DAPOL

Distributed Auditing Proofs of Liabilities

Konstantinos Chalkias
Kevin Lewi
Payman Mohassel
Valeria Nikolaenko

calibra

Proposed standard

ZKProof 2020

Problem statement

a particular class of auditing cases, where the audited entity does NOT have any incentive to increase its liabilities or obligations.

Applications

-- at least 3 major categories & 15 domains --

a) finance (i.e., fundraising, solvency)

b) e-voting (i.e., disapproval voting)

c) official reports (i.e., number of COVID-19 daily cases)

Recommended approach

allow users to privately verify inclusion of their balance / vote / case in the reported total.

Purpose

Agree on a scheme to be used as a reference for the development of auditing technology that is transparent, secure, interoperable and which supports selective privacy-preserving properties.

Ideally applicable to as many domains as possible, even as a complementary tool to traditional auditing.

Aim

Fix this:

- *industry inconsistency on the auditing tools (especially in proofs of solvency),*

- oftentimes auditors need to re-implement the algorithms proposed by the audited companies (provers) as there is not a common reference implementation for them,

- it is clear that outdated weaker privacy-preserving techniques are still widely used (i.e., the original scheme of Maxwell).

Pragmatic expectations

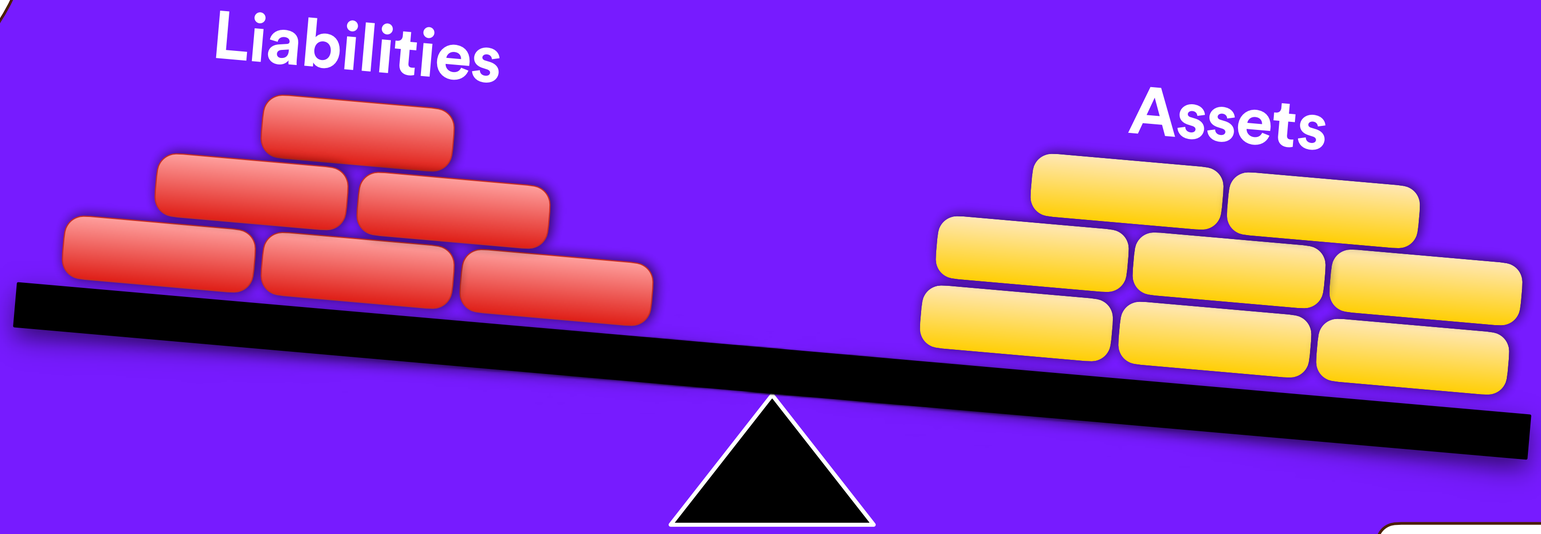
- Agree on one or at most two concrete designs (specific ZKP scheme).
  This is very important to avoid delays in the adoption of the proposed solution.
- Have a reference PoC implementation and test vectors.
- Consider it as a benchmarking use case when comparing ZKP protocols.
  DAPOL requires efficient range proofs, set membership and proof aggregation.

Non-goals

- In some contexts, extra security features might required *(i.e., defend against coercion and bribery in e-voting)*. These are out scope.
- Specific signature schemes *(required for dispute resolution)* and private information retrieval (PIR) methods *(required for private proof download)* won't be addressed.

Proof of Solvency

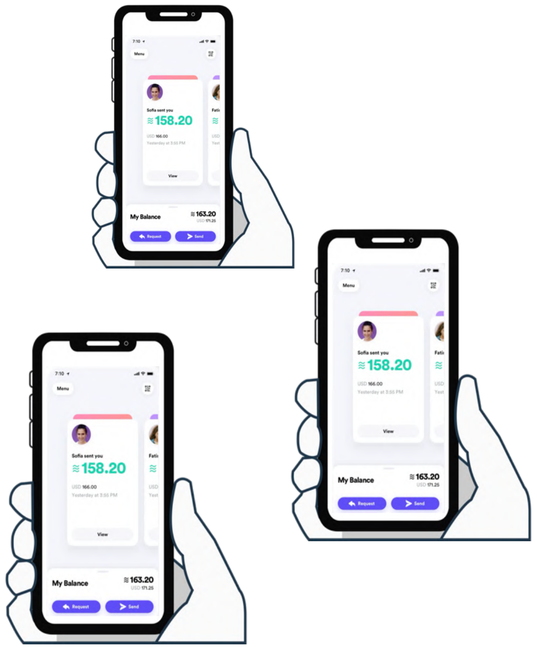
Lower the value of **Solvency Ratio** indicates a greater **probability of default** on the debt obligations



