ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

DEDIS Research Mashup

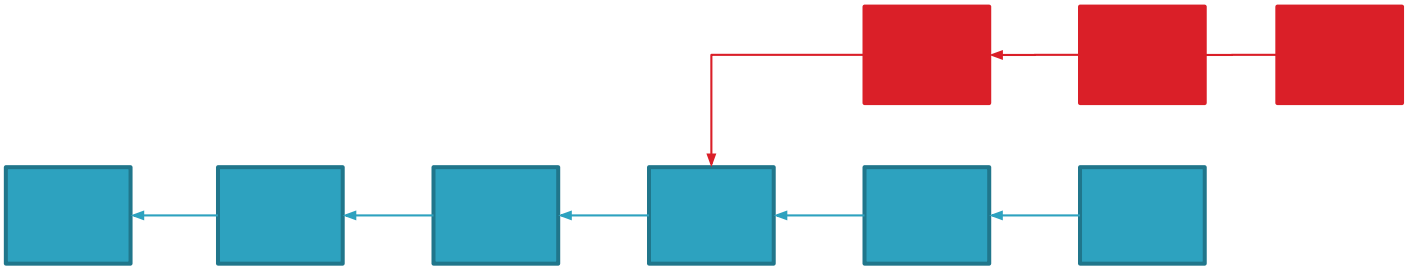
COM-402: Information Security and Privacy

(slide credits: all of DEDIS)

Overview

- ByzCoin / OmniLedger (Lefteris)
- Proof of Personhood (Linus)
- Decentralized Identity Management (Nicolas)

The Blockchain



Problem Statement

1. In Bitcoin there is no verifiable commitment of the system that a block will persist
   - Clients rely on probabilities to gain confidence.
   - Probability of successful fork-attack decreases exponentially

Bitcoin Blockchain

- What we have now:
  - Real-time verification is not safe (1 hour of delay)
  - Throughput is low (4 tx/sec)

Byzcoin Blockchain

- What can Byzcoin do:
  - Irrevocable transaction commitment in 20-90 sec
  - Throughput up to 974 TPS
  - Robust against double-spending, eclipsing, selfish mining
  - Light-weight client verification (suitable for mobile phones)

How?

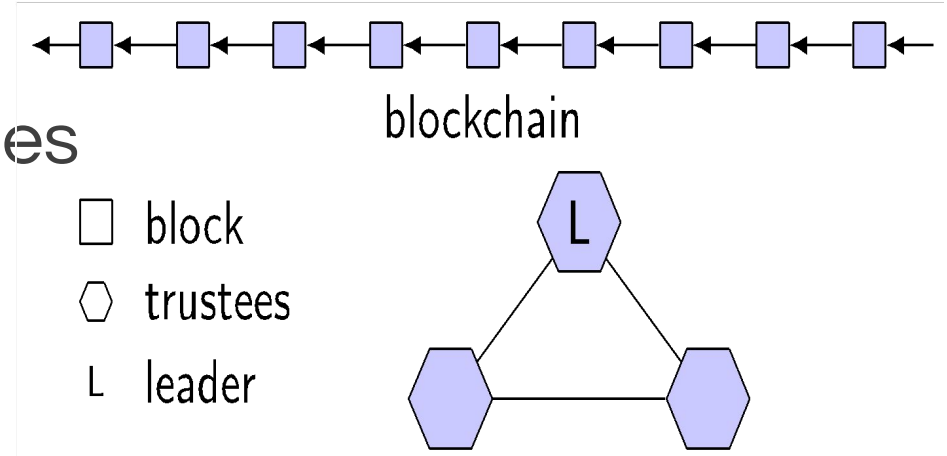
- Use Practical Byzantine Fault Tolerance protocol to provide non-probabilistic strong consistency
- Use Collective Signing to scale PBFT and decrease latency
- Use PoW to create hybrid permissionless BFT
- Use Bitcoin-NG to increase throughput

Talk Outline

- Bitcoin and its limitations
- **Strawman design: PBFTCoin**
- Opening the consensus group
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

Strawman Design: PBFTCoin

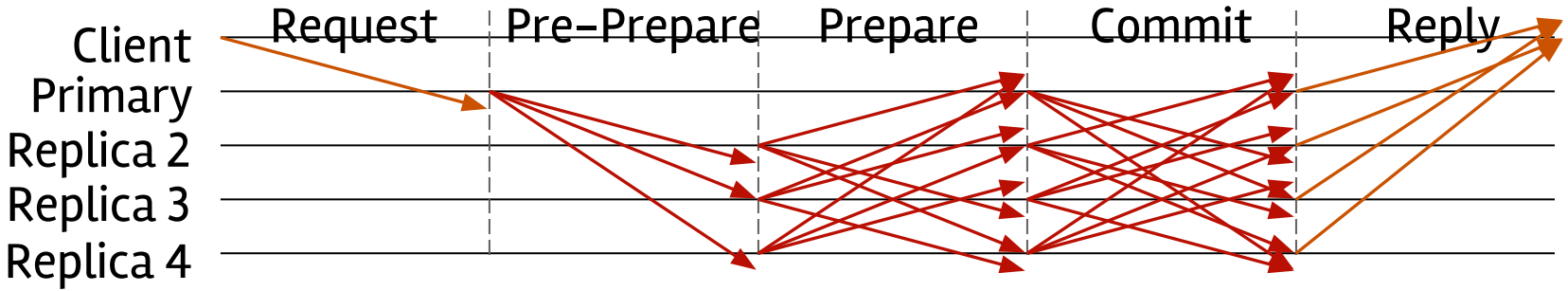
- **3f+1** fixed "trustees" running PBFT* to withstand **f** failures
- Non-probabilistic strong consistency
  - Low latency
- No forks/inconsistencies
  - No double-spending



*Practical Byzantine Fault Tolerance [Castro/Liskov]

Strawman Design: PBFTCoin

- Problem: Needs a static consensus group
- Problem: Scalability
  - Dense communication pattern (limits consensus group size)
  - High client-side verification cost (excludes mobile phones/IoT clients)
  - Absence of third-party verifiable proofs (limits number of clients)

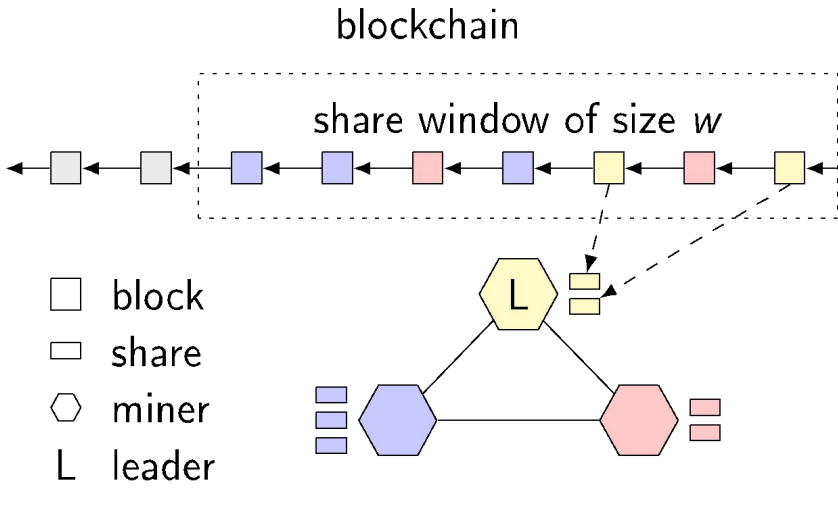


Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- **Opening the consensus group**
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

Opening the Consensus Group

- PoW against Sybil attacks
- One share per block
  - % of shares ∝ hash-power
- Window mechanism
  - Protect from inactive miners



Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- **From MACs to Collective Signing**
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

From MACs to Signing

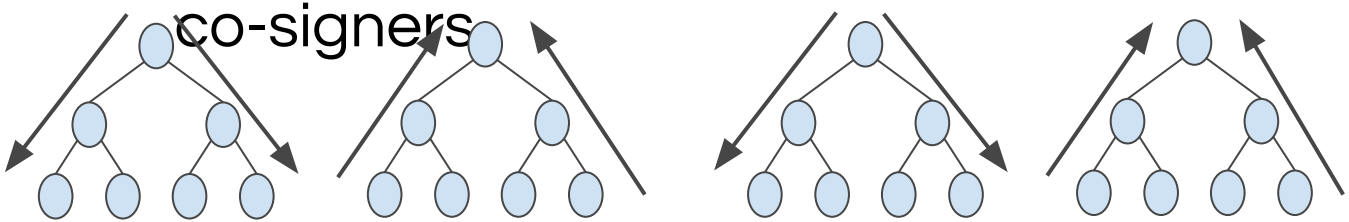
- Substitute MAC-based authentication (symmetric crypto) with public-key cryptography
  - ECDSA provides more efficiency
  - Third-party verifiable
  - PoW Blockchain as PKI
  - Enables sparser communication patterns (ring or star topologies)

From MACs to Collective Signing

- Can we get better communication patterns?
  - Multicast protocols transmit information in sub-linear steps
  - Use trees!!
- Can we allow for lightweight verification?
  - Schnorr multisignatures could be verified in constant time
  - Use signature aggregation!!
- Schnorr multisignatures + communication trees = Collective Signing [Syta et all, IEEE S&P '16]

CoSi

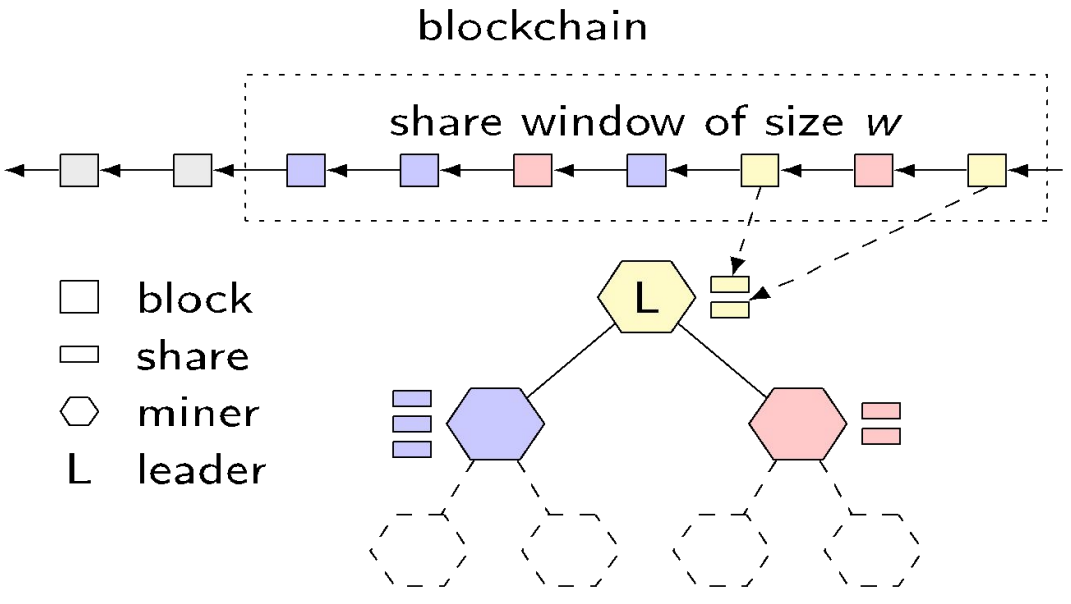
- Efficient collective signature, verifiable as a simple signature
- For the Ed25519 curve
  - 82 bytes instead of 9KB for 144* co-signers
  - 190 bytes instead of 63KB for 1008* co-signers



* Number of ~10-minute blocks in 1-day/week time window

Discussion

- CoSi is not a BFT protocol
- PBFT can be implemented over two subsequent CoSi rounds
  - Prepare round
  - Commit round



Problem Statement

1. In ~~Bitcoin~~ ByzCoin there is ~~no~~ a verifiable commitment of the system that a block will persist
2. Throughput is limited by forks
   - Increasing block size increases fork probability
   - Liveness exacerbation

Talk Outline

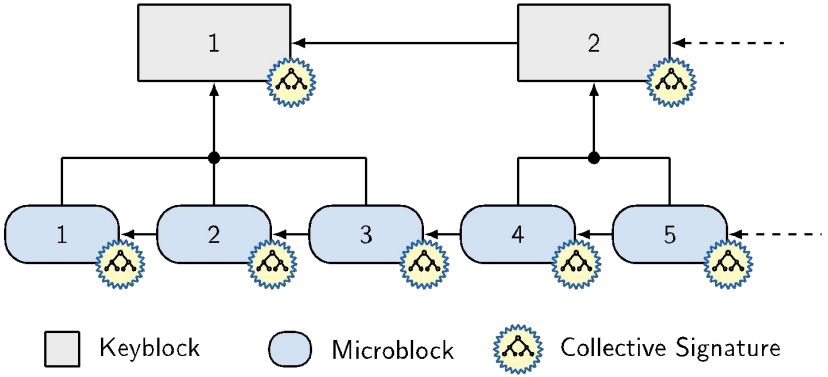
- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- From MACs to Collective Signing
- **Decoupling transaction verification from leader election**
- Performance Evaluation
- Future work and conclusions

Bitcoin-NG [Eyal et all, NSDI '16]

- Makes the observation that block mining implement two distinct functionalities
  - Transaction verification
  - Leader election
- We enhance Bitcoin-NG with Byzantine consensus
  - No double-spending
  - Non-probabilistic security
  - Leader cannot misbehave

Decoupling Transaction Verification from Leader Election

- Key blocks:
  - PoW & share value
  - Leader election
- Microblocks:
  - Validating client transactions
  - Issued by the leader



Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- **Performance Evaluation**
- Future work and conclusions

Performance Evaluation

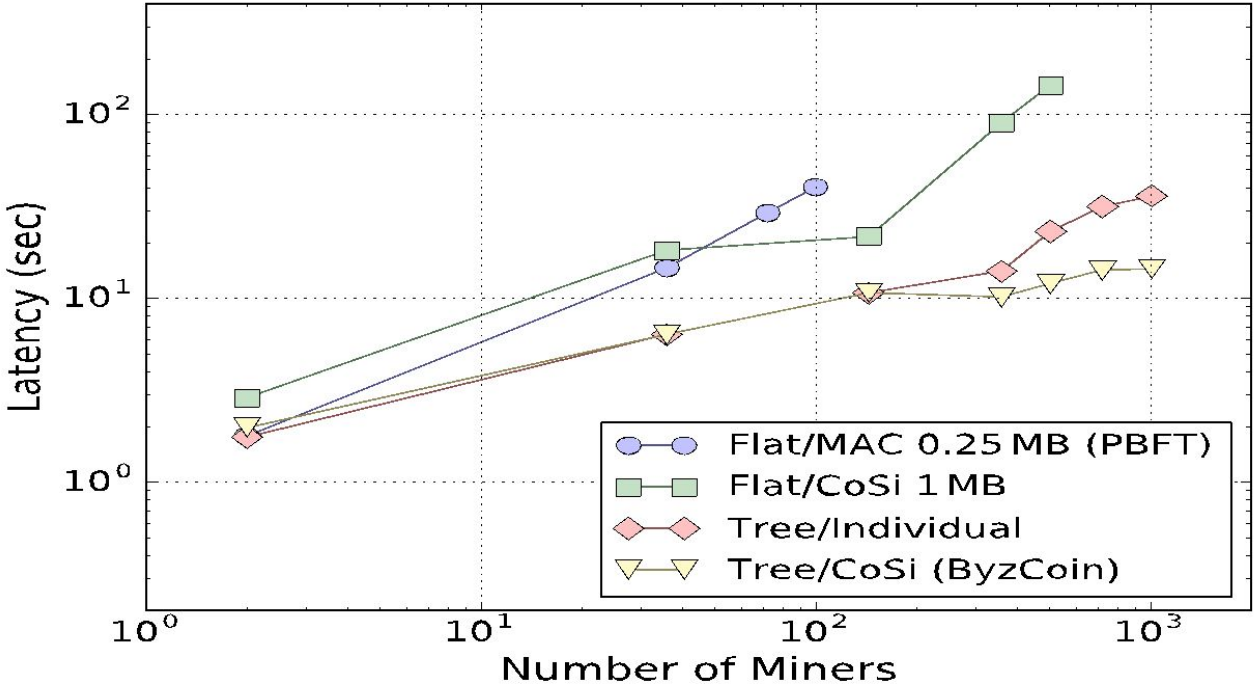
- Experiments run on DeterLab network testbed
  - Up to 1,008* miners multiplexed atop 36 machines
  - Impose 200 ms latencies between all servers
  - Impose 35 Mbps bandwidth per miner

