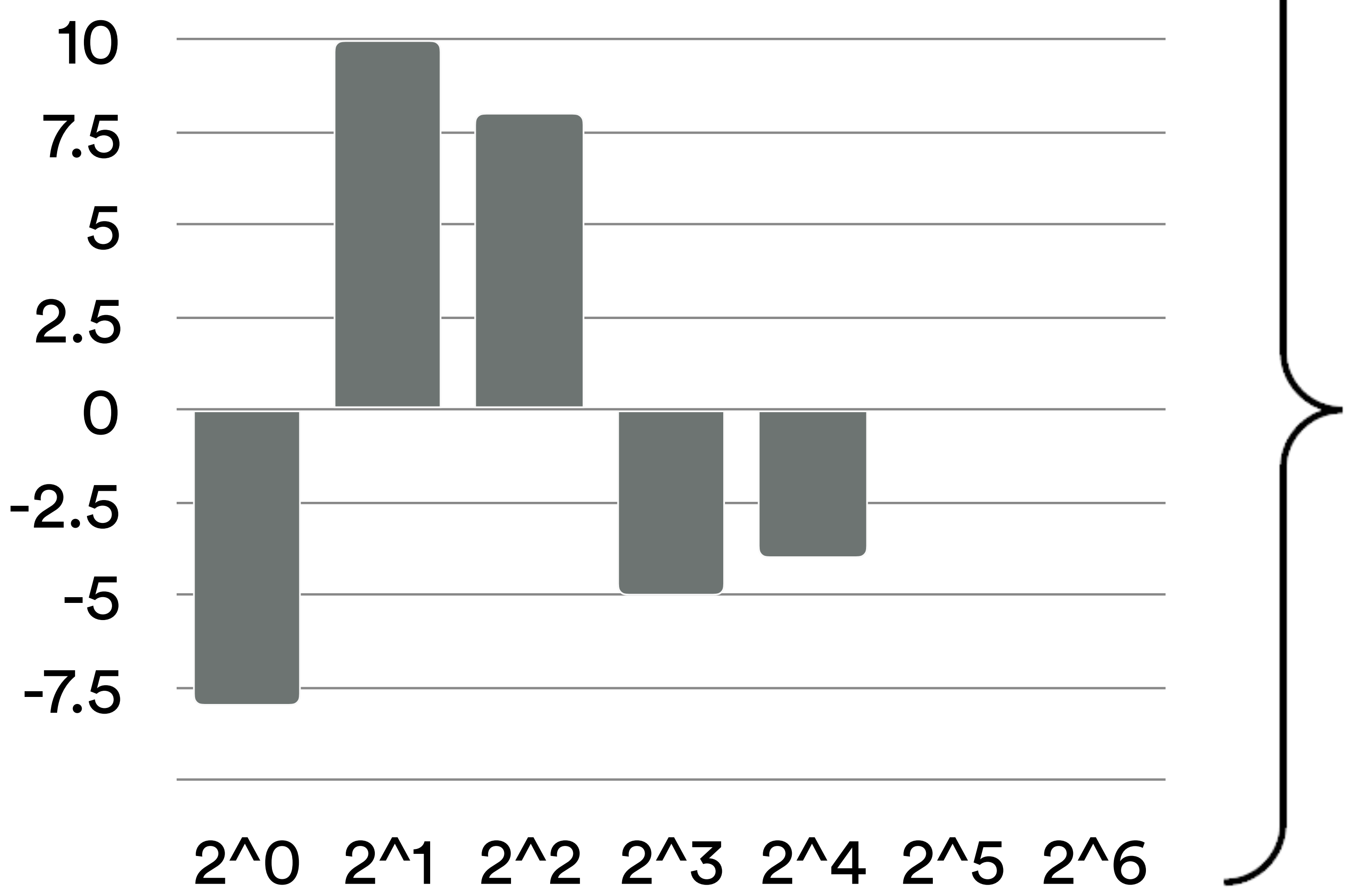
move large balances (and others randomly) to Ped commitments

Distribution after balance removal

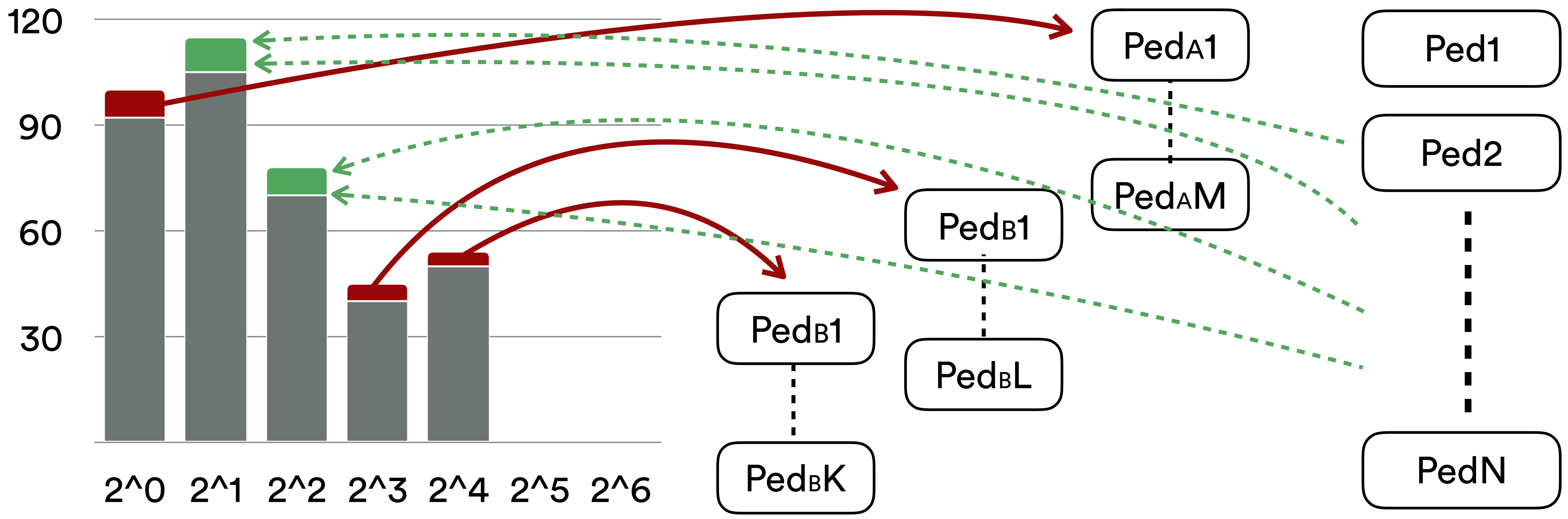




DP noise



Distribution after DP



# Limitations and open problems

## ZKP related

- circuit based zkSNARKs to support hashed key addresses for proofs of assets
- who runs the trusted setup (if required)?
- multi-sig addresses and custom scripts
- locked funds (payment channels & atomic swaps)

# Limitations and open problems

## Process related

- frequency of audits
- proof of non-collusion (how to sync)
- dispute resolution (cryptographic evidence)
- ability to spend ≠ willingness to pay
- eventual Vs immediate solvency
- challenge-response protocol to prove ownership
- auditor sampling

# Limitations and open problems

## Misc

- HSM / cold wallets (are valet keys enough?)

- risk-free collusion in payment channels

- level of privacy Vs efficiency (hybrid schemes)

- privacy-preserving cryptocurrencies

- multi-asset blockchains

# Thank you

## Privacy Preserving Proofs of Solvency

contact: Kostas Chalkias

kostascrypto@calibra.com