Ensure

Liabilities <= Assets

Proofs of Solvency



UserA: **100**
UserB: **250**
UserC: **0**
..
..
UserZ: **500**

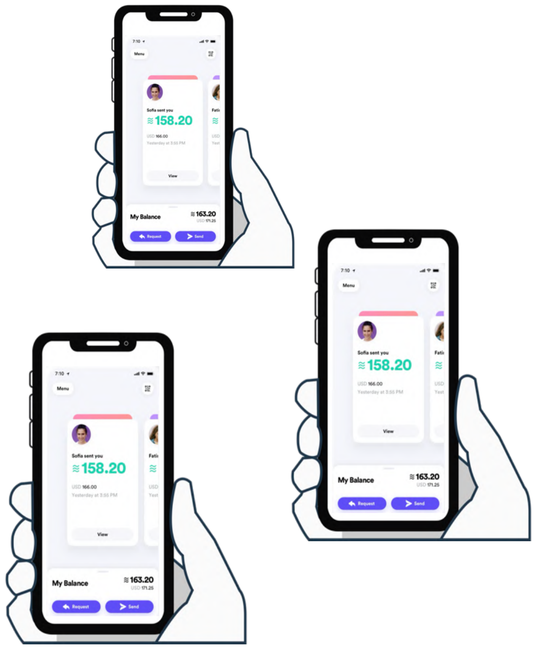
Liabilities

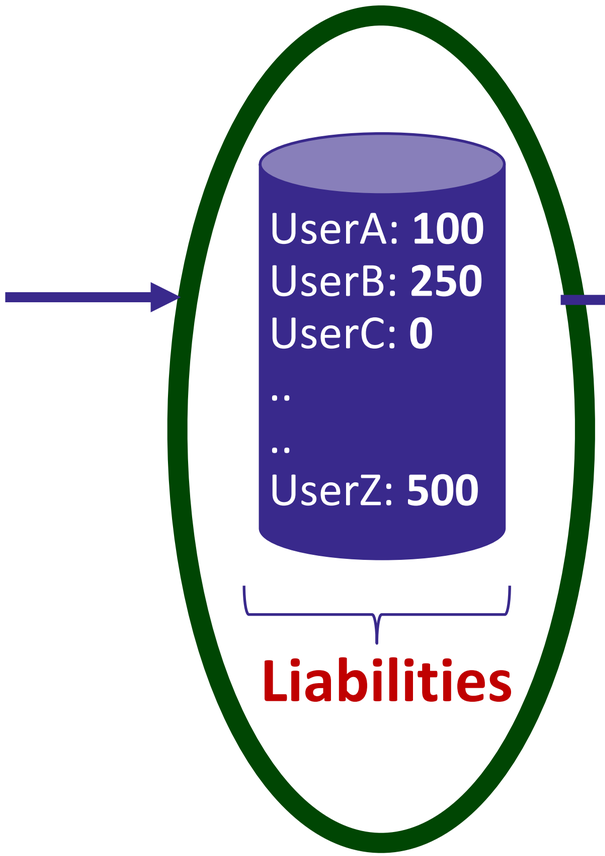
Blockchain

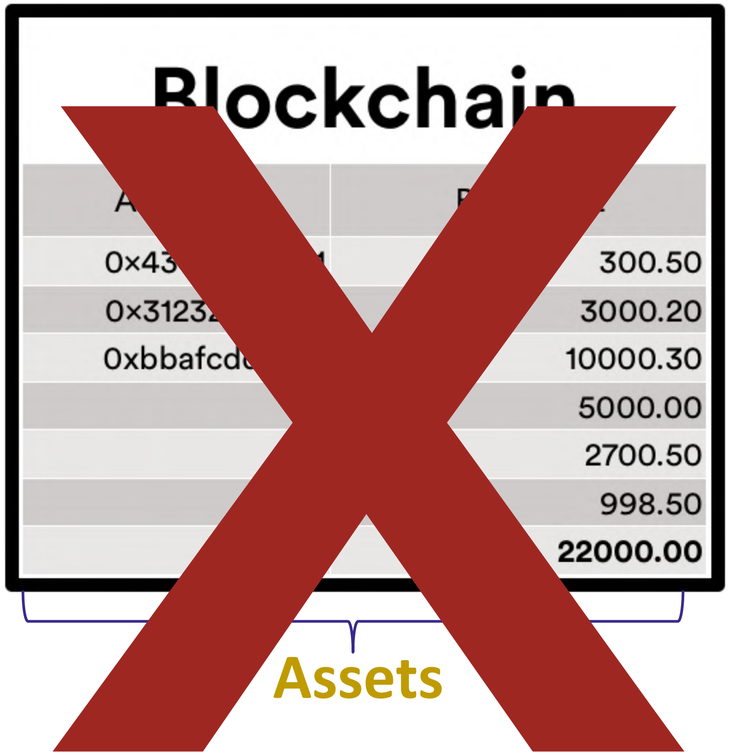
ADDRESS	BALANCE
0×434aaba2151	300.50
0×312323441aa	3000.20
0xbbafcddd1aa	10000.30
...	5000.00
...	2700.50
...	998.50
TOTAL	**22000.00**

Assets

Proofs of Solvency

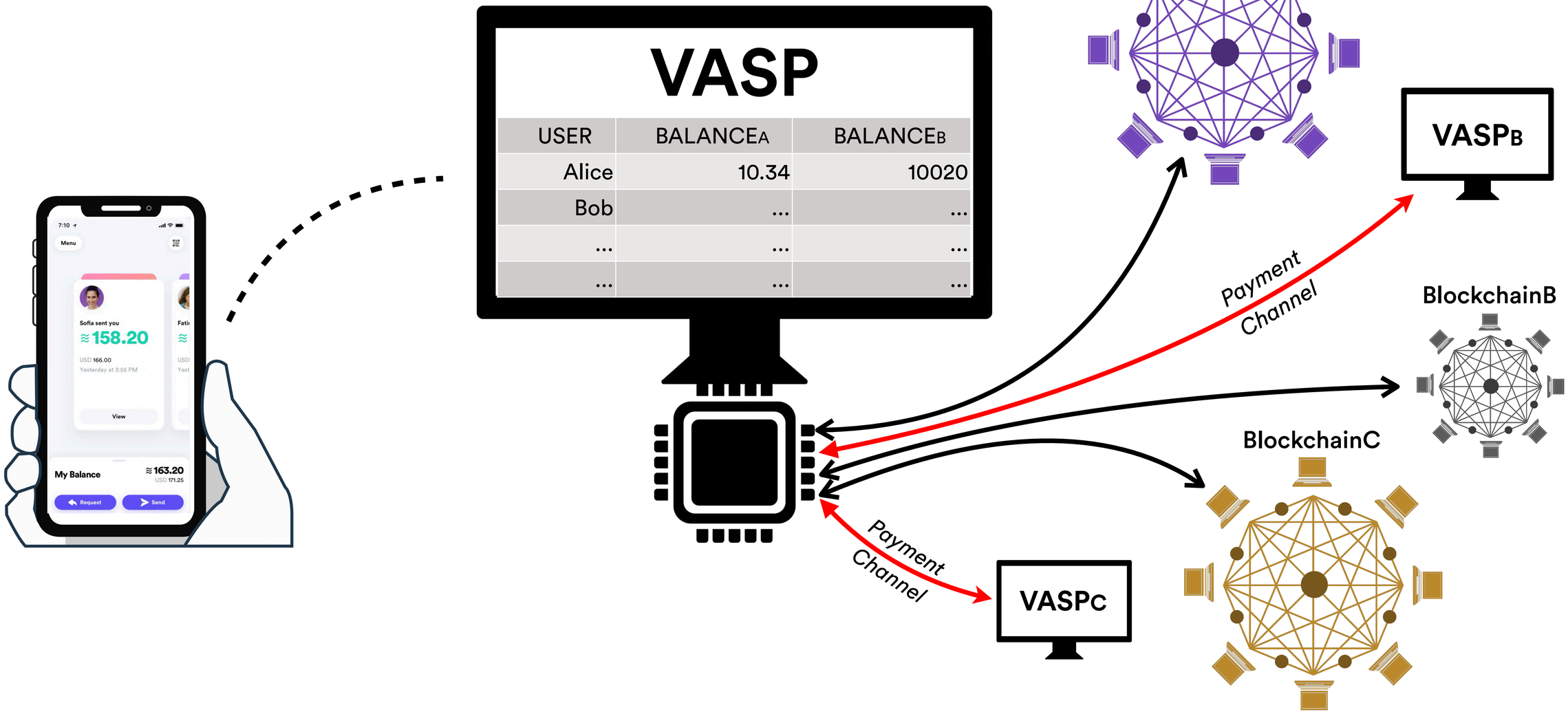






Wallet providers and exchanges

VASP = Virtual Asset Service Provider



Is your VASP solvent?

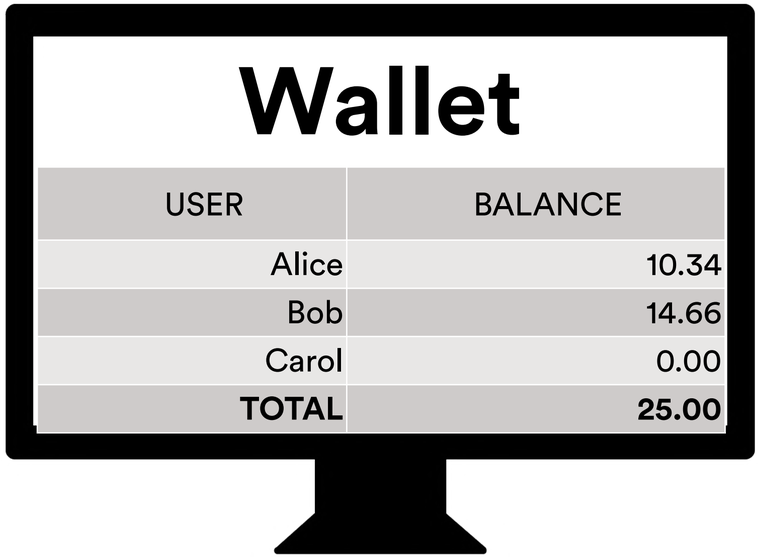
MtGox: over 850,000 Bitcoins had been stolen, including 750,000 Bitcoins owned by its customers

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.

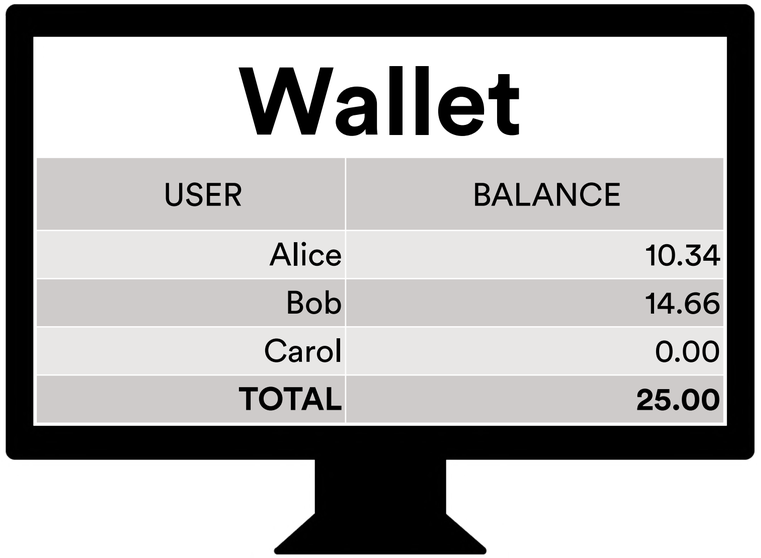
Option A [Broadcast Everything]



Publicly expose

- individual wallet balances
- wallet identities
- wallet performance
- zero balance customers
- total liabilities

Option B [Publish to Auditor(s) only]



Expose to auditors

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

Wallet - Auditor collusion?