* 1008 = # of ~10-minute key-blocks in 1-week time window

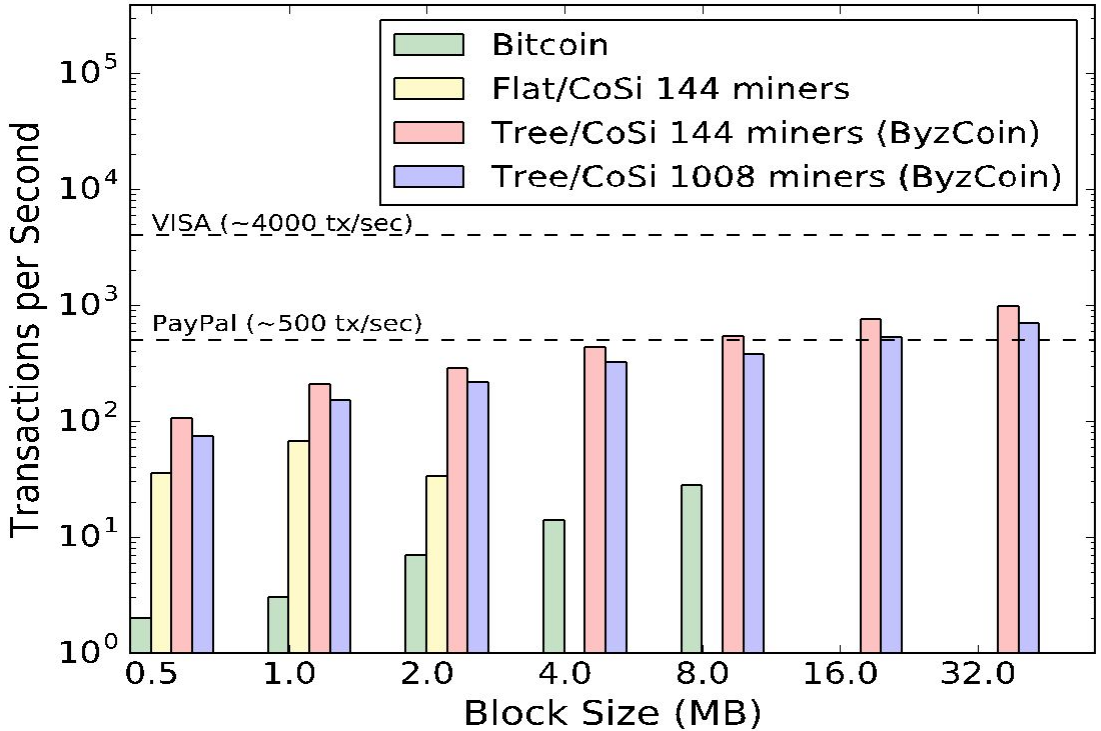
Performance Evaluation

- Key questions to evaluate:
  - What size consensus groups can ByzCoin scale to?
  - What transaction throughput can it handle?

Consensus Latency



Throughput


Bitcoin
Flat/CoSi 144 miners
Tree/CoSi 144 miners (ByzCoin)
Tree/CoSi 1008 miners (ByzCoin)
VISA (~4000 tx/sec)
PayPal (~500 tx/sec)
Transactions per Second
10^0
10^1
10^2
10^3
10^4
10^5
0.5
1.0
2.0
4.0
8.0
16.0
32.0
Block Size (MB)

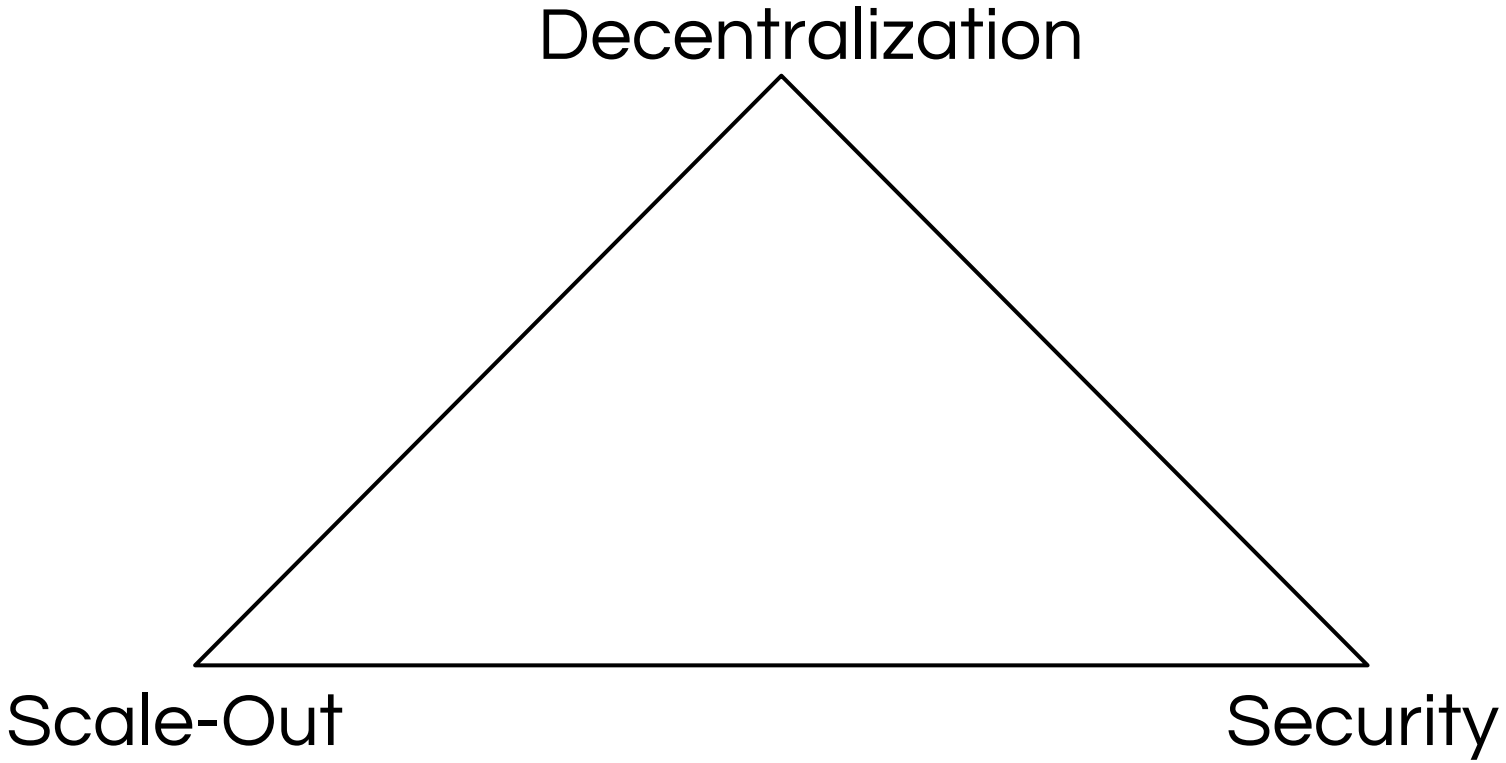
OMNILEDGER: A SECURE, SCALE-OUT, DECENTRALIZED LEDGER

Lefteris Kokoris-Kogias, Philipp Jovanovic,
Linus Gasser , Nicolas Gailly, and Bryan Ford
EPFL
EPFL

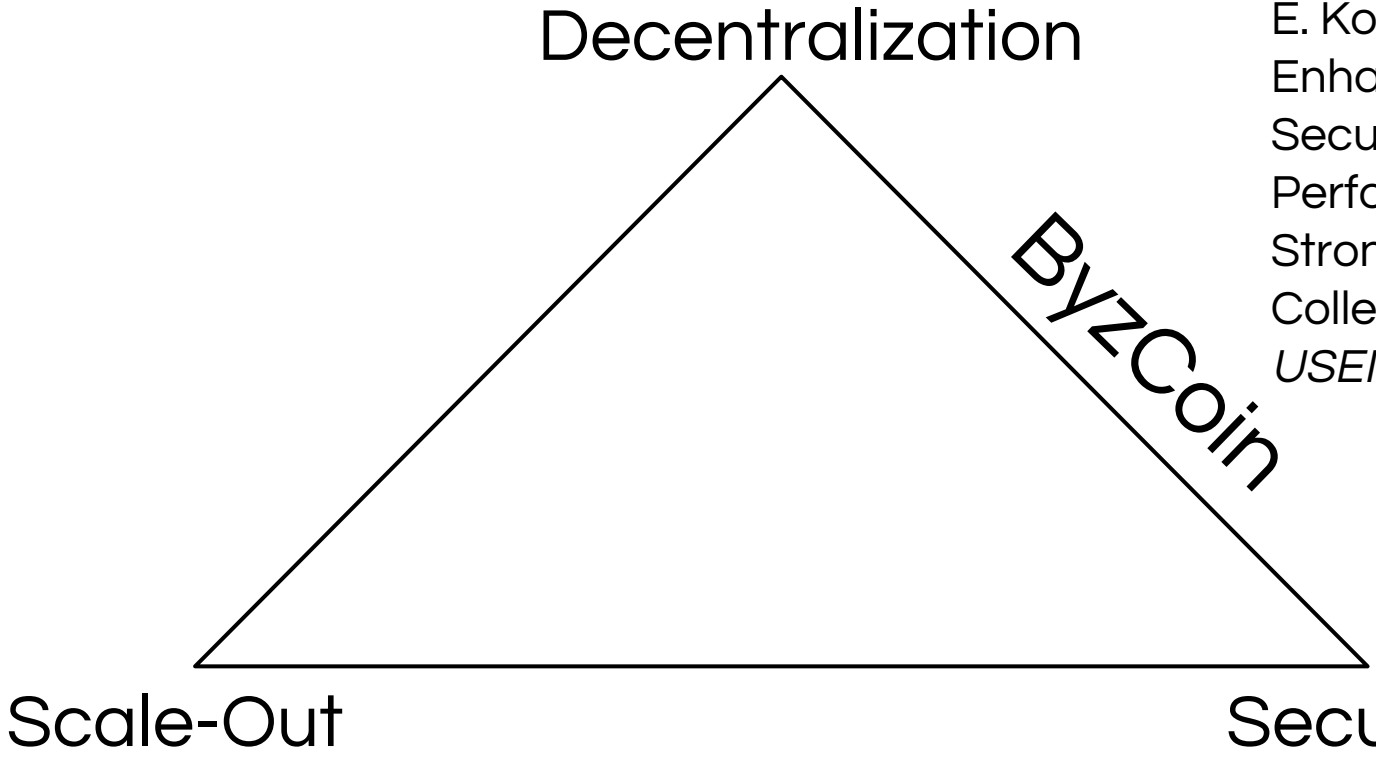
@LefKo
k

Swiss Federal Institute of Technology
Lausanne

Trade-offs in Current Proposals



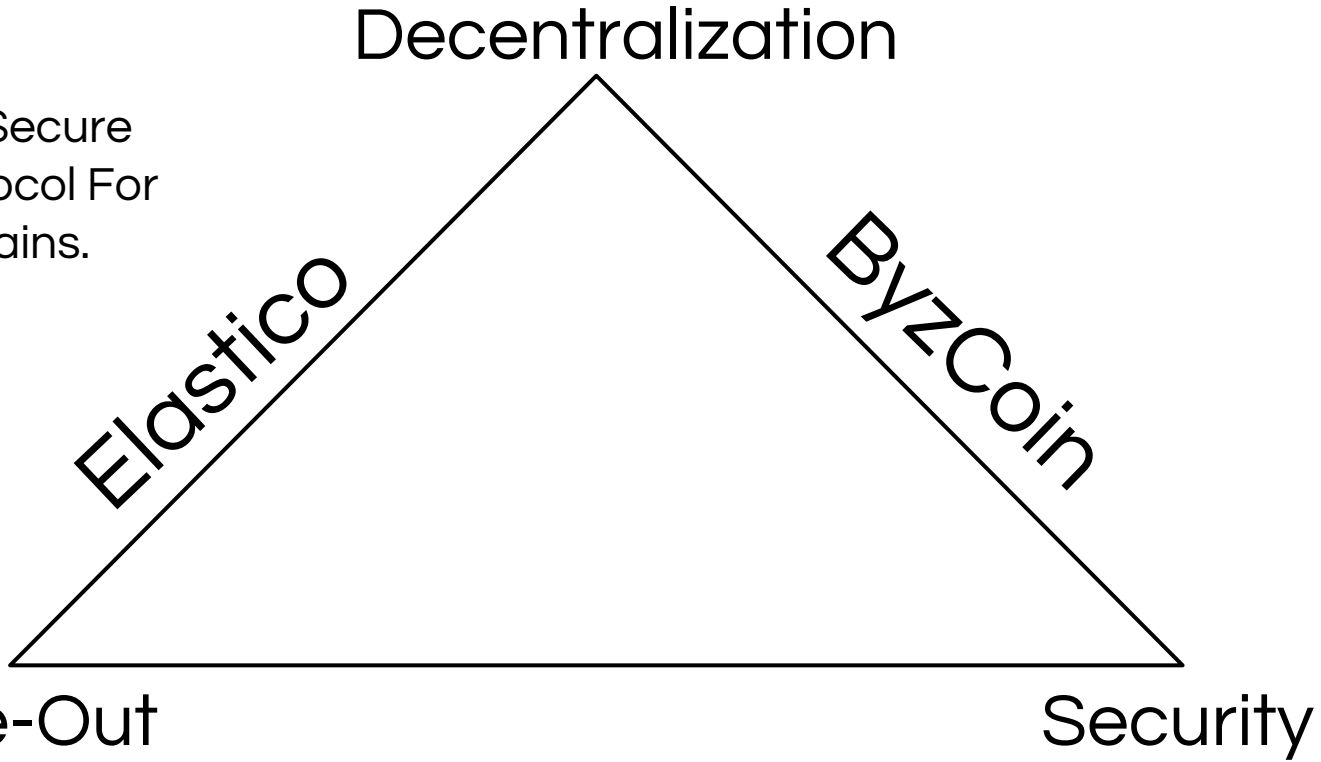
Trade-offs in Current Proposals



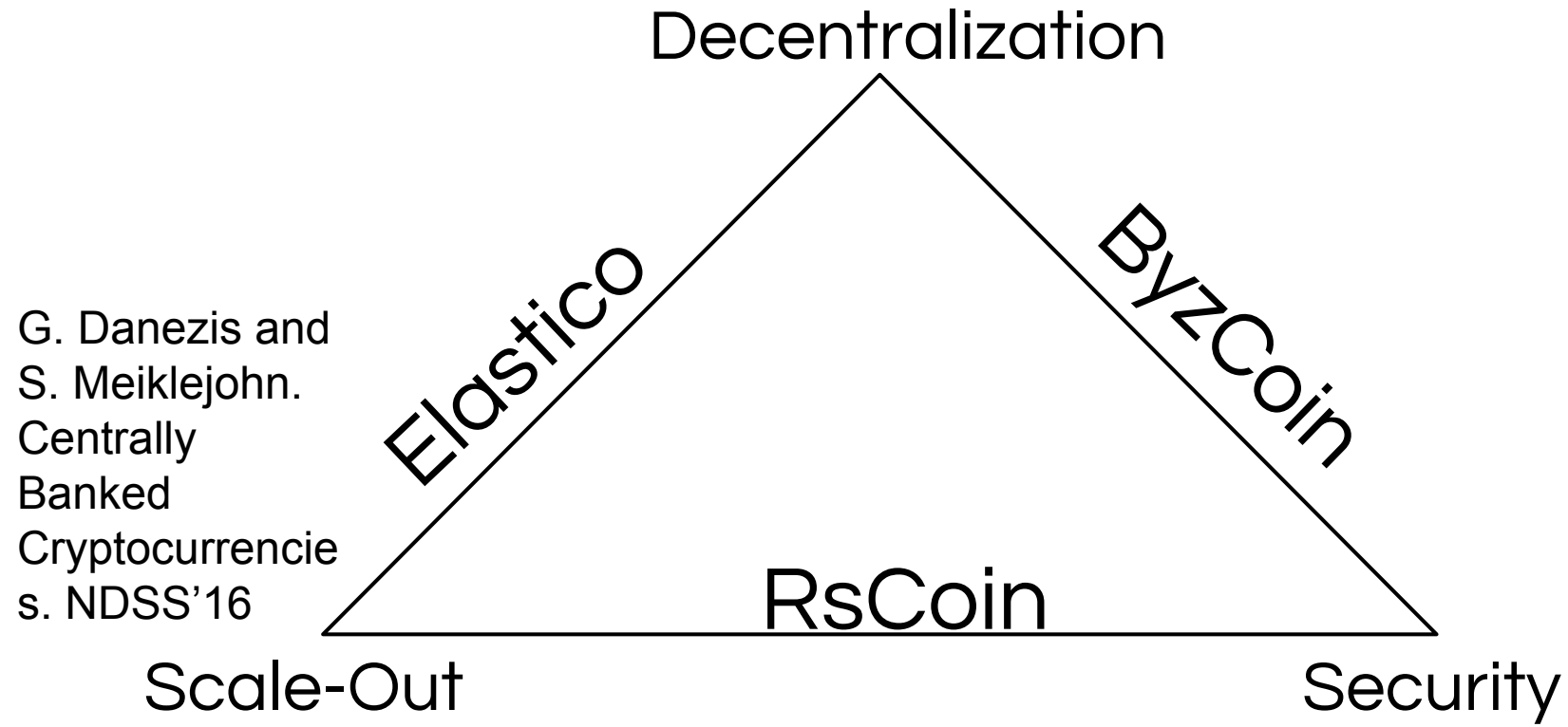
E. Kokoris-Kogias et al. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. *USENIX Sec'16*

