PeaceFounder
evoting for democracy

PeaceFounder

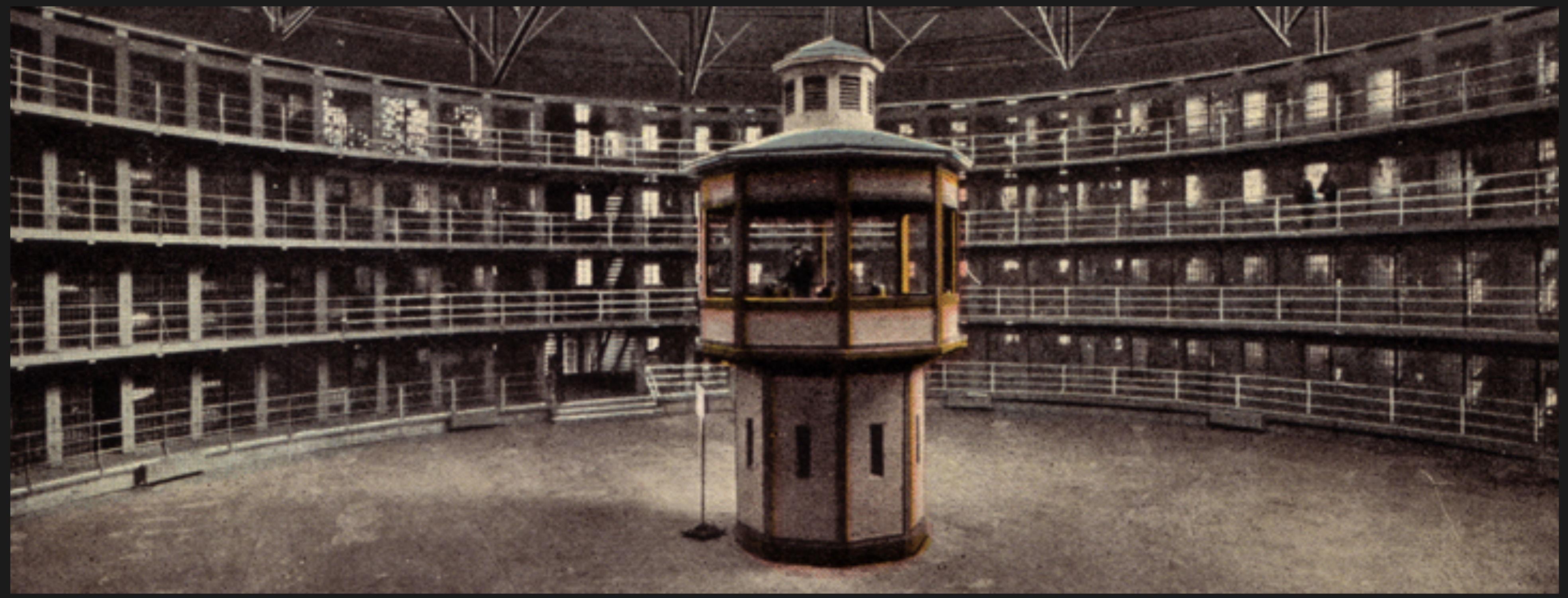
Unveiling Full Stack Development

Dr. Janis Erdmanis

GitHub: JanisErdmanis, janiserdmanis.org

EVoting

Attack vectors



- Surveillance
  - Fear of expressing voters' true choices
  - Coercion/Bribery

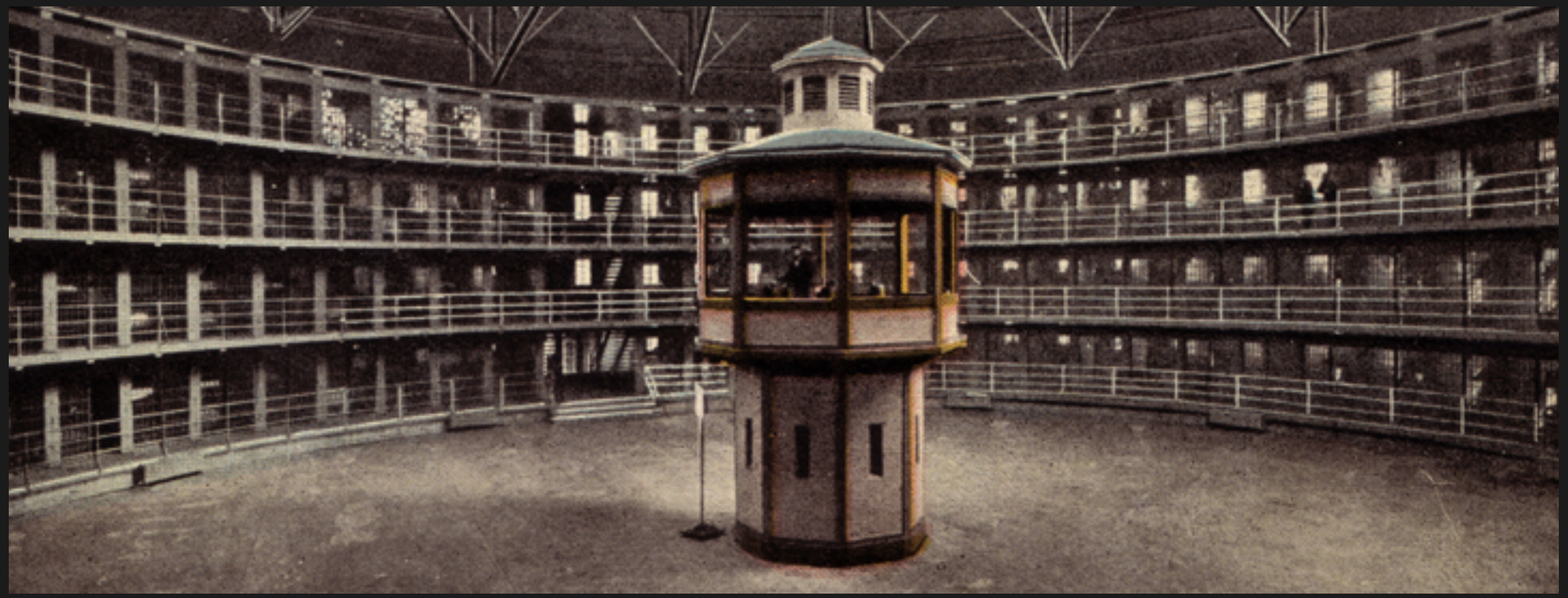
- Deception
  - Presenting secretly manipulated election outcomes as the result
  - Adversary convinces the public that the result can't be trusted
  - Malware on the device lies to voters how the vote is cast