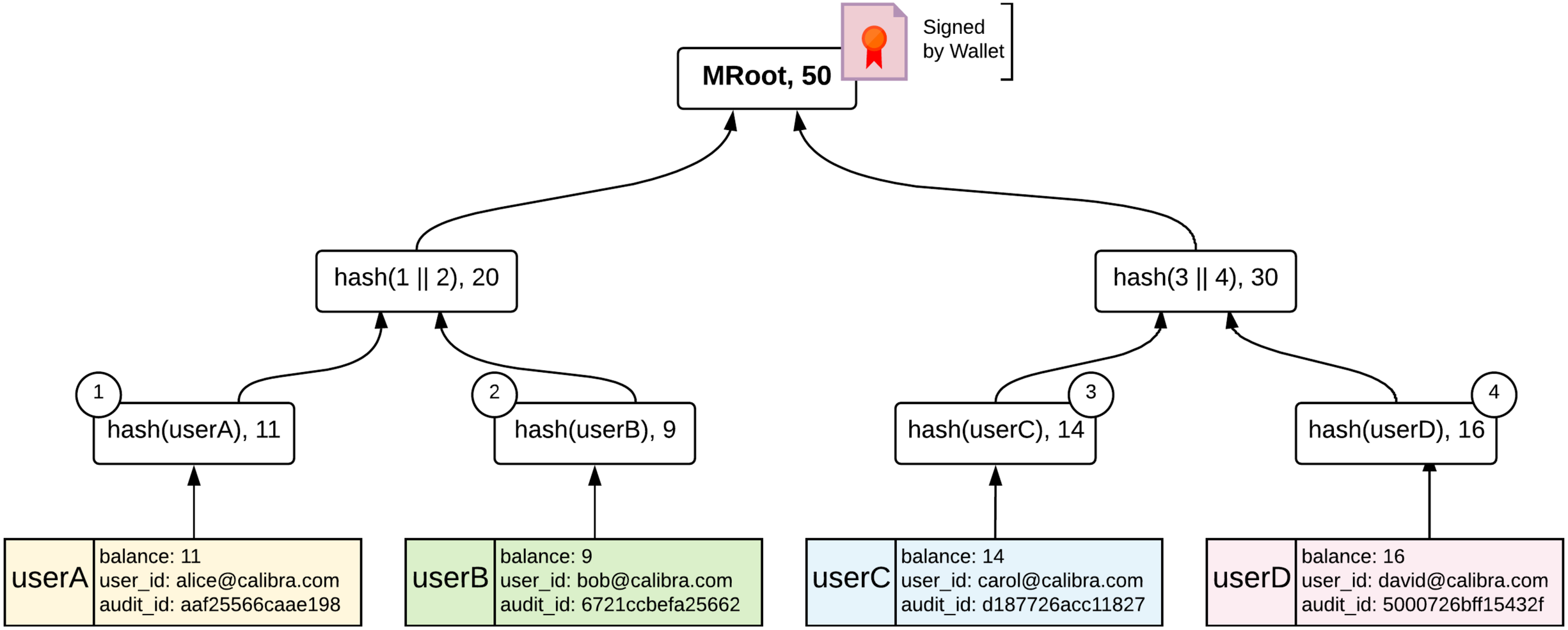
2014 - Bitstamp proves its Bitcoin reserves to Mike H.

"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

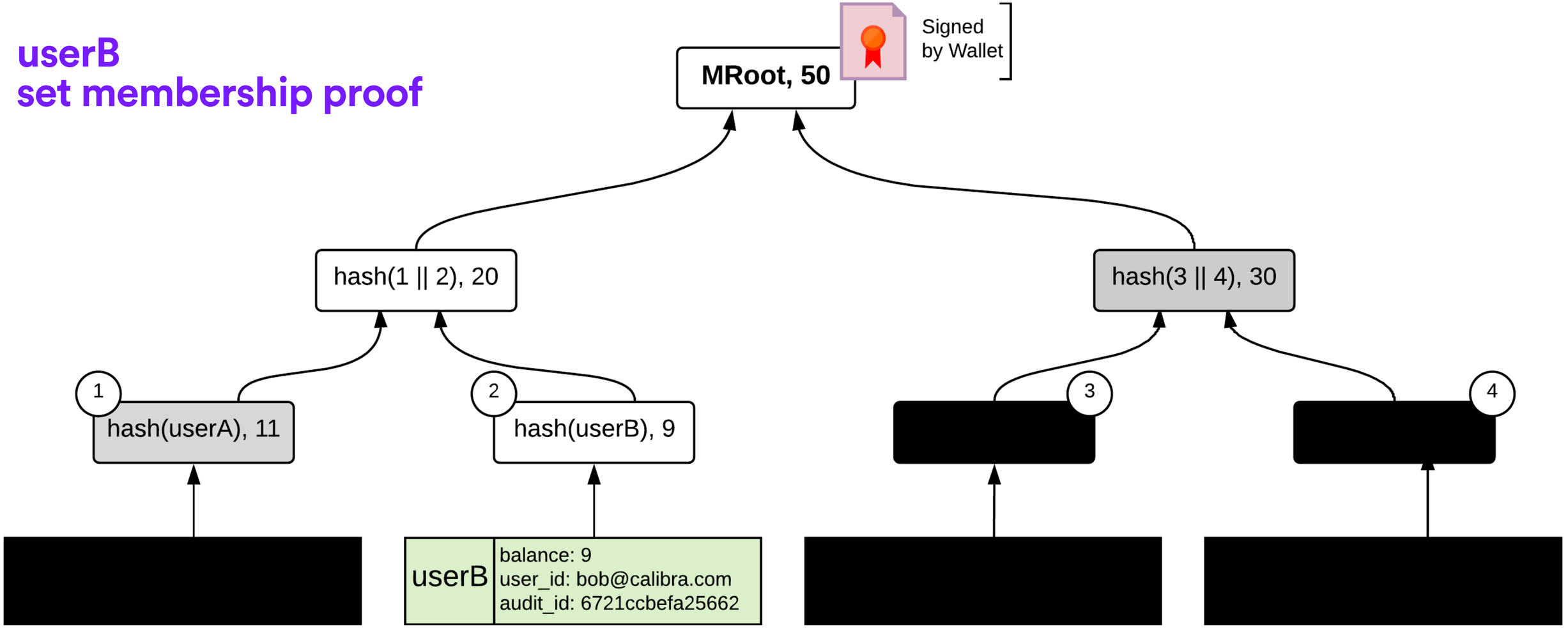
"Until we can implement fully zero-knowledge, cryptographically provable audits, you have to trust the auditor, i.e. me, to have done my job correctly"

Option C [Summation Merkle Trees]



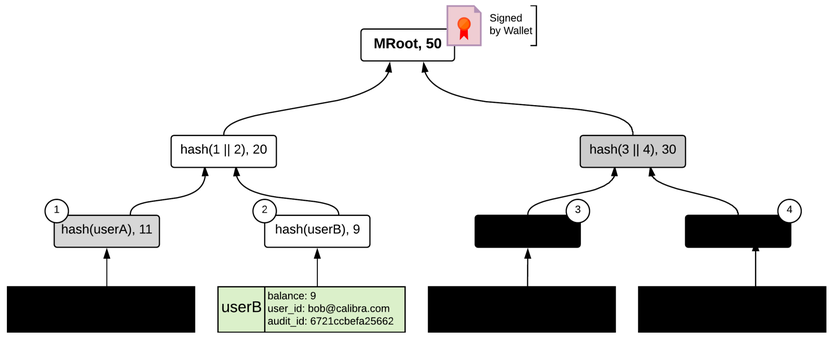
Option C [Summation Merkle Trees]

userB
set membership proof

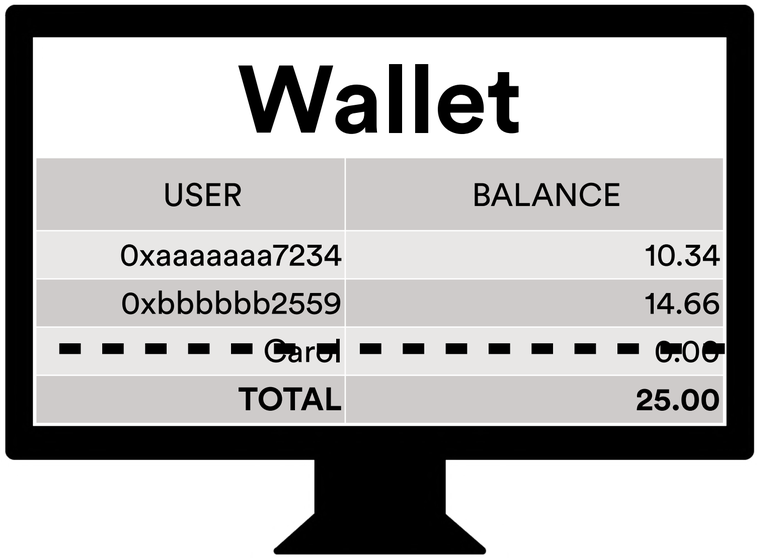


Option C [Summation Merkle Trees]

customer sees



auditor sees



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"