Trade-offs in Current Proposals

L. Luu et al. A Secure Sharding Protocol For Open Blockchains. CCS '16.



Trade-offs in Current Proposals



G. Danezis and S. Meiklejohn. Centrally Banked Cryptocurrencies. NDSS'16

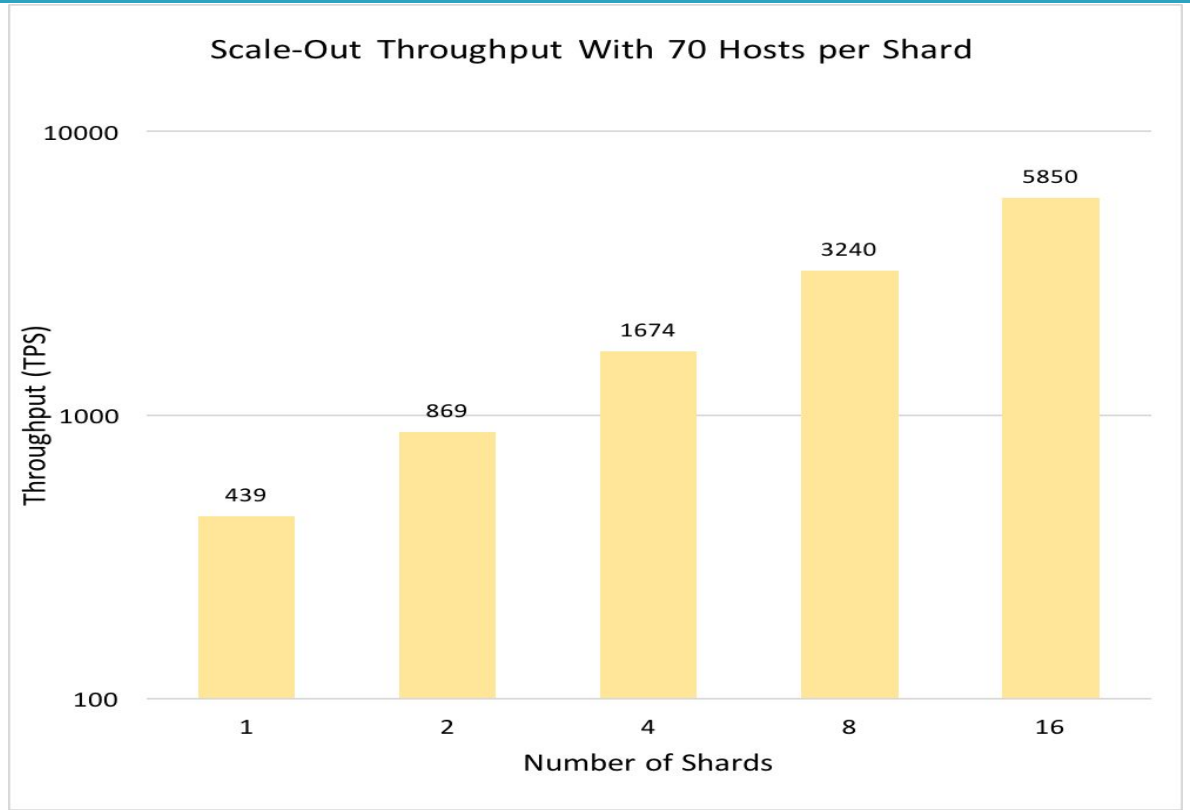
OmniLedger

- Secure - breaks once in three hundred years
- Scale-Out - Performance comparable to VISA
- Decentralized – Sybil resistance agnostic (PoW, PoS, PoP, Permissioned)

OmniLedger - Contributions

- Parallel Consensus for faster PBFT than ByzCoin
- Extend RandHound for decentralized randomness creation without the need for an accountable client
- Secure Atomic Commit on top of BFT shards
- State Blocks for log truncation

OmniLedger – Performance



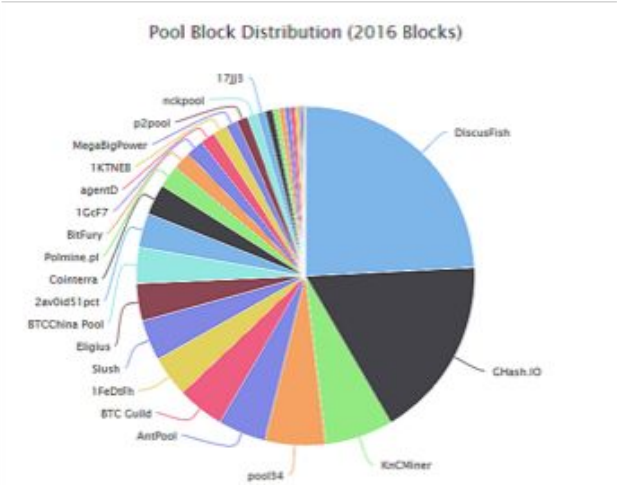
Lots more

- Scalable bias-resistant randomness
- Decentralized software updates
- Censorship resistant decentralized web
- Decentralized identity-based access control

Who Participates in Consensus?

Permissionless blockchains (Bitcoin, Ethereum): "anyone" who invests in solving crypto-puzzles.

- Now practical only with ASICs and cheap power
- Re-centralization: e.g., 4 pools now hold >50%




Environmental Costs
Proof-of-work = “scorched-earth” blockchains
Tremendous energy waste,
now comparable to all of Ireland

We Need Zero-Knowledge "Proofs of Real-Personhood"

But must have a foundation in the real world!

- IP addrs, Proof-of-[Work,Storage,Stake,etc] are just different measures of legacy wealth/power
- We *could* build on government-issued IDs
  - But who wants to trust governments to get it right?
- We *could* build on social media, federated ID
  - Anonymized via, e.g., Crypto-Book [CODASPY '16]
  - Weak, but Sybil attack cost measurable and >0

Could a "personhood-proof" foundation depend on little or no government, commercial infrastructure?

Open *Democratic* Blockchains?

Proof-of-Personhood: "one person one vote"

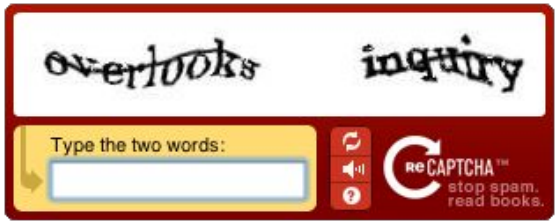
- e.g., via Pseudonym Parties [SocialNets '08]
- Participants mint new currency at equal rate
  - Decentralized analog to "basic income"?

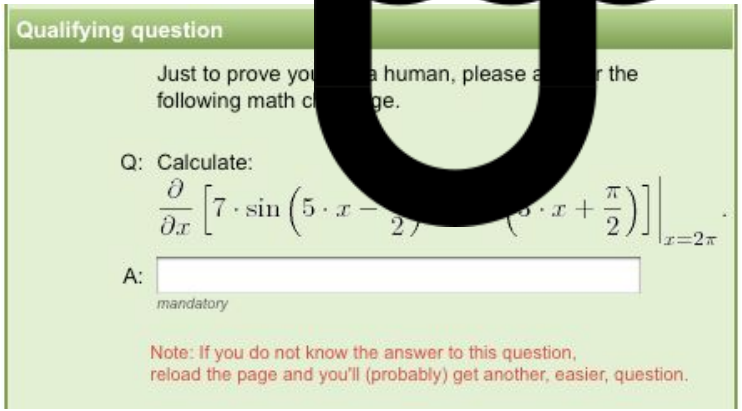
Proof of Personhood

- **Anonymity vs accountability**
- Proof of personhood (PoP)
- Pseudonym party
- Usage of PoP-tokens
- Possible applications

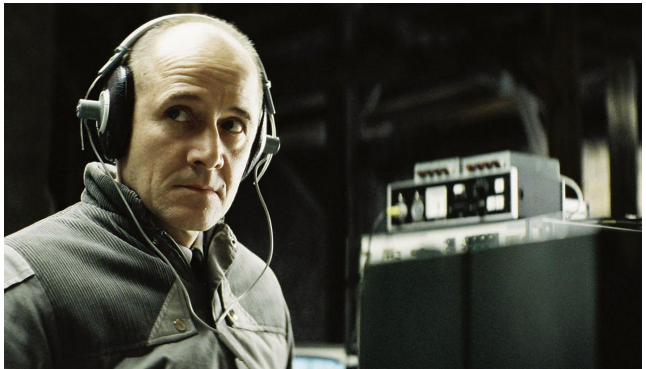
Anonymity vs accountability

Fake accounts





Real accounts



Accountability <-> Anonymity

WIKIPEDIA
The Free Encyclopedia

Accountability

Anonymity

Accountability - how?

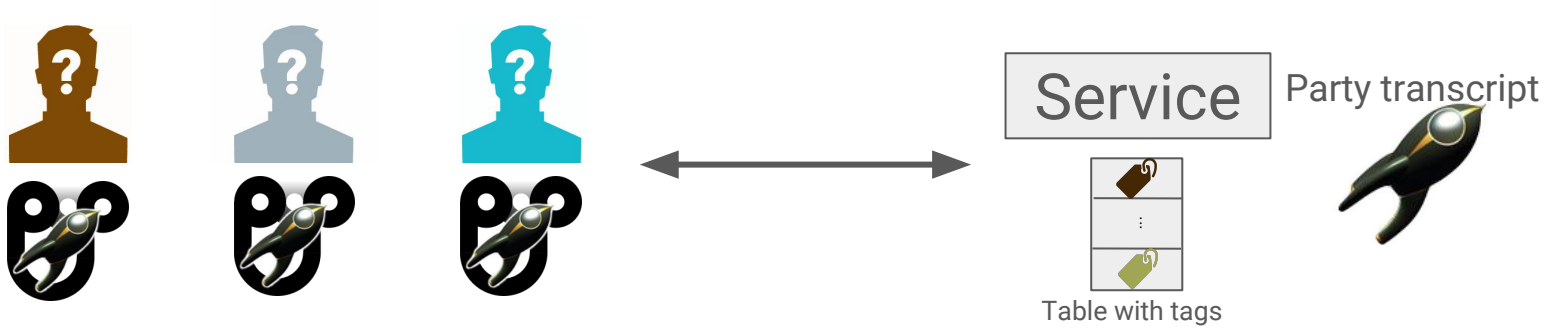
Distributed

Proof of Personhood

- Anonymity vs accountability
- **Proof of personhood (PoP)**
- Pseudonym party
- Usage of PoP-tokens
- Possible applications

Proof of Personhood

Objective:



How: Organizing a party in which people are verified, but not identified

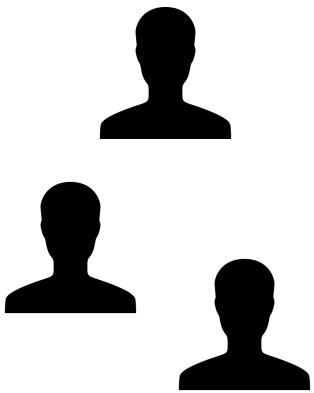
Proof of Personhood

- Anonymity vs accountability
- Proof of personhood (PoP)
- **Pseudonym party**
- Usage of PoP-token
- Possible applications