- Sabotage
  - Election result unannounced
  - Casting a vote is not possible due to a DDOS attack or due to corrupt authority

EVoting

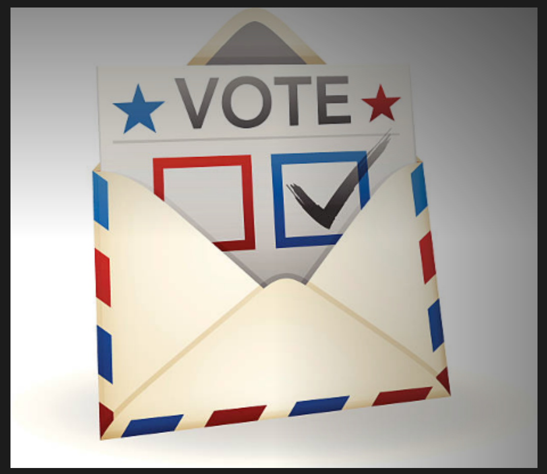
Desirable properties

- Surveilance
    - Anonimity
    - Receipt freeness
- Deception
    - E2E verifiability (individual and universal verifiability)
    - Eligibility verifiability
- Sabotage
    - Robustness
    - Availability



Typical E2E-V EVoting

Setup: m_A and m_B encodes choice for candidate A or B. Public key pk.