Expose to auditors

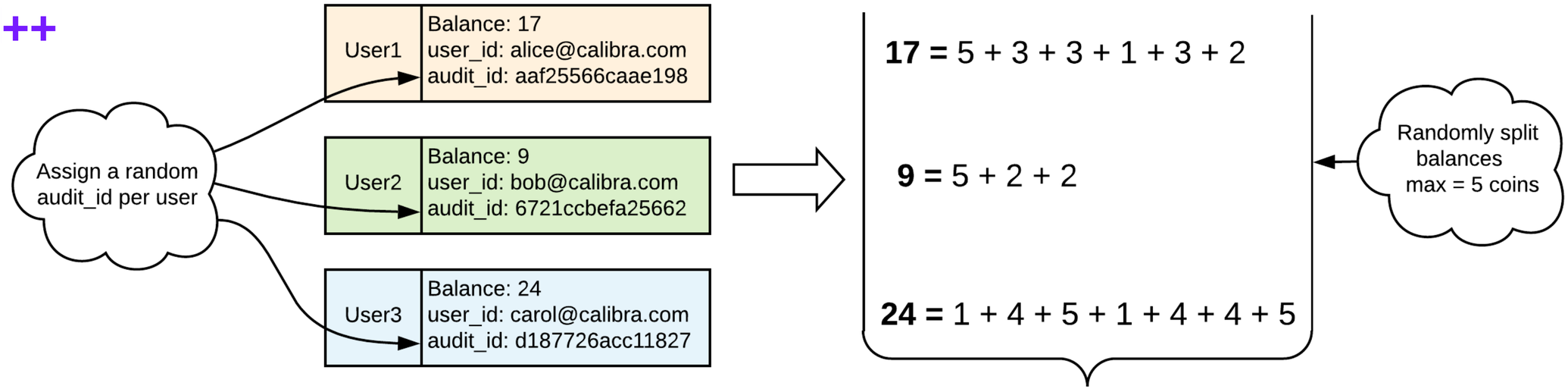
- individual wallet balances
- number of customers
- leak from multiple PoSolv
- total liabilities

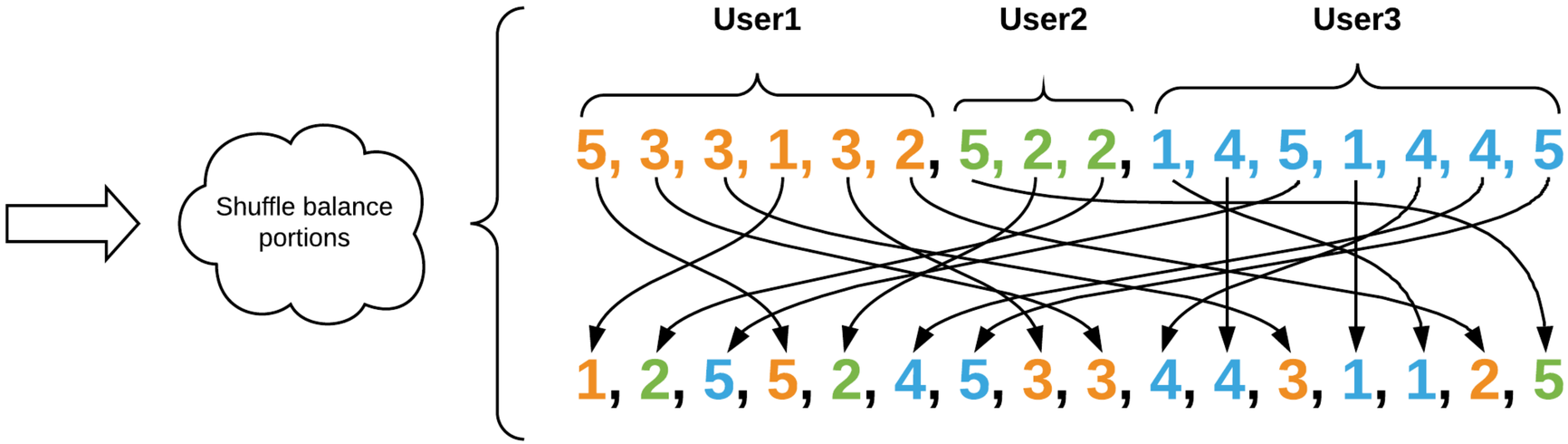
Expose to customers

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

Option D [Random Denomination Trees]

Maxwell++





Option D [Random Denomination Trees]

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

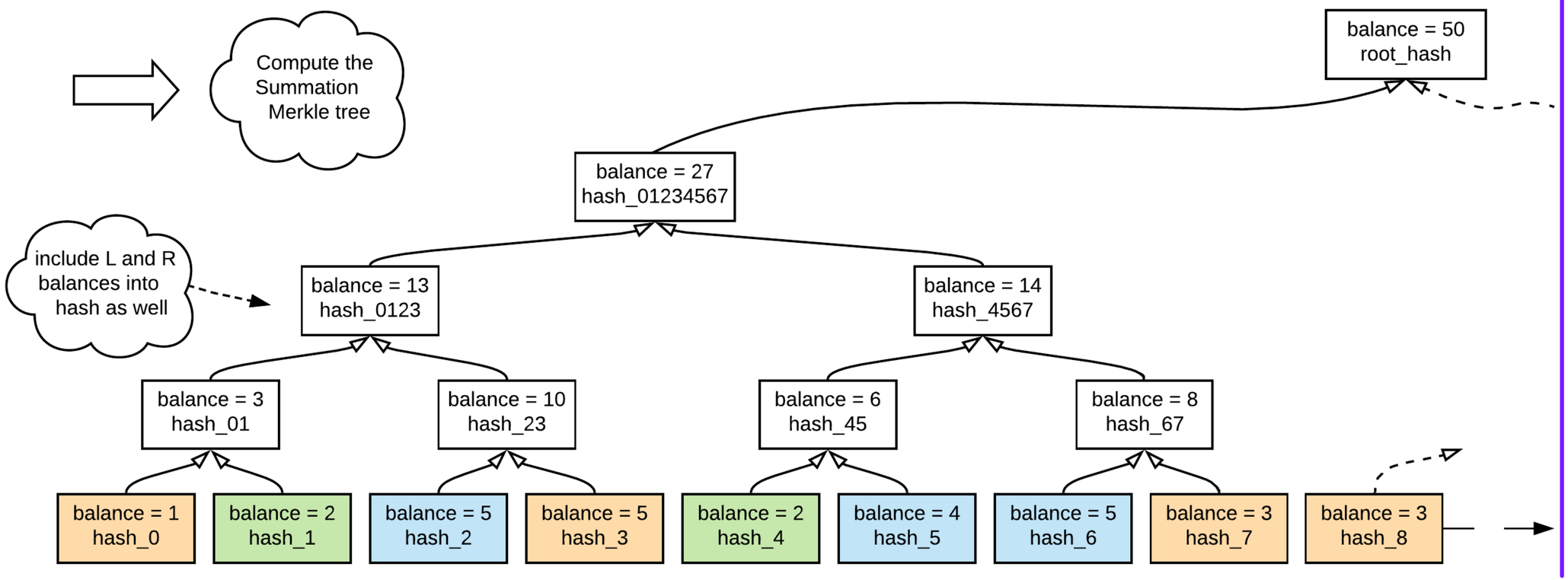
for example the first leaf will be

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

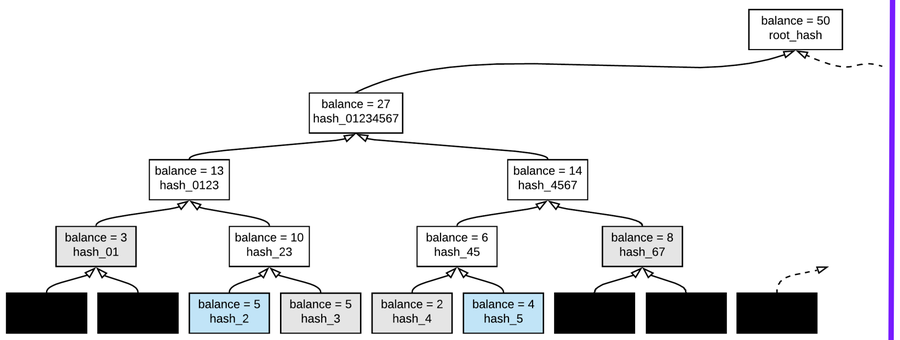


Send **balance portions**, **leaf indices**, **tree paths** and **audit_id** to users