Experimental

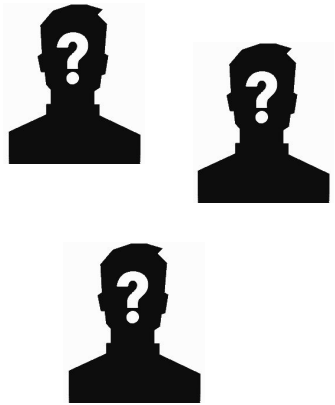
Pseudonym-party - Setup

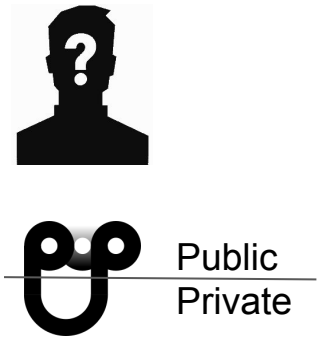
Organizers



Party-
Configuration

Attendees

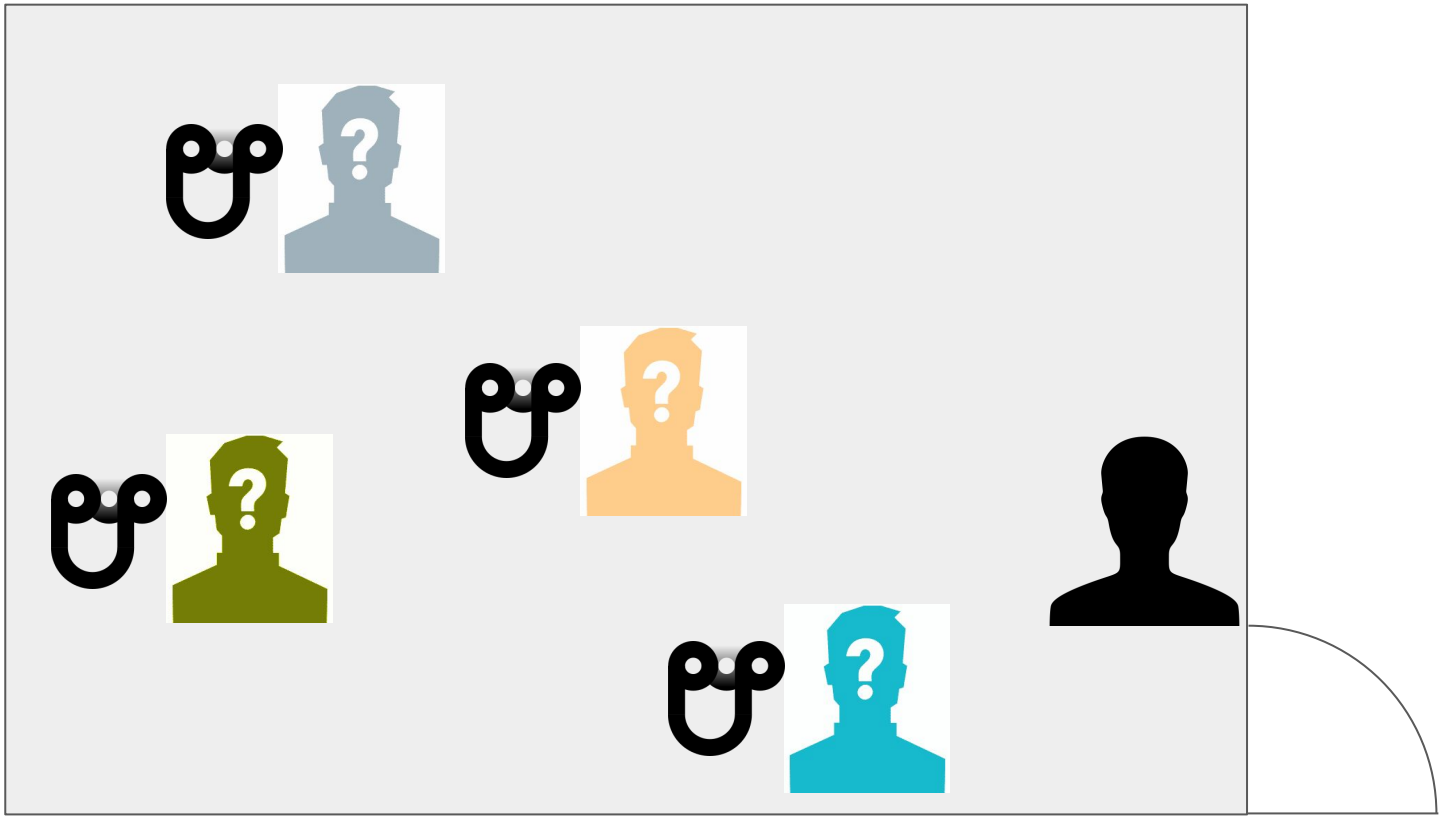




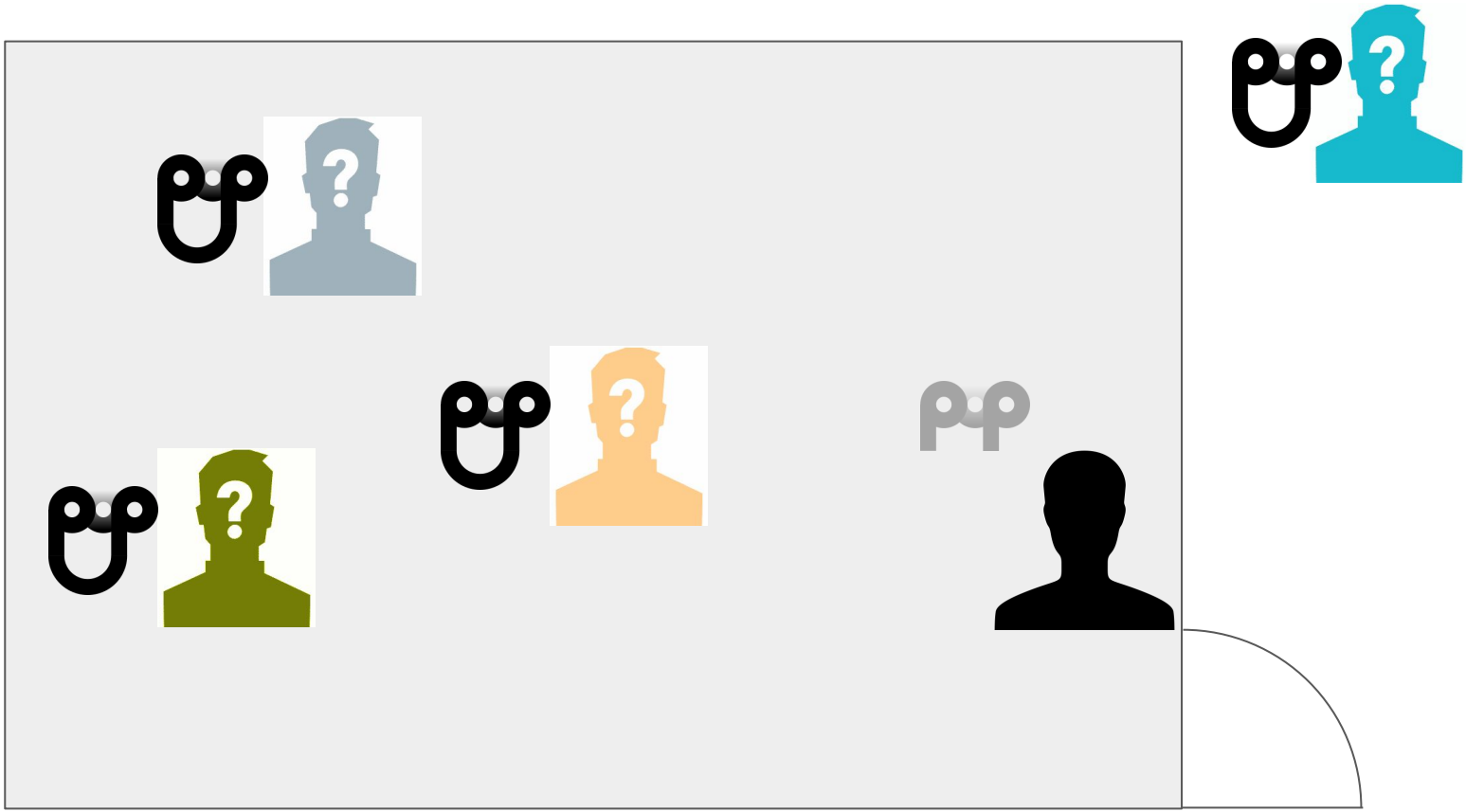
Public
Private

Room

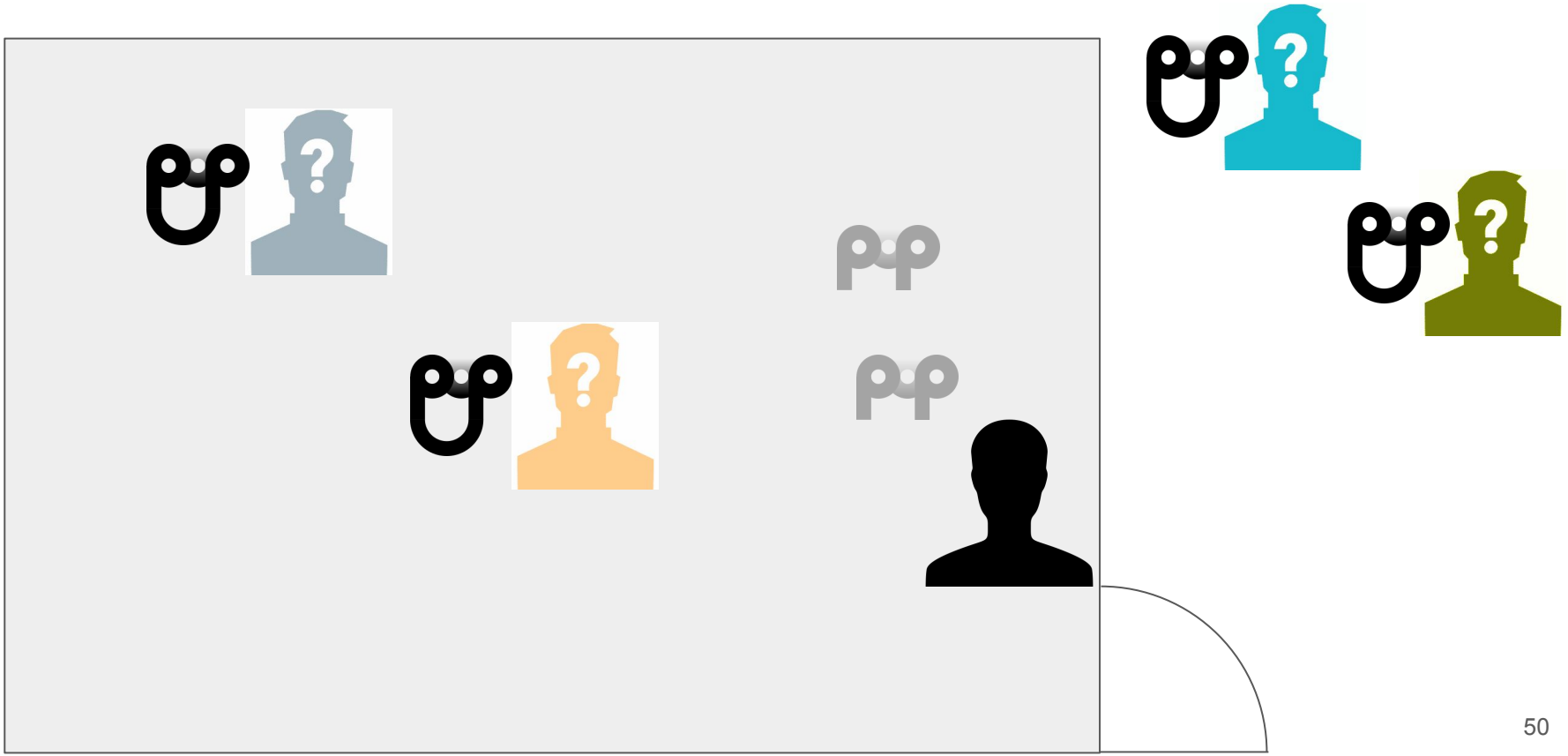
Pseudonym-party



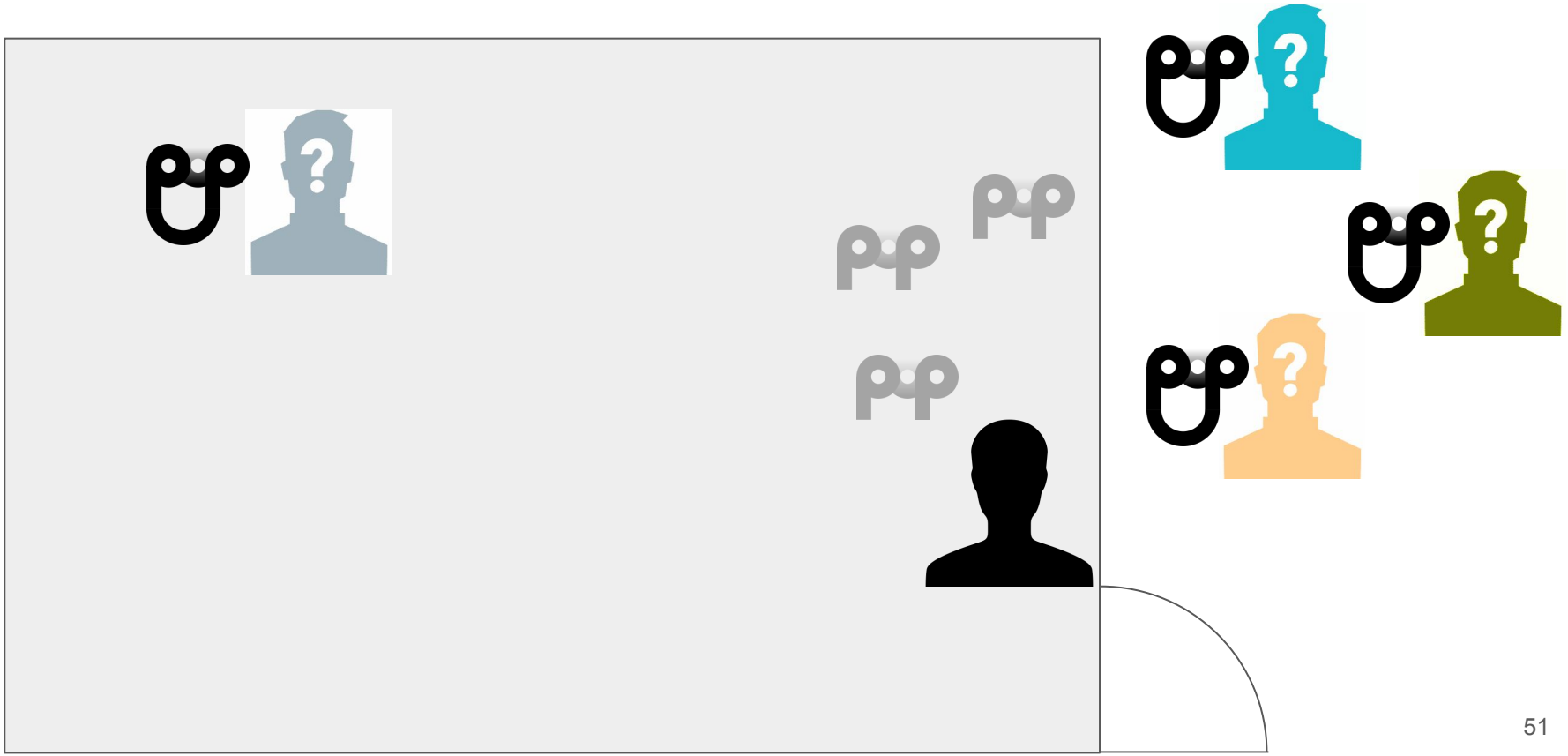
Pseudonym-party



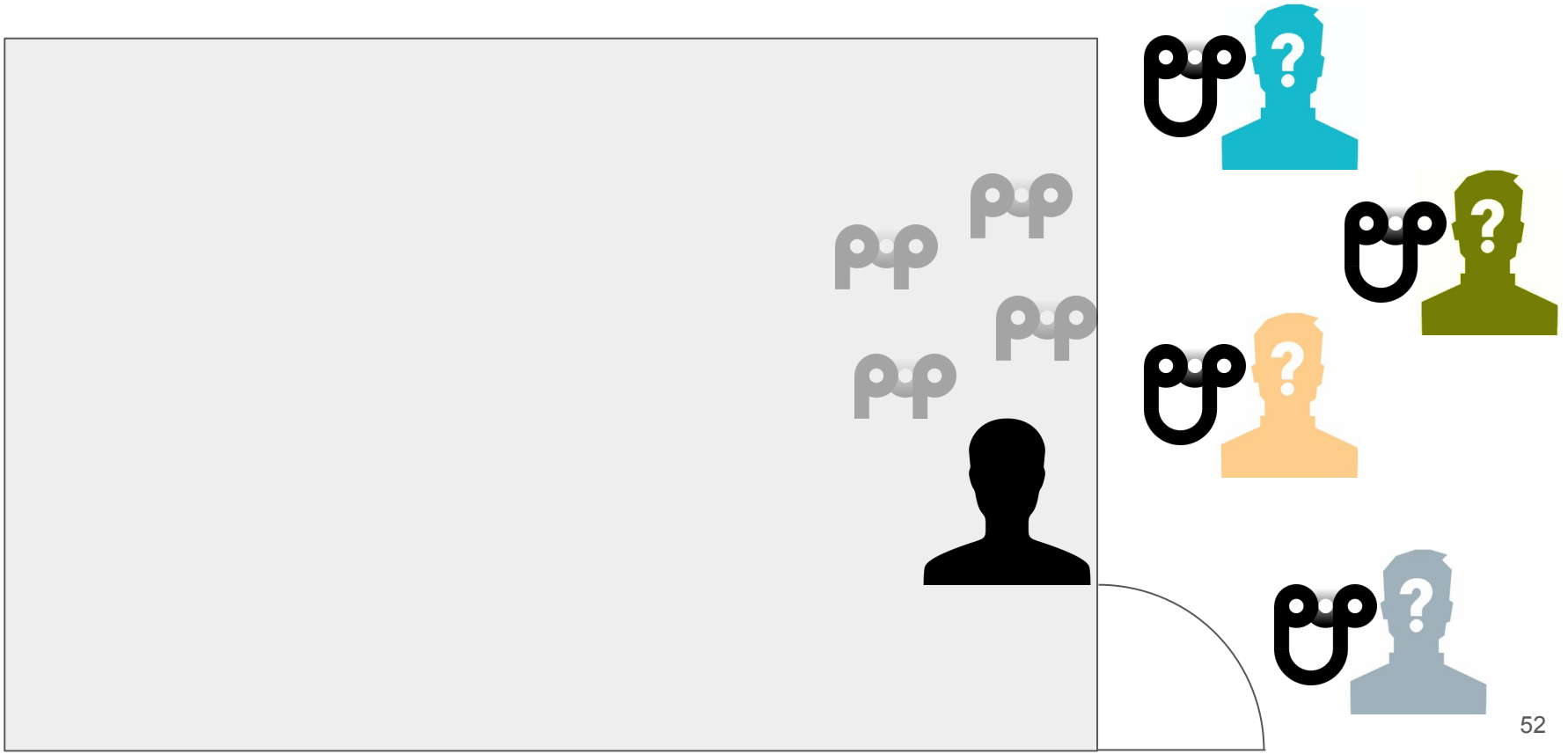
Pseudonym-party



Pseudonym-party

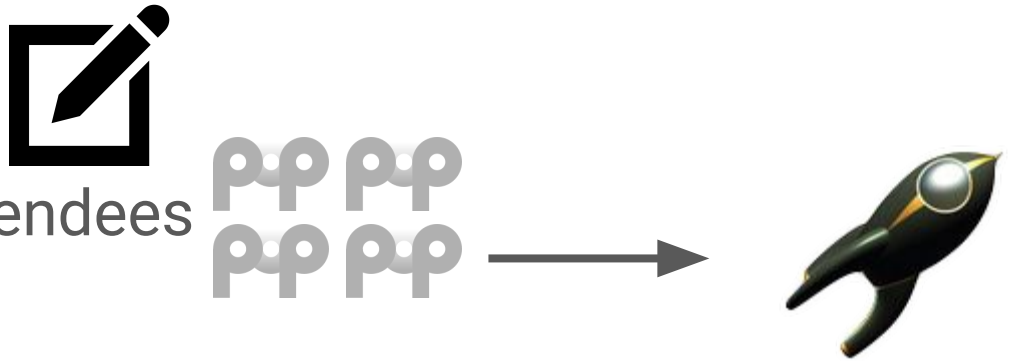


Pseudonym-party



Pseudonym-party - Finalization

- Party transcript:
  - Configuration file
  - Public keys of attendees
  - Party-information
  - Collective signature