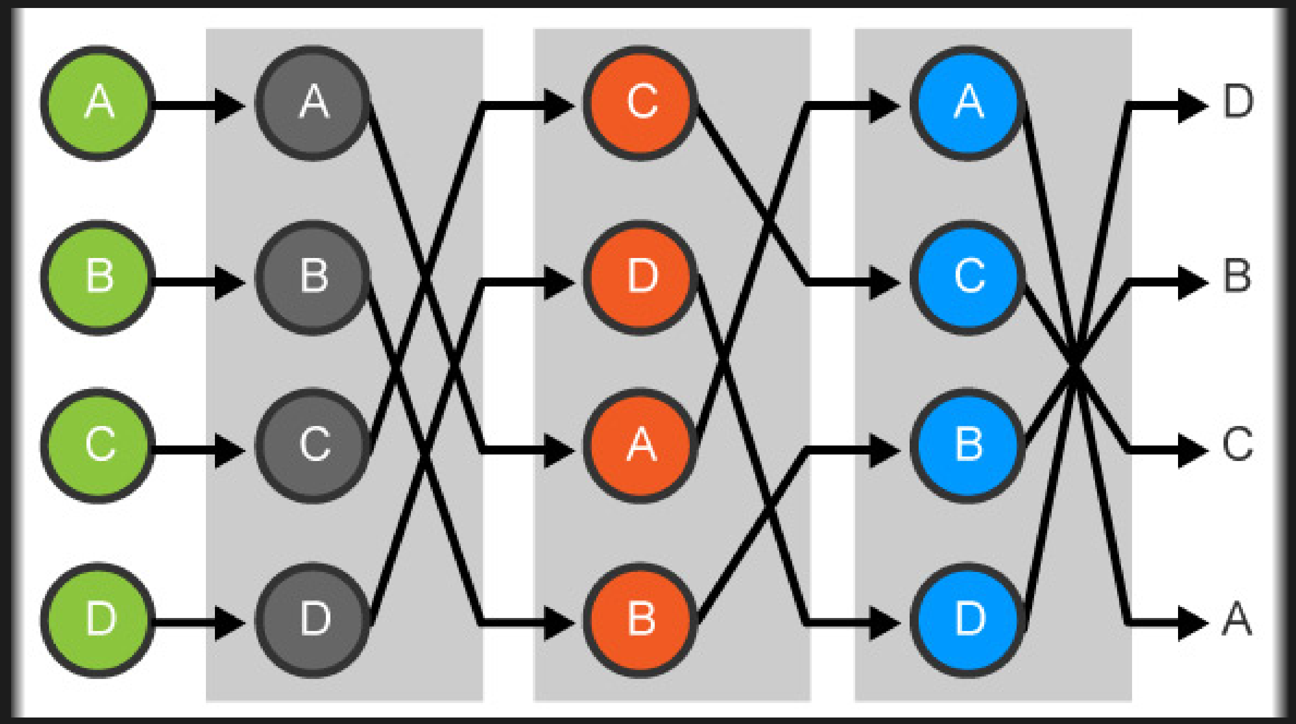
Encrypted Votes

$$A : m = 4$$
$$B : m = 9$$

$$V_1 : (g^{r_1}, m_A * pk^{r_1})$$

$$V_2 : (g^{r_2}, m_A * pk^{r_2})$$

$$V_3 : (g^{r_3}, m_B * pk^{r_3})$$



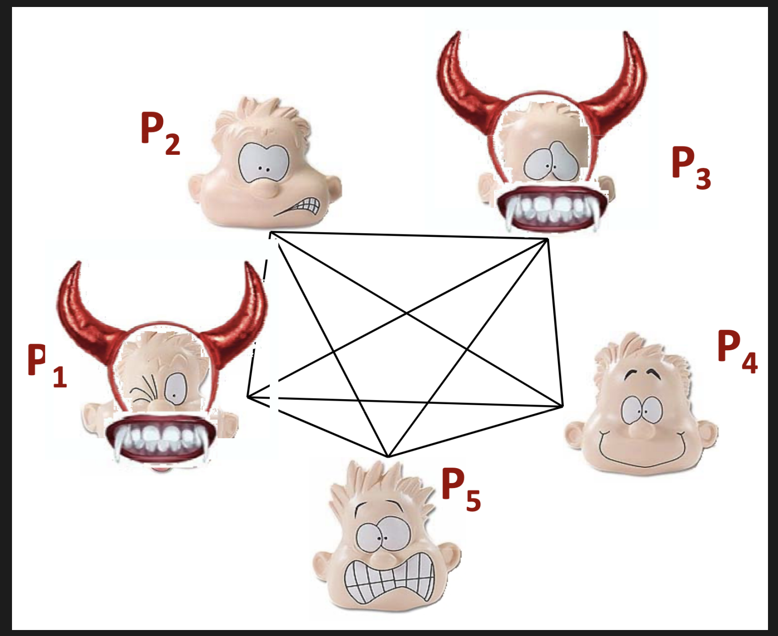
Mix Cascade

Input

	(a, b)
V_1	(9, 16)
V_2	(4, 9)
V_3	(13, 12)

Output

(a', b')
(4, 9)
(8, 16)
(16, 3)