Send **all leaves** to the auditor

Option D [Random Denomination Trees]

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
- sibling denominations
- ~~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

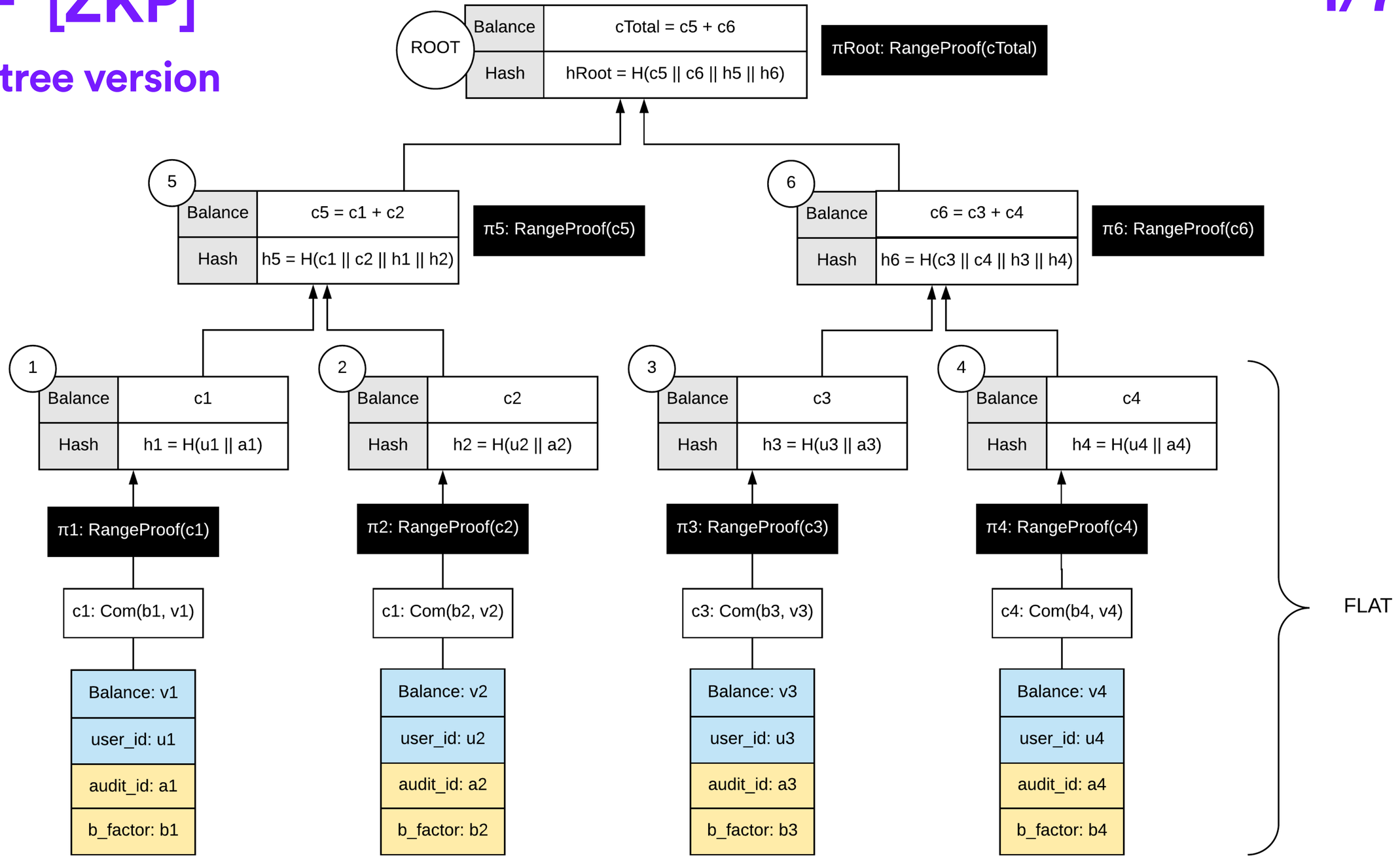
- compute Merkle roots
- check *all balances > 0* && *liabilities <= assets*
- verify key(s) ownership
- sign(Liab_MRoot, Addresses_MRoot, result)

- 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

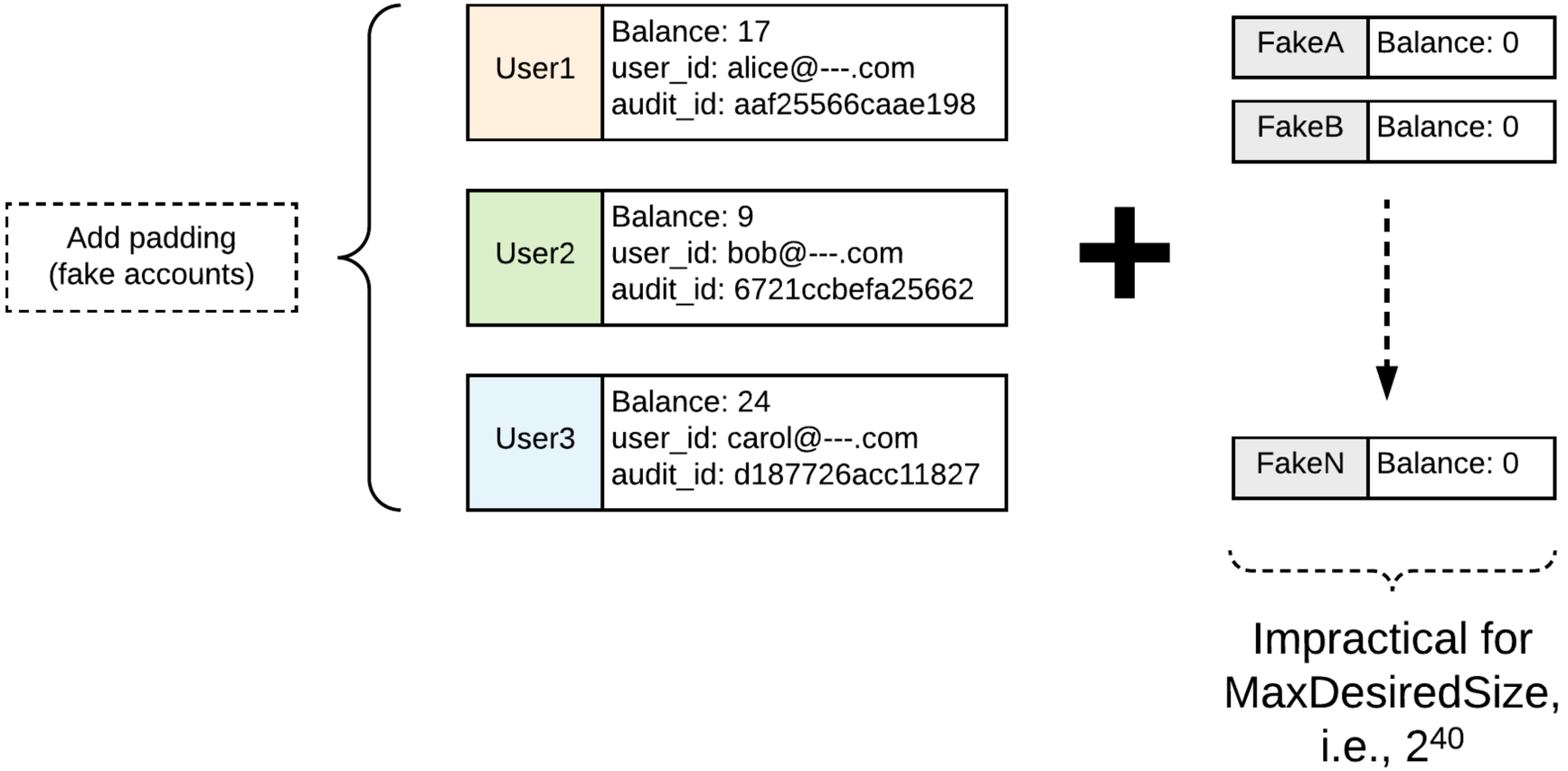
Option F [ZKP]