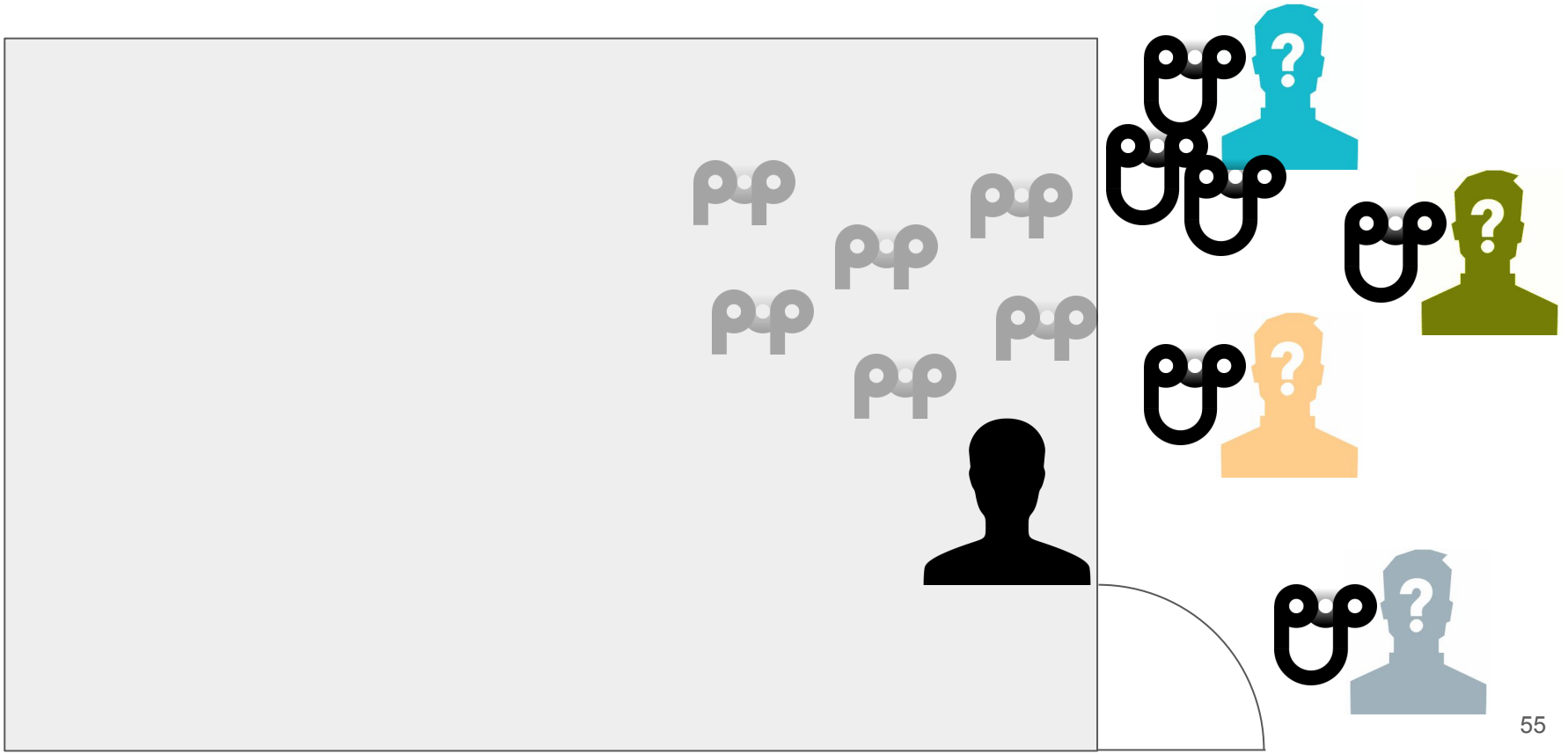
Pseudonym-party - Tokenization

Party transcript

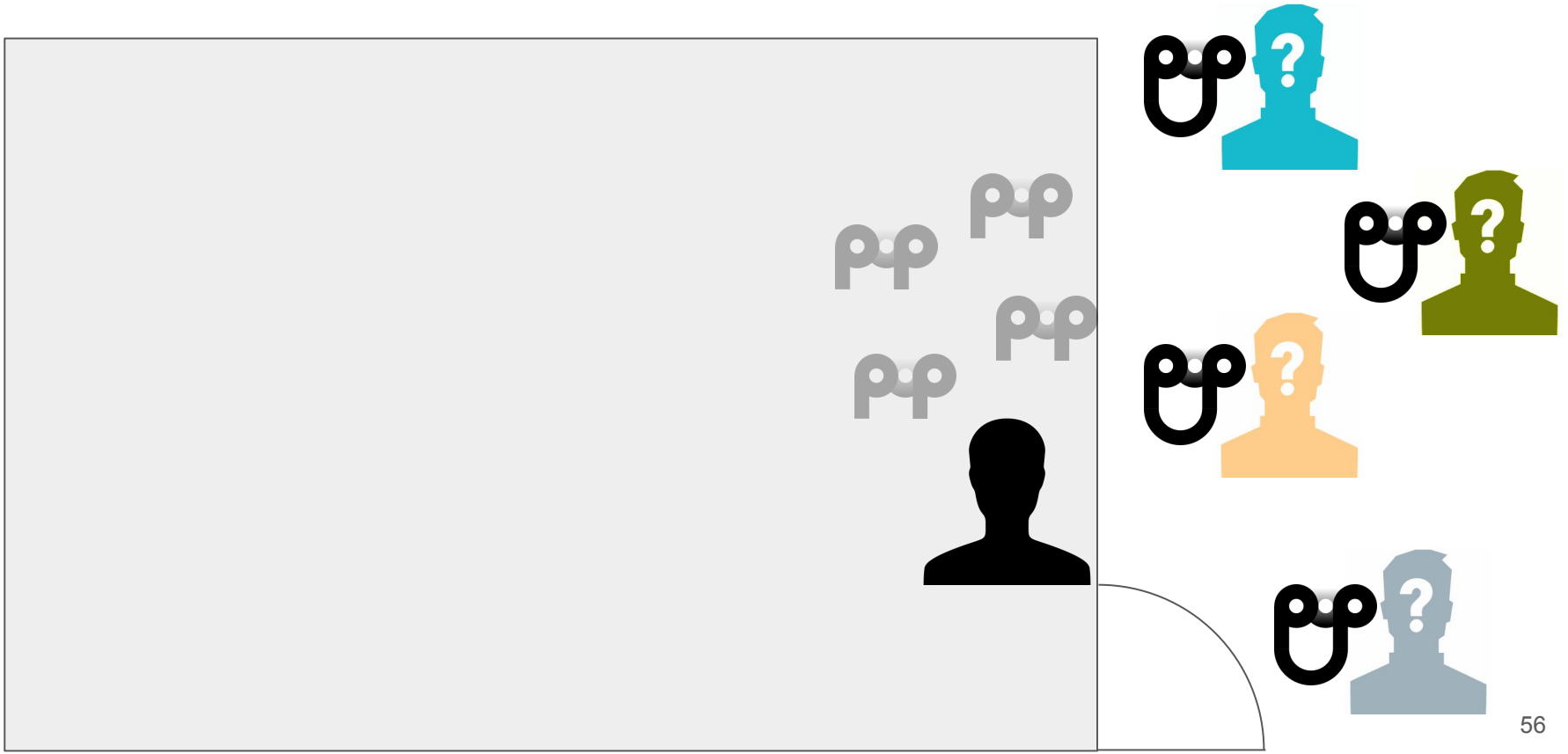
Keypair

PoP-token

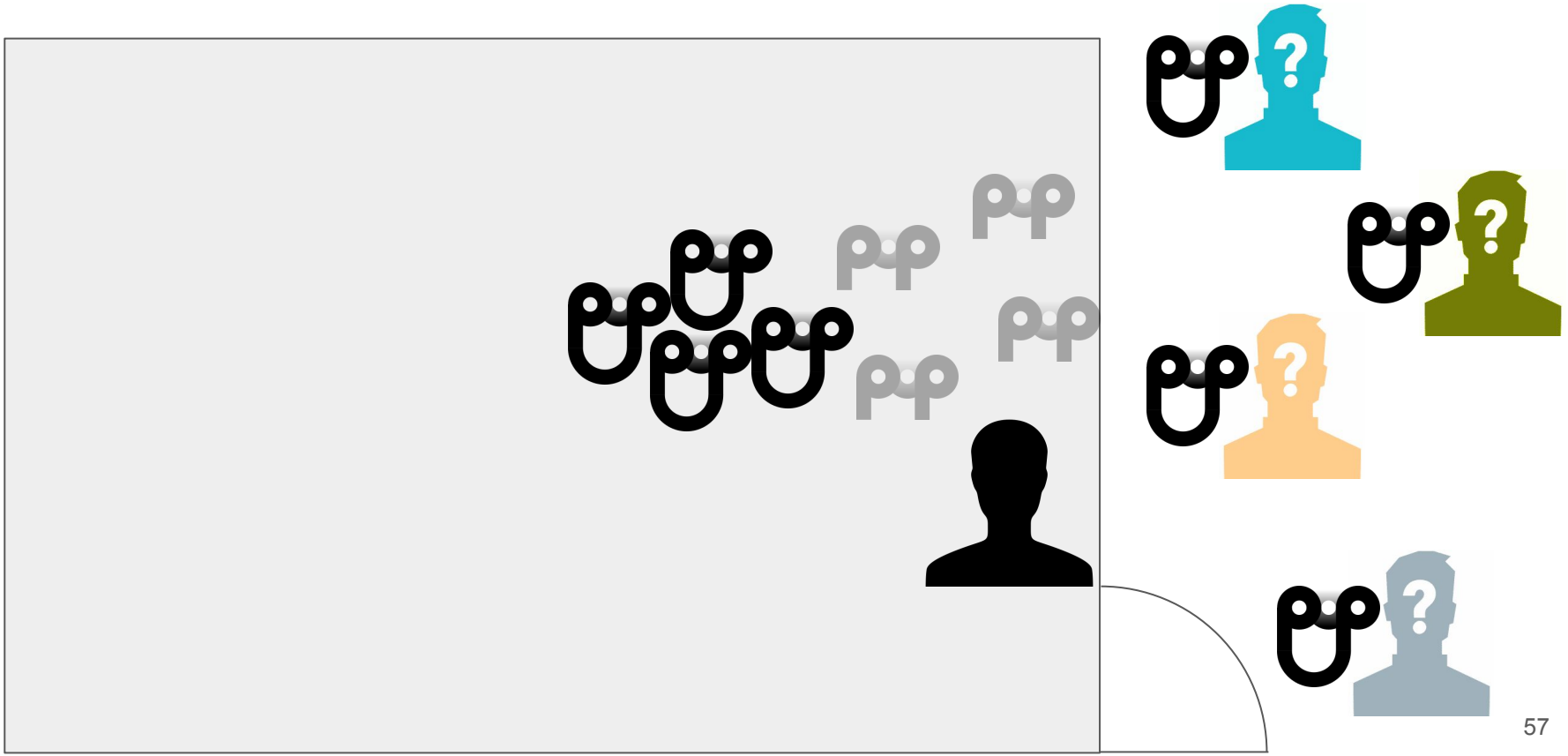
Problem #1 - one attendee getting more tokens



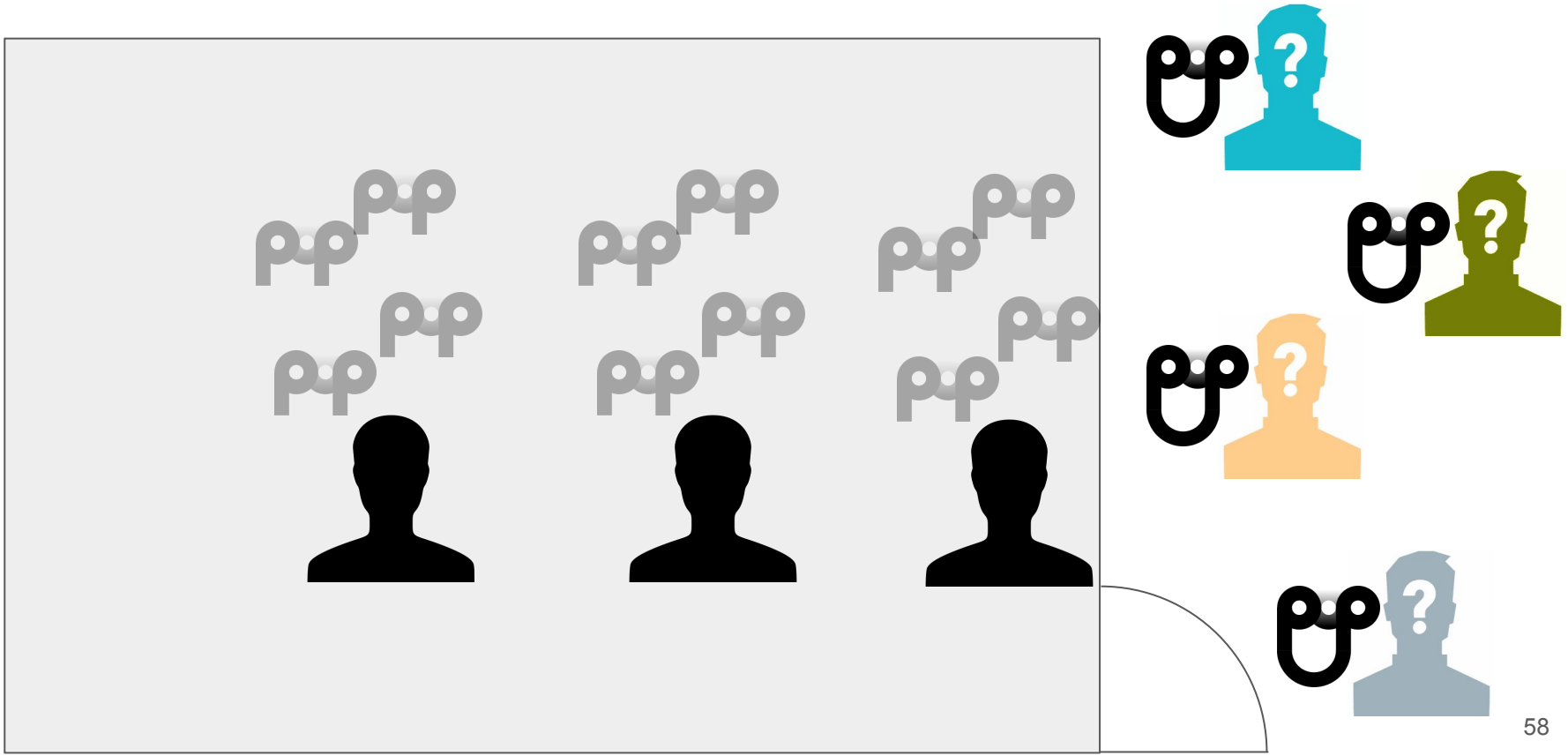
Solution #1 - only one exit, no re-entry



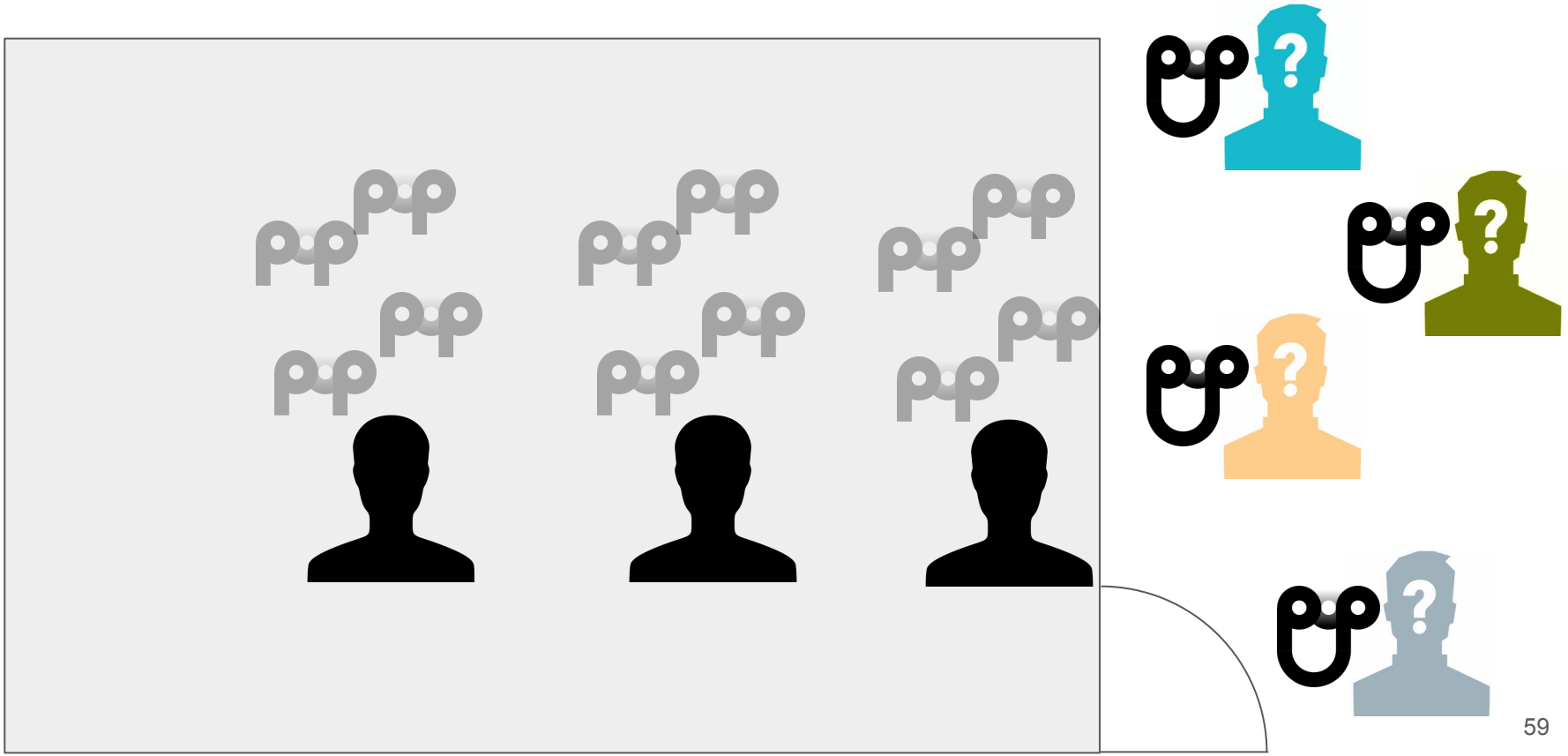
Problem #2 - malicious organizer adds tokens



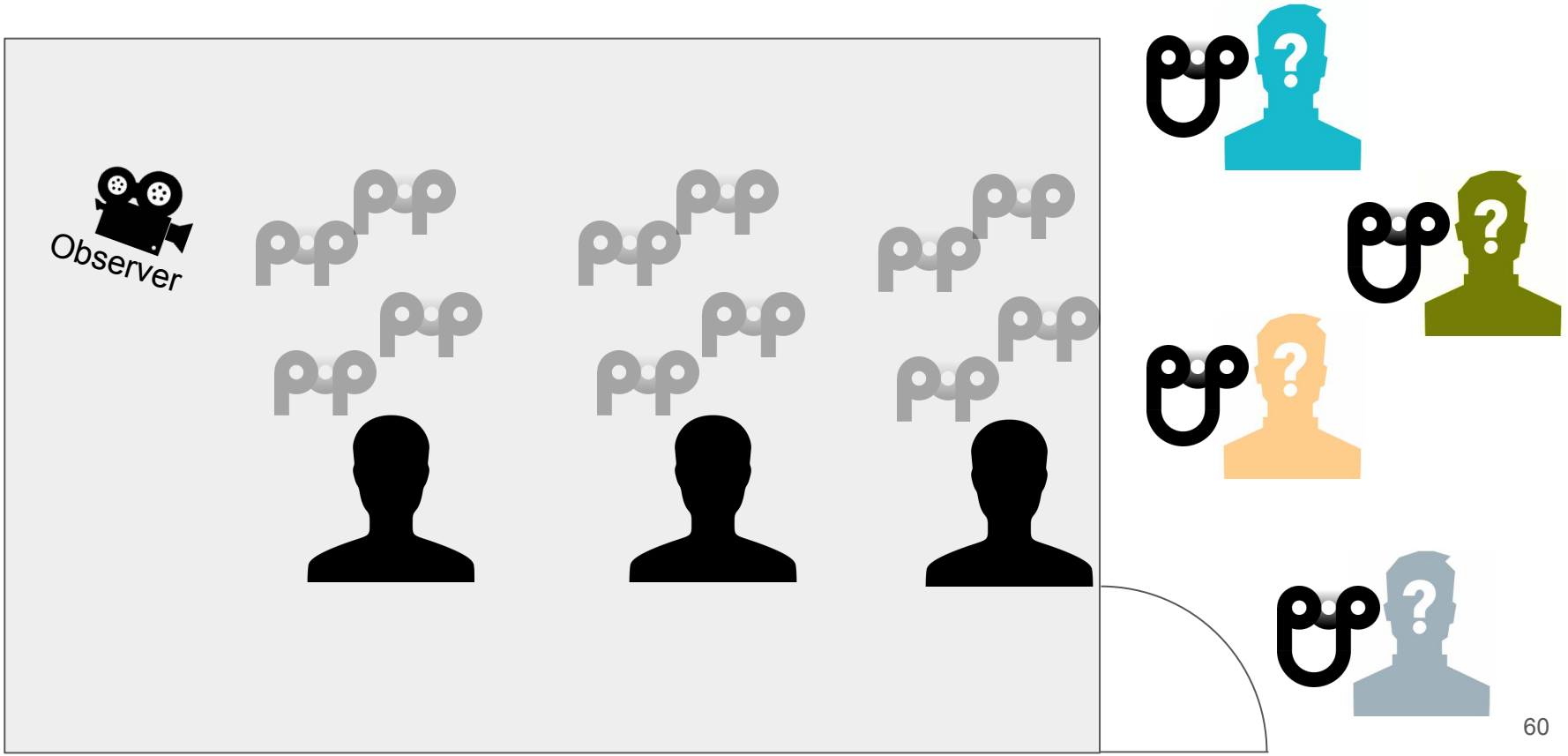
Solution #2 - consensus among multiple organizers



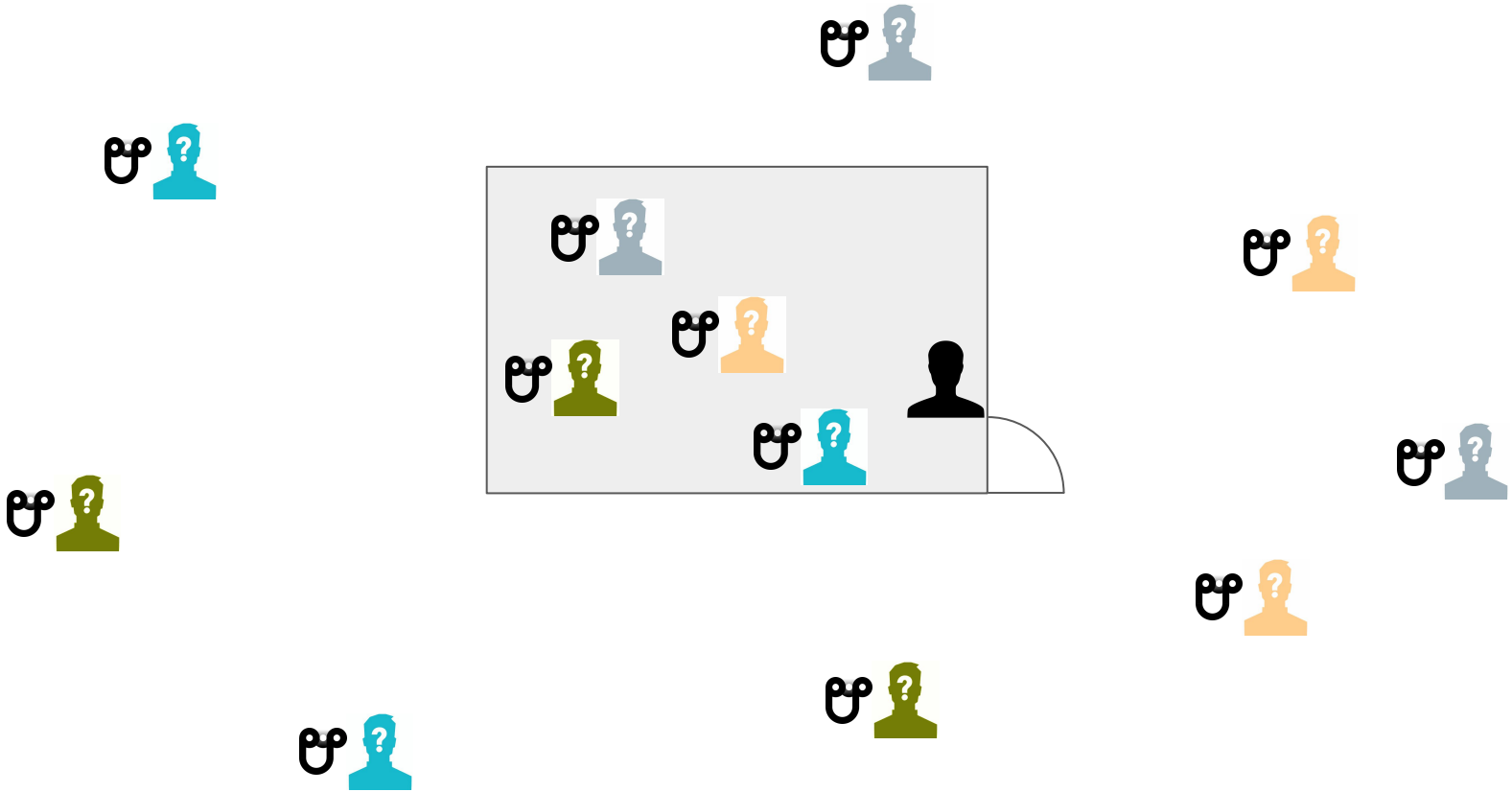
Problem #3 - rejection of an attendee



Solution #3 - record the party

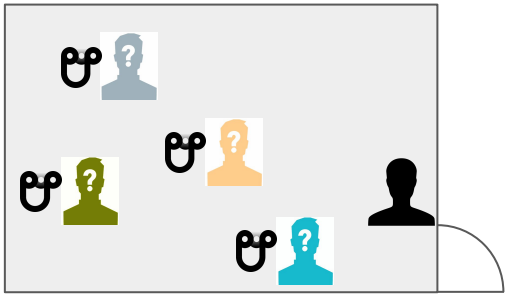


Problem #4 - how to scale?

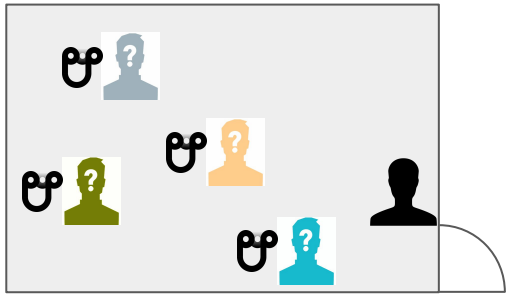


Solution #4 - multiple parties simultaneously

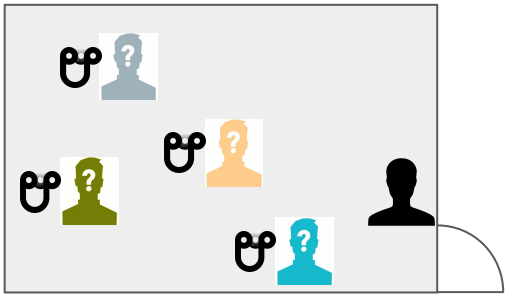
Tuesday, 21st of June 2017, at 6pm UTC+2