Provisions - tree version



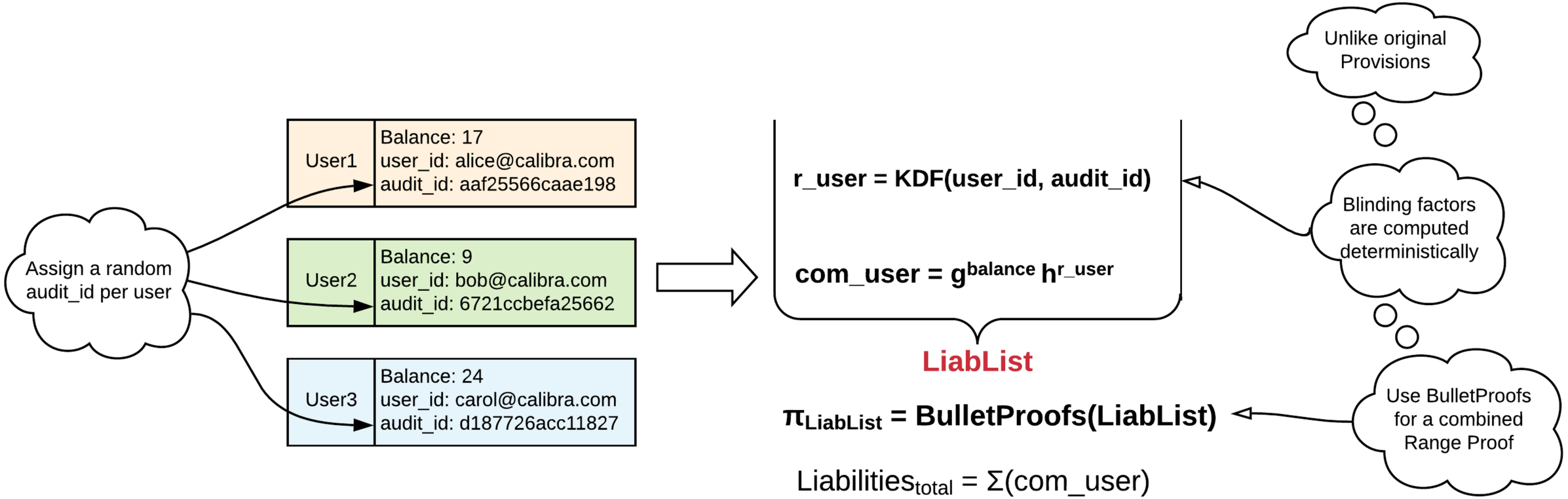
Option F [ZKP]

Provisions - padding



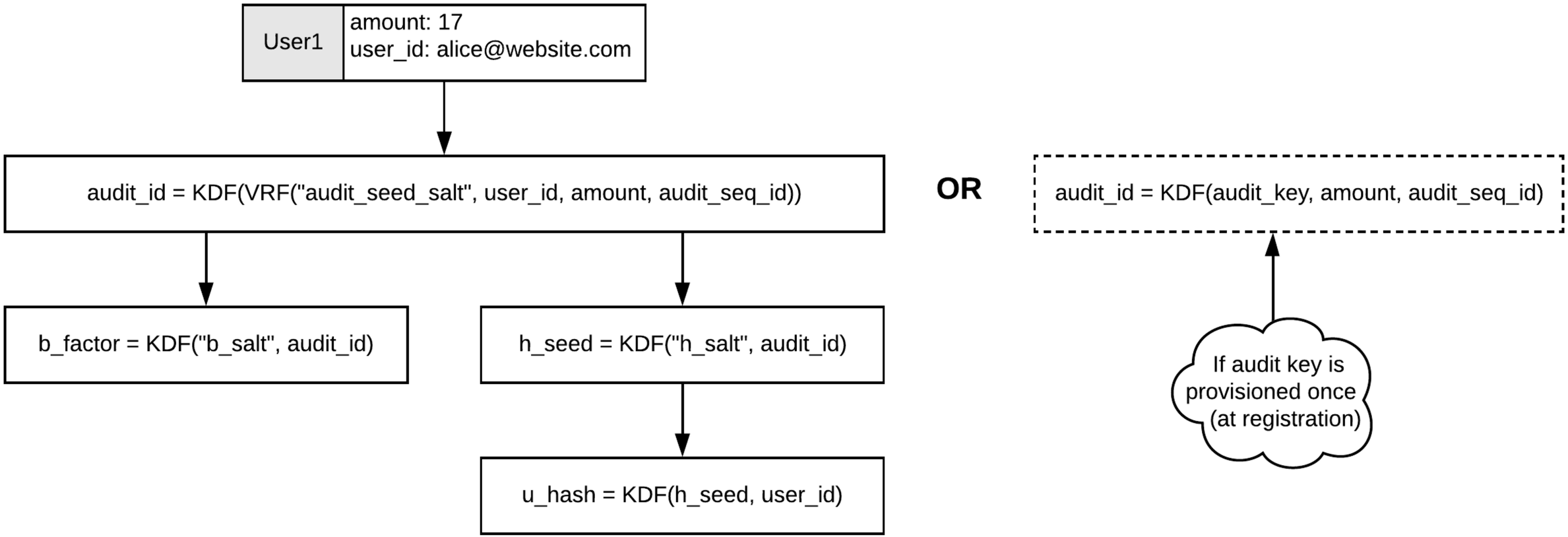
Option F [ZKP]

Optimize Provisions



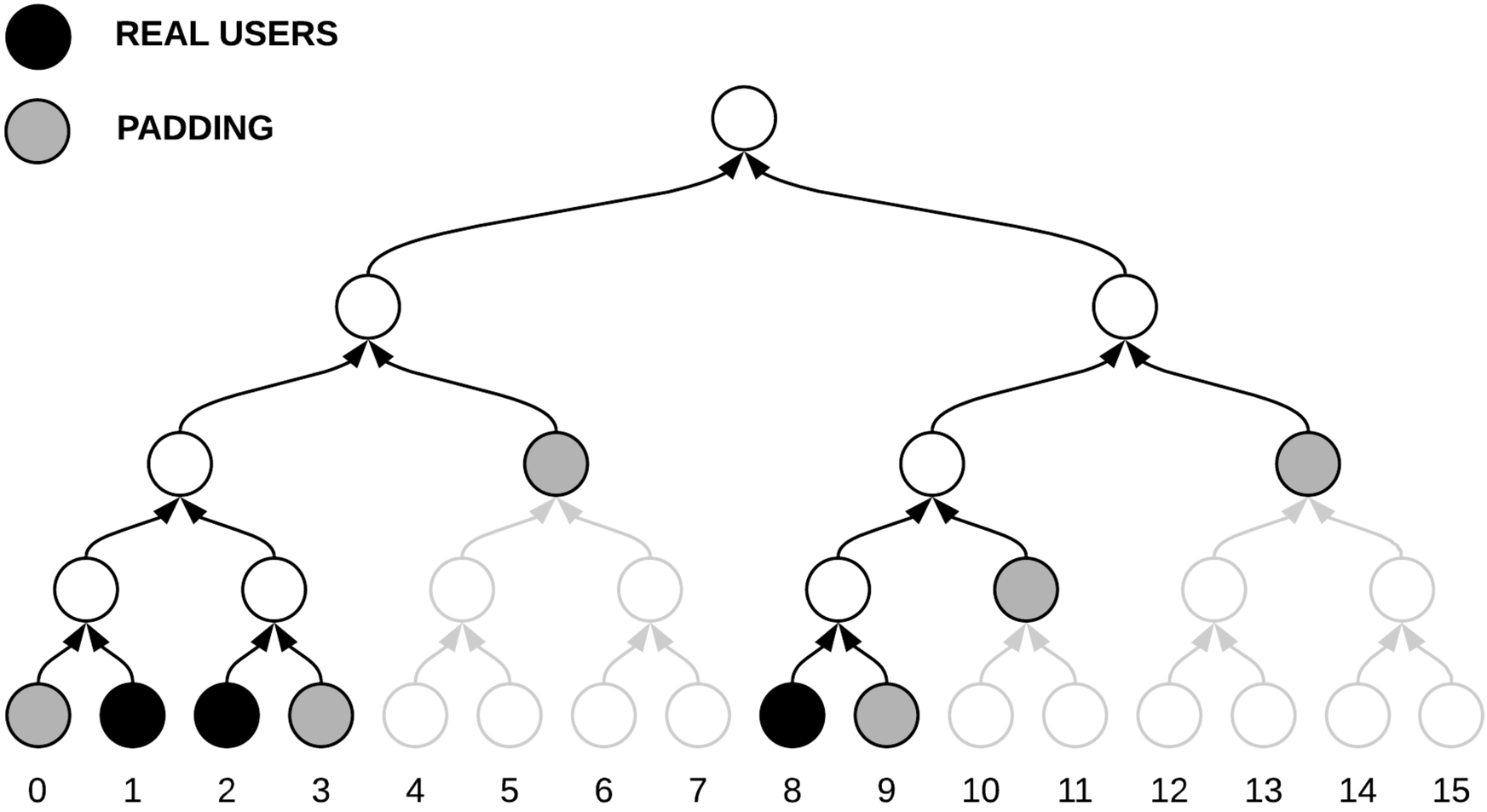
Option F [ZKP]

deterministic derivation
(fixed tree-index + decouple amount from userID)



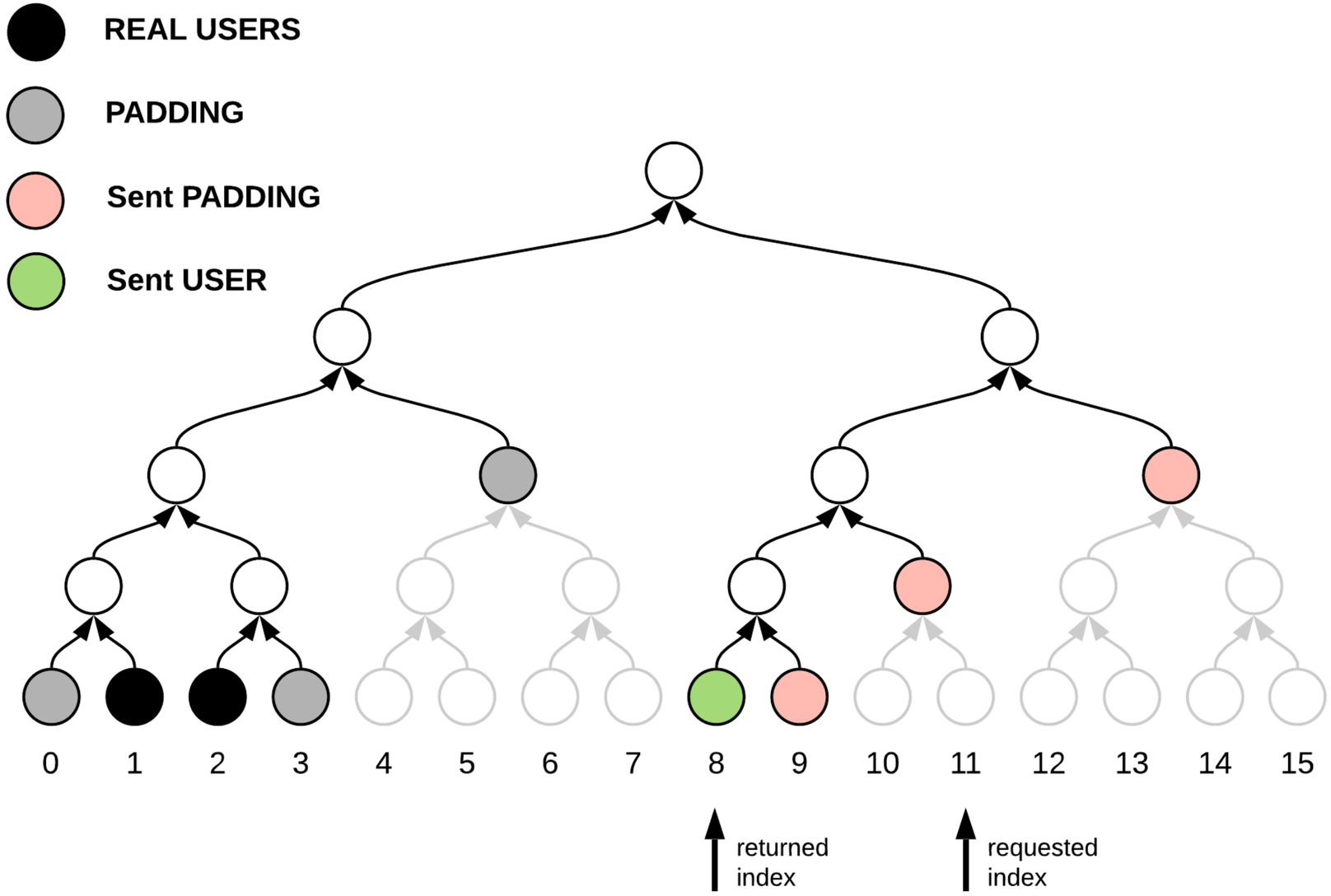
Option F [ZKP]

sparse tree (succinct padding)



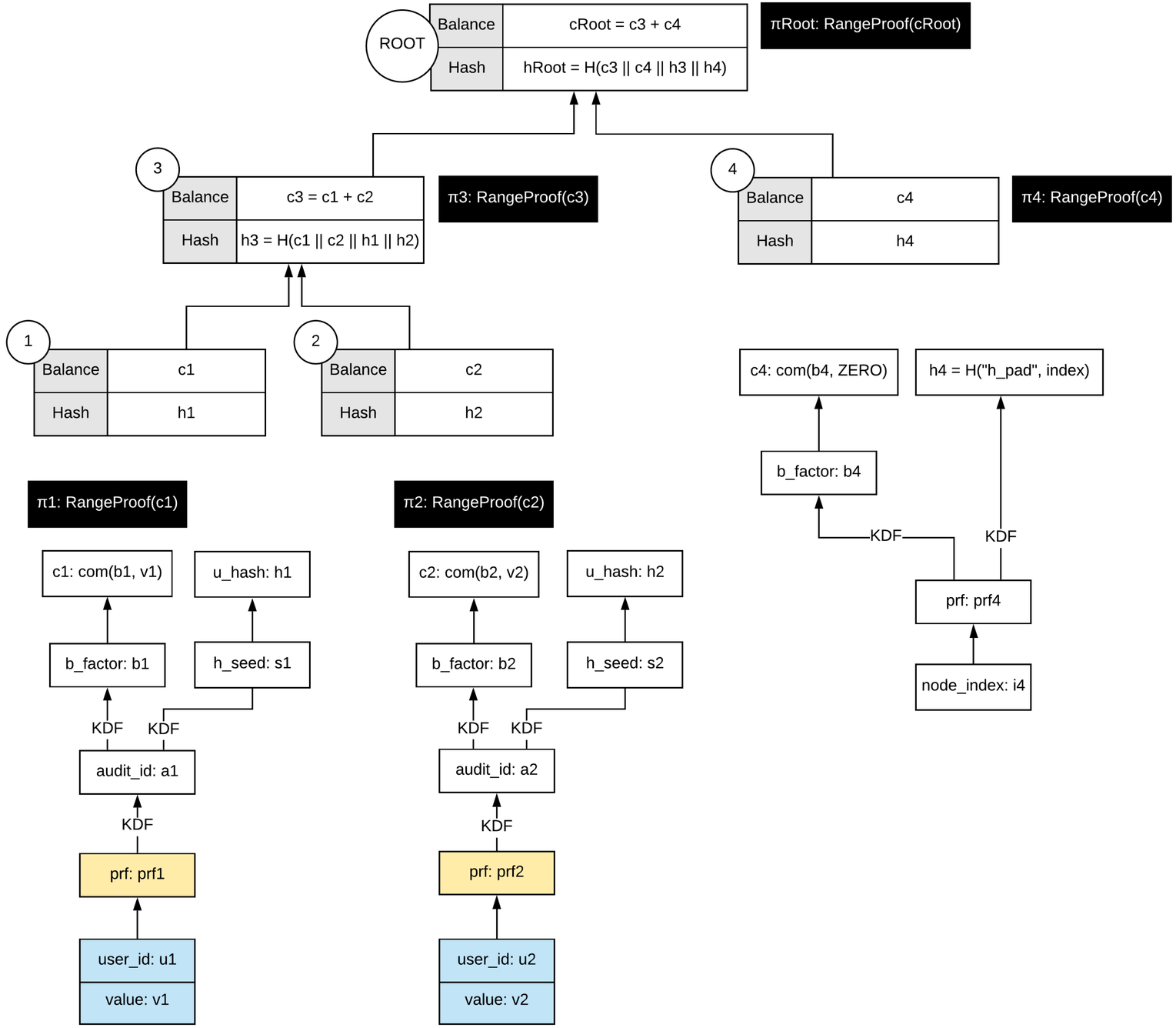
Option F [ZKP]