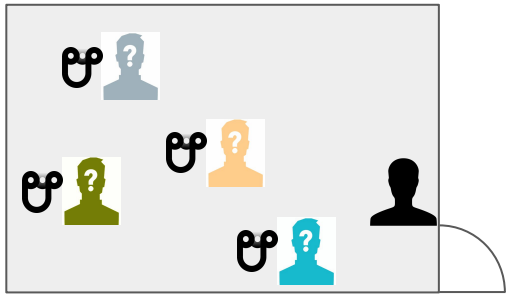
Hamburg



München

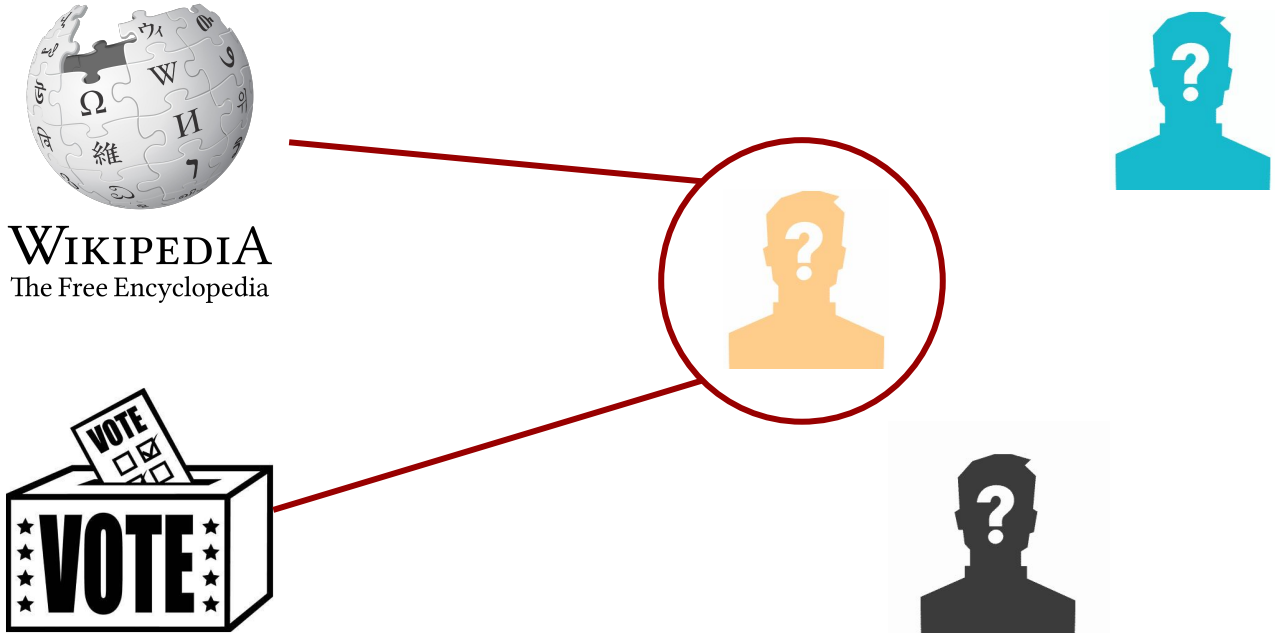


Lausanne

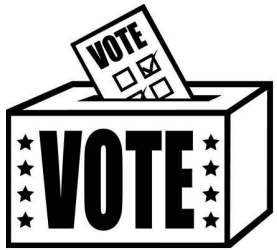


Paris

Problem #5 - Cross-service de-anonymisation



Solution #5 - Use anonymous group signatures



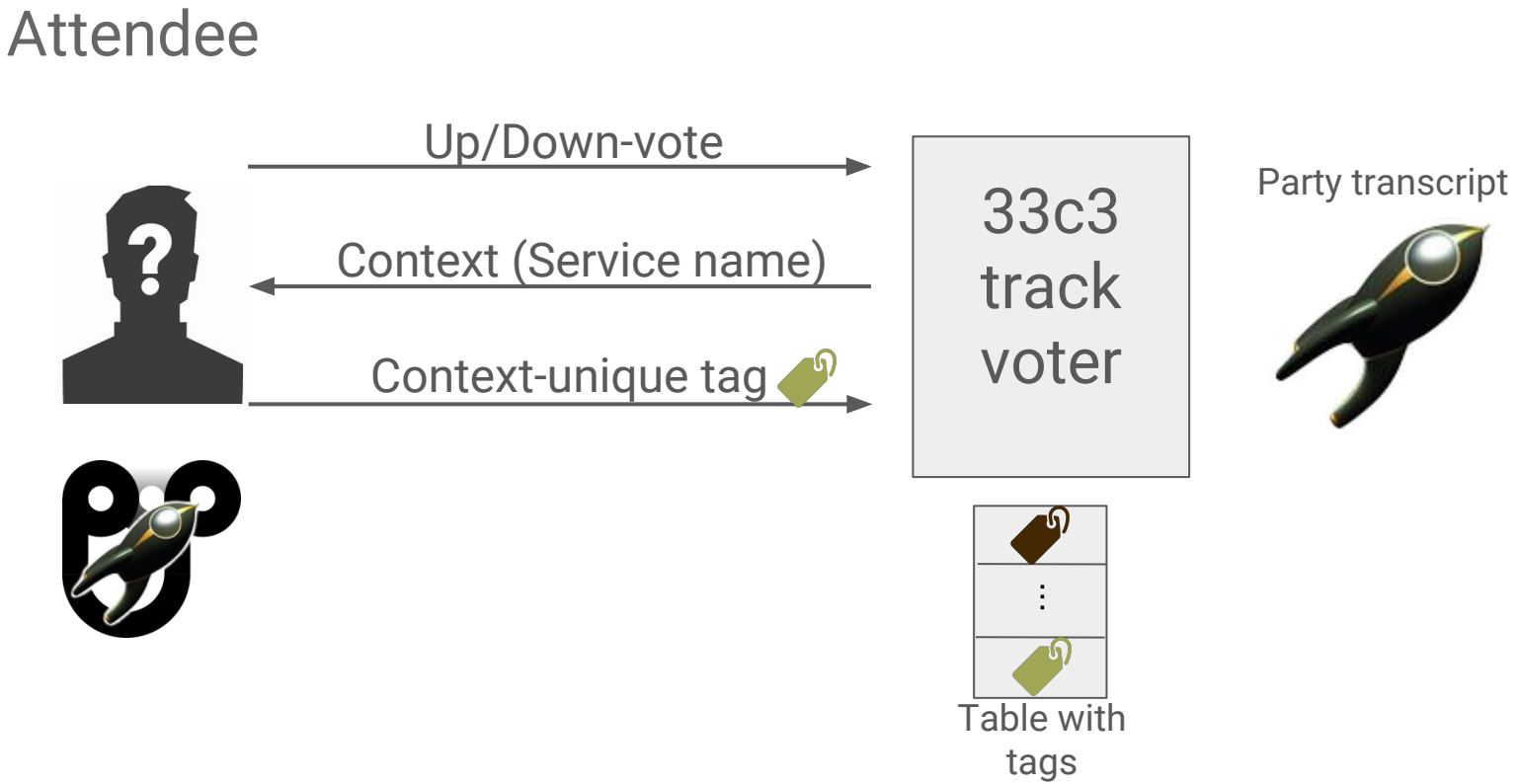
Properties

- One token per person
- No additional tokens if at least one organizer is trusted
- Fairness by observers who record the party
- Scaling through simultaneous parties
- De-anonymisation through anonymous group signatures

Proof of Personhood

- Anonymity vs accountability
- Proof of personhood (PoP)
- Pseudonym party
- **Usage of PoP-tokens**
- Possible applications

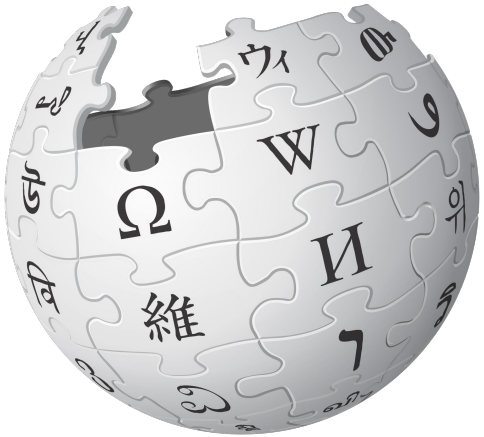
PoP-Tokens for authentication



Proof of Personhood

- Anonymity vs accountability
- Proof of personhood (PoP)
- Pseudonym party
- Usage of PoP-tokens
- **Possible applications**

Accountability <-> Anonymity

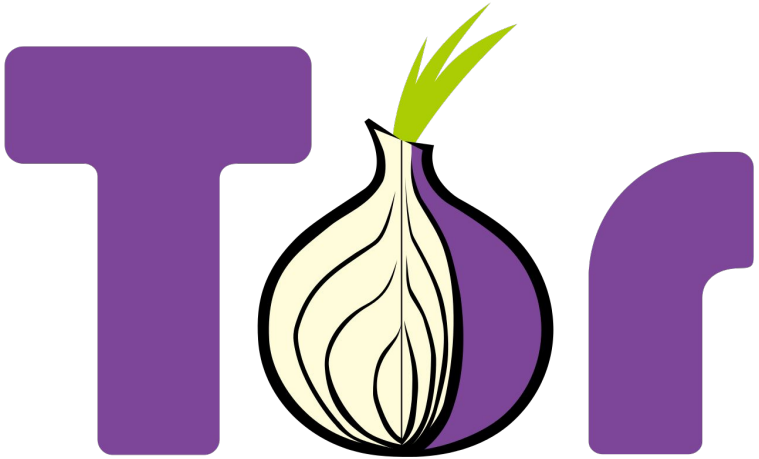


WIKIPEDIA
The Free Encyclopedia

Accountability

Anonymity

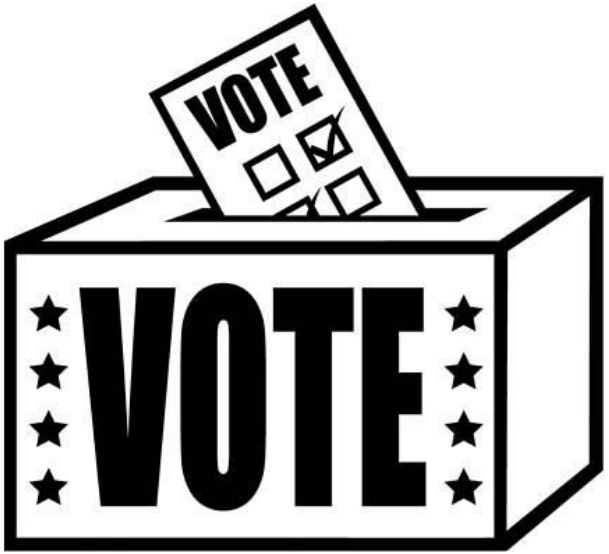
Accountability <-> Anonymity



Accountability

Anonymity

Accountability <-> Anonymity



Accountability

Anonymity

Accountability <-> Anonymity

Accountability

Anonymity

Other projects

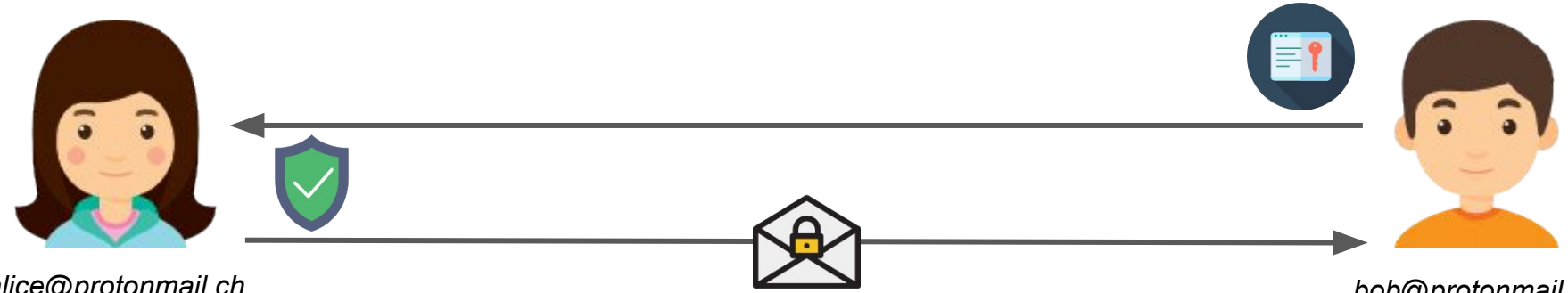
- Cothority - Collective Authority
  - Framework for Decentralized Distributed Systems (DEDIS)
  - Contains the DEDIS-blockchain
  - Many more services like CISC, Pulsar, SkipChains, ByzCoin, ...
- ONet - Overlay Network
  - Framework for writing protocols, services and apps
- Kyber - Cryptographic Library with Basic Primitives
  - For research on elliptic curves, RSA
  - Allows operations that are not available in go standard libraries
- Frontends to Cothority
  - Cross-platform mobile application, javascript libraries and website, Python libraries

Decentralized Identity Management

DEDIS, EPFL

Problem - Identity Discovery

- Alice must know Bob's public key to send him encrypted messages.
- Guarantees on the key's authenticity, freshness, consistency etc… (see requirements) in presence of active adversary.



alice@protonmail.ch

bob@protonmail.ch

Why bother ?

- Certificate authorities (TLS…) are
  - Centralized
  - Full of security holes

- GPG uses directory servers (https://pgp.mit.edu/ etc)
  - A key is valid if there is a **trusted path** from your keys to the recipient's key
    - Not used wildy, difficult to scale because of pgp signing parties
  - Not privacy preserving: refreshing keys disclose your whole list of contact!
  - Not using the trusted path, any directory server can **MitM**.

https://venturebeat.com/2015/03/23/google-security-temporarily-compromised-by-fake-digital-certificates/

Why bother #2

- Gmail **does not** end-to-end encrypt emails
  - Currently working on Key Transparency project
- Whatsapp / Signal have **end to end**

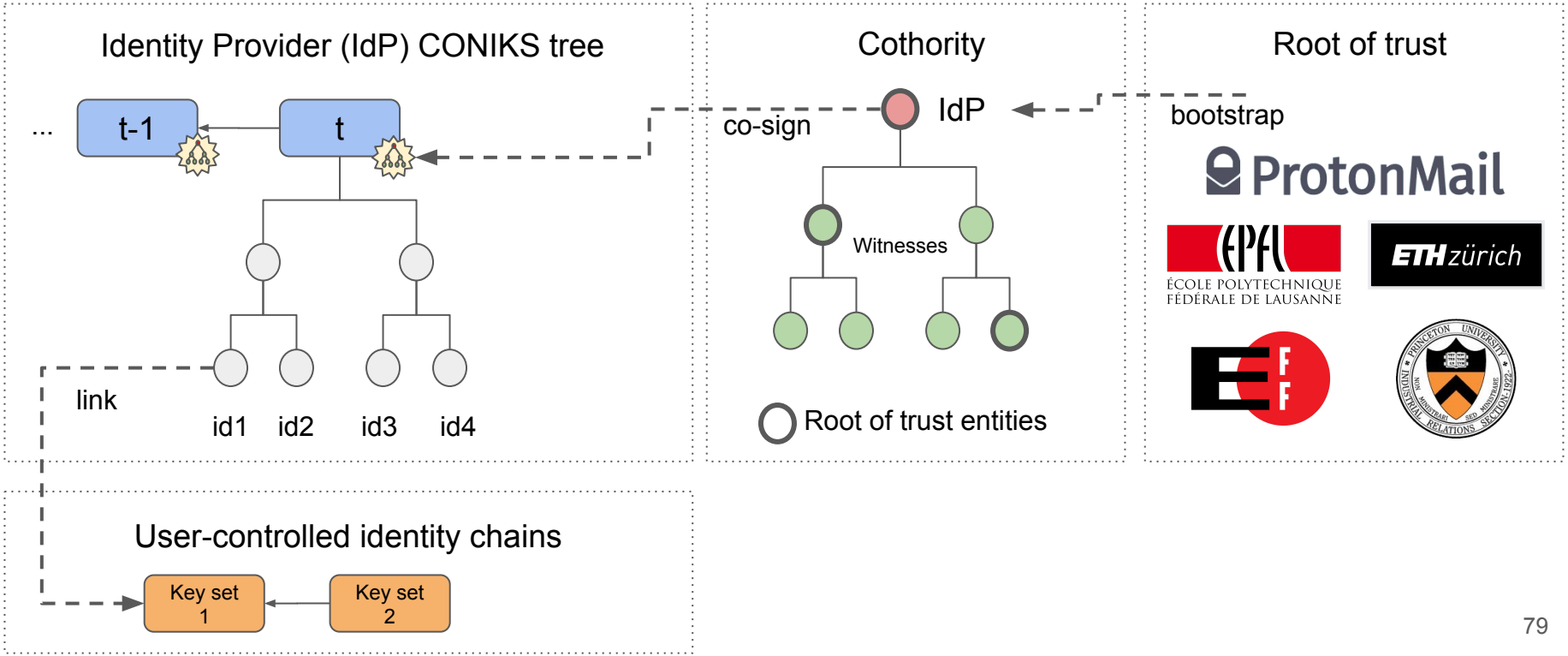
encryption of messages! But...

  - Centralized repository of user keys
  - Can still **MITM**
  - Verification done by QR code scanning…
  - Can’t scan at 10’000 km of distance !

Requirements

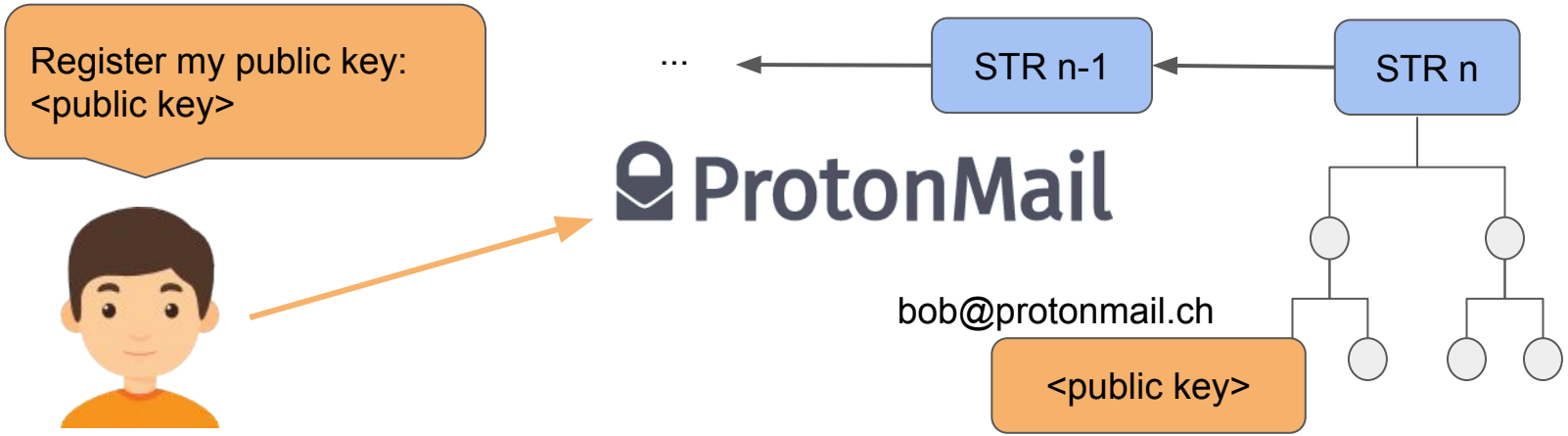
- User friendly
- Identity ownership for the user
- Identity Providers should not be trusted
- Publicly auditable and verifiable records of key changes
  - proof of authenticity, absence, freshness...
- Strong consistency with proactive security
- Scalable
- System open to any identity provider
- Identity recovery
- Multiple keys / devices
- ...

System Overview



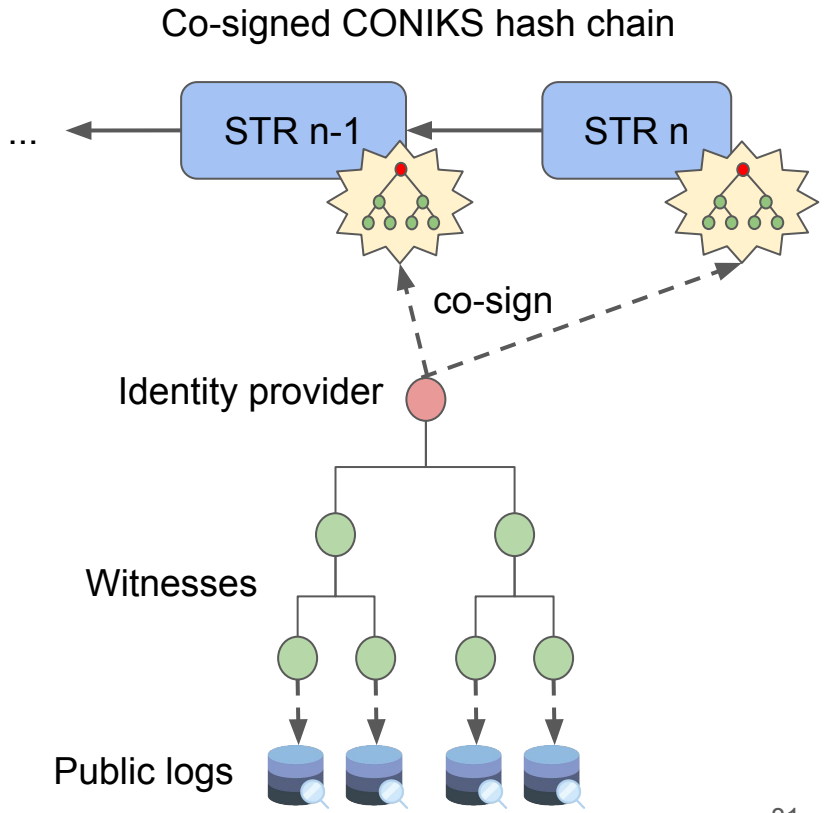
Strawman - CONIKS

- Public keys stored as leaves in Merkle tree
- Path = VUF(email) (~Hash functions)
- Leaf update = new tree
- Signed Tree Roots (STRs) linked in hash chain



Collective Authorities

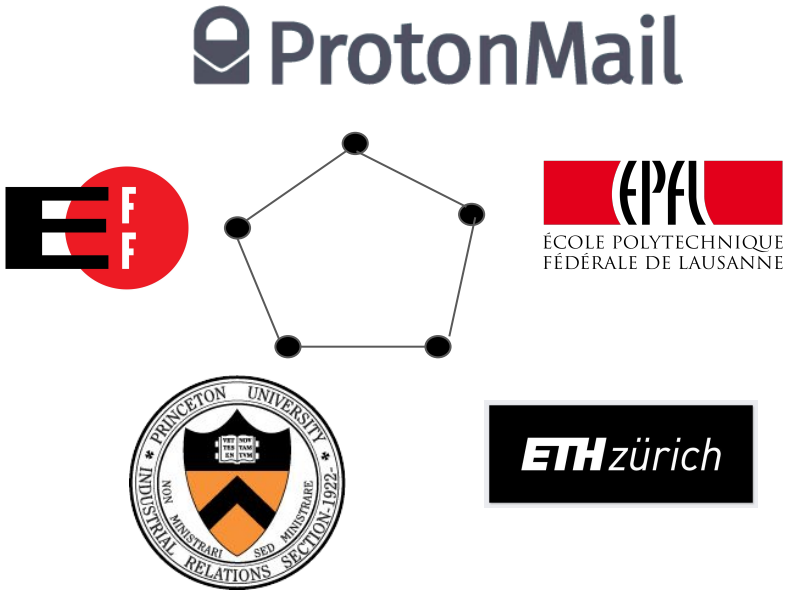
- Collective authorities:
  - Identity provider
  - Witnesses (diverse and independent)
- Witnesses
  - Verify and attest hash chain consistency
  - Provide public auditable logs
- Collective signing (CoSi) provides:
  - Strong consistency
  - Proactive security
  - Thousands of nodes under 2sec



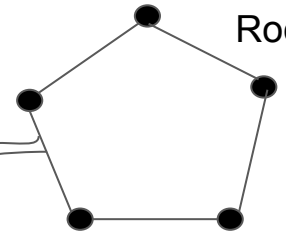
Root of trust

Trustworthy organizations assemble themselves into a cothority (trust splitting).

The cothority is the **root of trust** for the users (i.e. a list of public keys embedded in apps).



Root of trust



Root of trust cothority

Root skipchain:

- Track evolution of root of trust cothority (low frequency).
- Using offline keys
- New members may involve human process.

Protonmail
EPFL
EHTZ
Princeton

Protonmail
EPFL
EHTZ
Princeton
EFF

Namespace skipchain (proposal):

- Track IdP cothorities.

Other possibilities:

- global search engine

Protonmail:
<link, genesis hash>

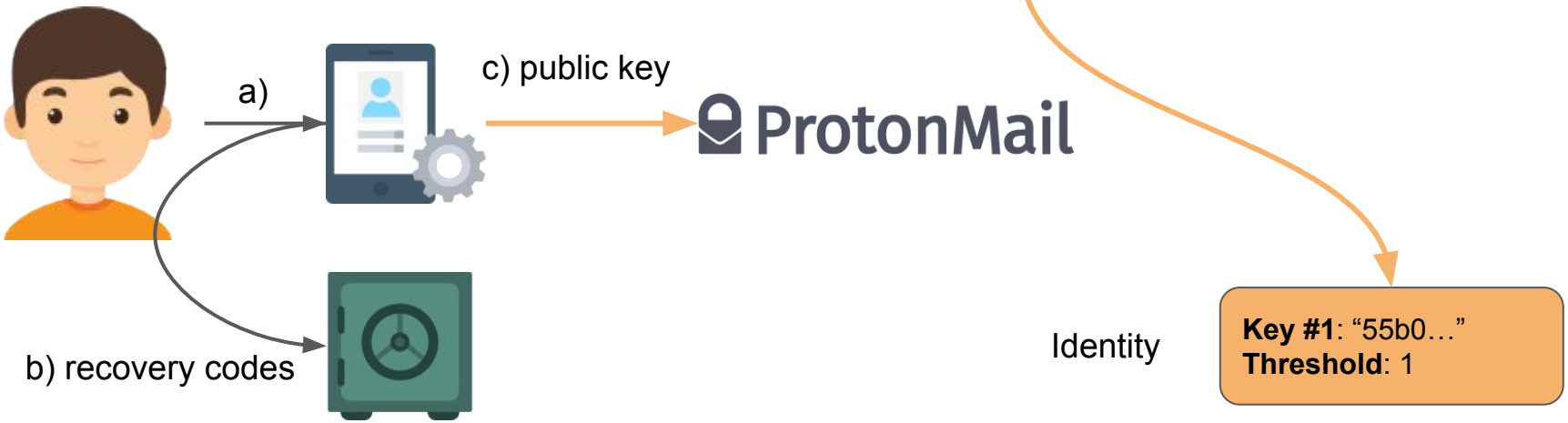
Protonmail:
<link, genesis hash>
EPFL:
<link, genesis hash>

ProtonMail

Typical usage: Registering

Bob creates an identity:

a) Create private / public key pair
b) Save its recovery codes (TBD)
c) Upload public key to Protonmail



Typical usage: Fetching key

Alice wants to retrieve Bob's public key:

a) Fetches the latest cothority set (cached)

b) Get Protonmail's skipchain (cached)

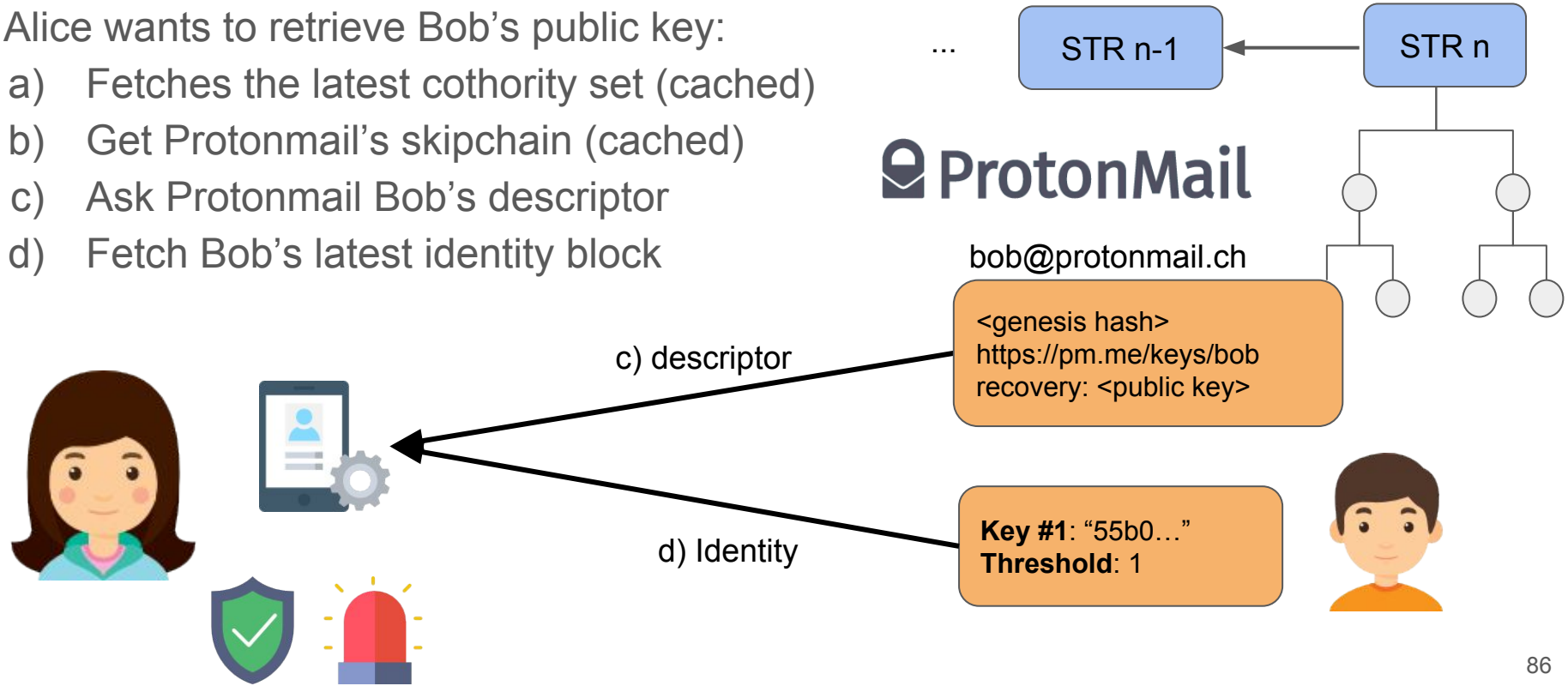
a) Protonmail, EPFL, ETH...

b) Protonmail: <link, genesis hash>

Typical usage: Fetching key

Alice wants to retrieve Bob's public key:

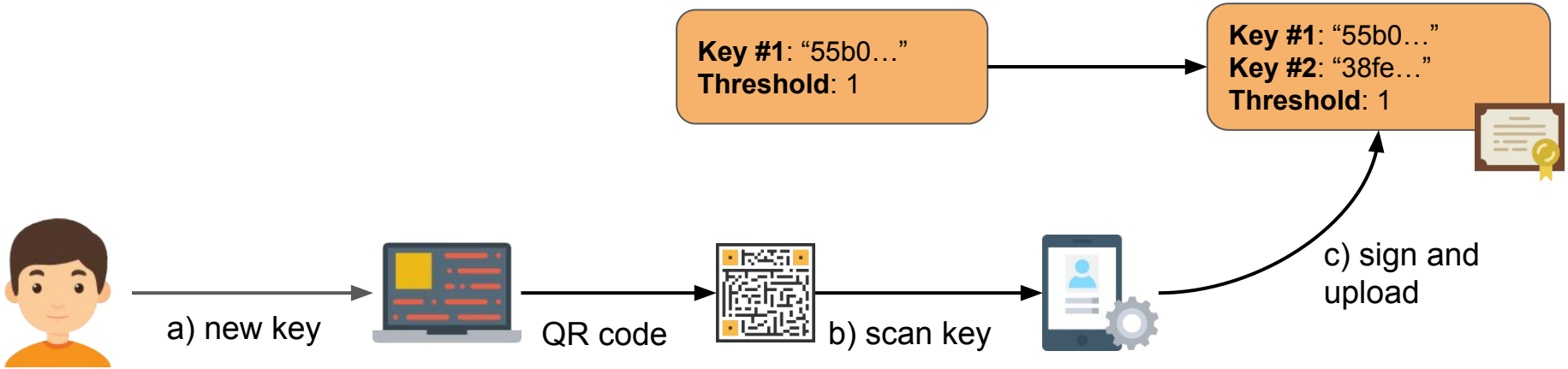
a) Fetches the latest cothority set (cached)
b) Get Protonmail's skipchain (cached)
c) Ask Protonmail Bob's descriptor
d) Fetch Bob's latest identity block



Typical usage: Add a Key

Bob wants to add key for its laptop

a) Create a new key with its laptop
b) Scan the QRcode encoded public key
c) Sign off new block and upload



Conclusion

Decentralized

- Scalable to thousands of witnesses / IP
- Openness of the system
- User owned identity

Secure

- Pro active security
- Strong consistency
- Trust splitting at scale

Transparent

- Tamper proof logs
- witnesses' public logs