auditor sampling

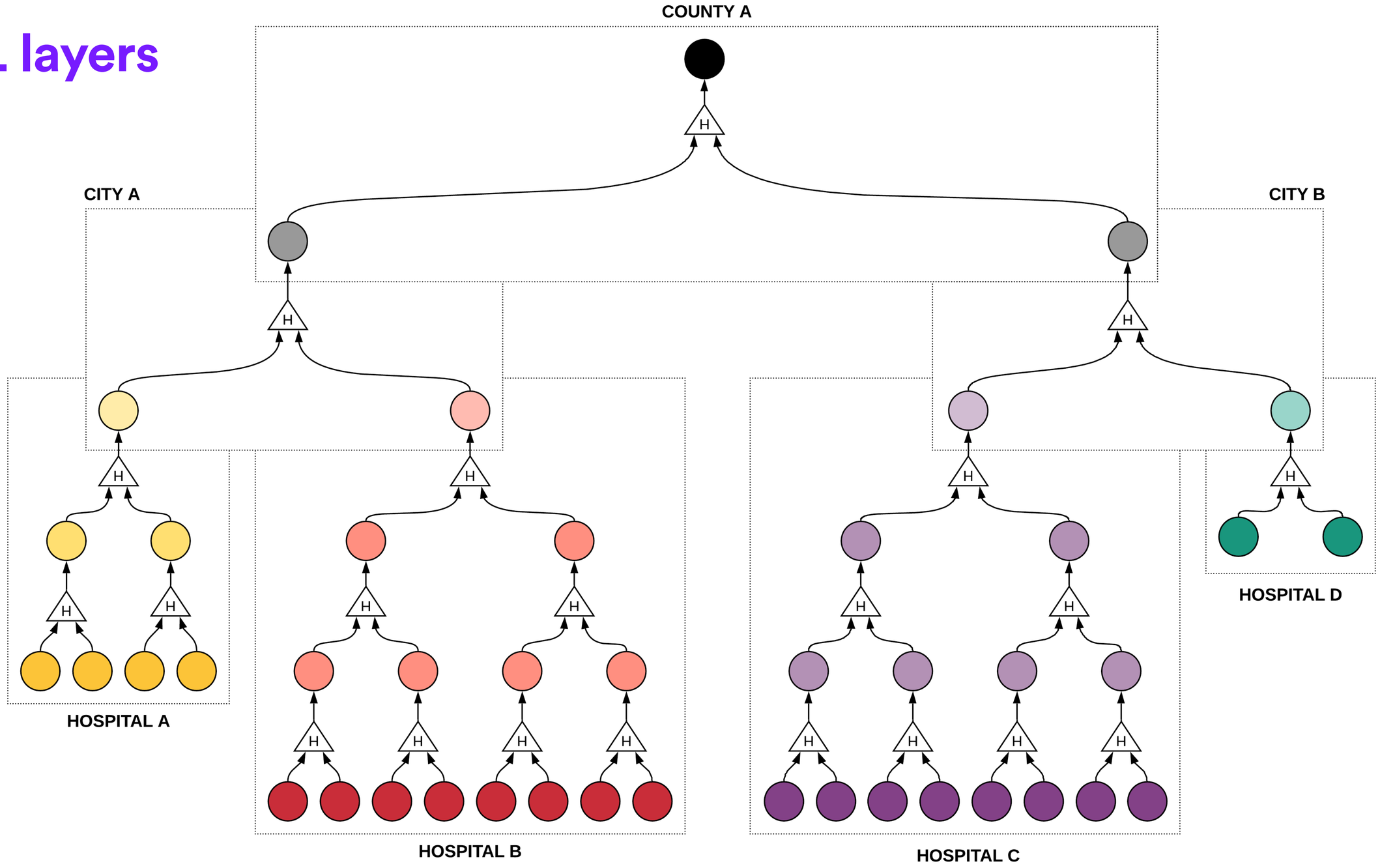


Option F [ZKP]

DAPOL



DAPOL layers



Features

A. Privacy

- Hide total value
- Hide account values and identities
- Hide size of population
- Larger ranges *(tree Vs flat)*

B. Other

- Deterministic tree
- Selective auditor sampling
  (random index, value/amount, identity)
- Hierarchical (layered)
- Reusable audit key
- New applications outside solvency

Process

Requirements & Recommendations

- Publish root to public bulletin board *(blockchain)*
- Use PIR to hide who downloaded the proof
- Users can provide their own audit_key *(devices might be equipped w/ enclaves)*
- Mutual contract - double signed (*as dispute resolution evidence)*

Applications

A. Finance

- Solvency
- Syndicated loans/insurance
- Taxable income
- Credit score
- Fundraising & ICO

B. Voting

- Counting dislikes
- Rating services/products
  (i.e., restaurants, hotels, gadgets)
- Disapproval voting
- 2-party decentralized elections

C. Reporting

- Virus outbreak daily cases
- Unemployment rate
- Work accidents

D. Misc

- Lottery jackpots
- Referral programs
- Ranking systems

COVID-19 transparent reporting

A research project which utilizes Zero Knowledge Proofs (ZKP) and the DAPOL protocol to Public Health, with the aim to provide organizations means for **transparent and private reporting**.

It's a novel algorithm that reduces dependency on centralized auditing and it enables active participation of individuals to privately authenticate public disclosure of aggregated statistics.

This technology is generic and has been accepted to ZKProofs 2020, an open-industry academic initiative that seeks to mainstream ZKP cryptography through a community-driven standardization process.

A tool to add-on to YOUR COVID-19 tracking

HOW DOES IT WORK?

REPORTING COVID-19 CASES

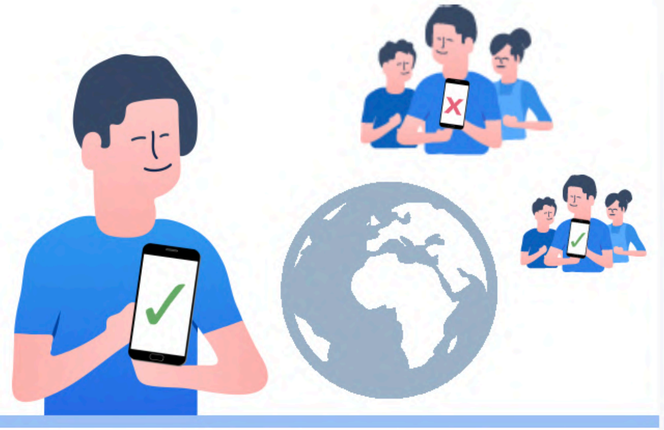
A 3-step process that allows transparent reporting of coronavirus cases.

Scan your COVID-19 test

If you are tested positive to coronavirus, local medical centers should just provide you with a signed statement. Scan it!

COVID-19 daily report

Agencies report infection numbers publicly at a daily basis. This report does not reveal patient data using privacy preserving techniques.



Everyone can verify

Those infected can privately check their inclusion in the reported aggregated number.

Thank you

DAPOL
Distributed Auditing Proofs of Liabilities

contact: Kostas Chalkias

kostascrypto@fb.com

DAPOL
Distributed Auditing Proofs of Liabilities

Acknowledgments: We would like to thank Antonio Senatore from Deloitte, Daniel Benarroch from Qedit, Brajesh Damani, Dmitry Korneev & Nihar Shah from Facebook and Christian Catalini, Evan Cheng, George Danezis, David Dill, Riyaz Faizullabhoy, J. Mark Hou, François Garillot, Ben Maurer, Dahlia Malkhi, Alistair Pott, Alberto Sonnino & Lei Wei from Calibra for their valuable and constructive feedback.

We also thank Gaby G. Dagher, Benedikt Bunz, Joseph Bonneau, Jeremy Clark and Dan Boneh, as this proposal is heavily based on their Provisions [**13**] protocol and the authors of ZeroLedge [**17**] (Jack Doerner, Abhi Shelat and David Evans), zkLedger [**29**] (Neha Narula, Willy Vasquez and Madars Virza) and Bulletproofs [**5**] (B. Bunz, Jonathan Bootle, D. Boneh, Andrew Poelstra, Pieter Wuille and Greg Maxwell), because their work had direct impact on the efficiency, feasibility and on enlisting edge-case attack vectors for the proposed solution.