Decryption Ceremony

Counting

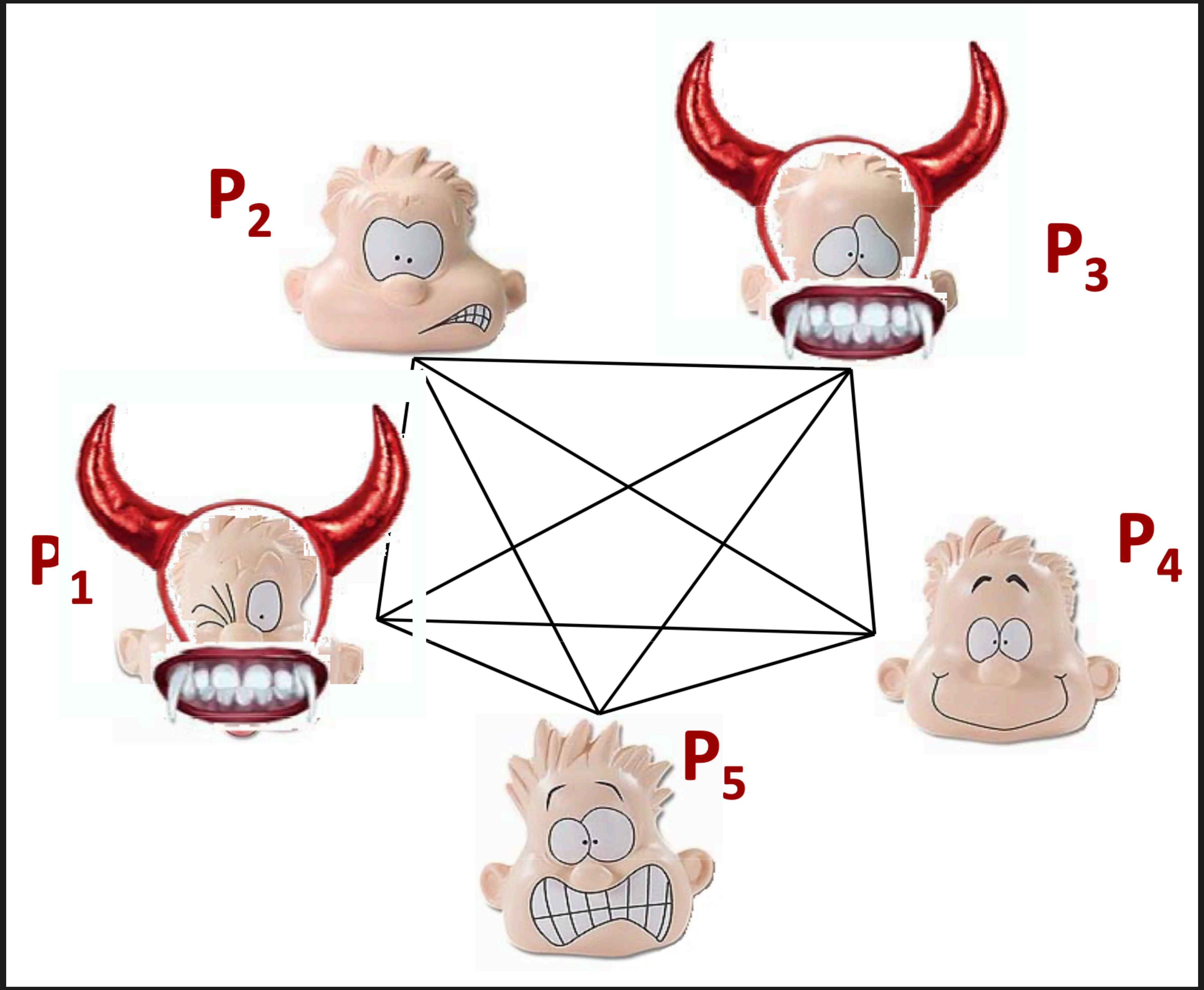
Decrypted Tally:

m
4
9
4

Deployment Complexity

Threshold Decryption Ceremony

- To ensuring vote privacy, the key must be distributed between multiple independent parties;
- In a large threshold, a corrupt minority could sabotage the description of the election result;
- A low threshold low risks corrupt minority to reconstruct the key and see how each voter had voted;
- Dishonest parties can be identified, but it may also be incompetence;
- Hence, privacy and robustness are in tension and ensuring their security is costly



Secure deployment of existing E2E verifiable voting systems is unfeasible for small and medium sized communities

PeaceFounder

DSA Signatures

Let's consider g to be a group element of a cryptographic group

$$sig_g \leftarrow \text{sign}(m, g, sk) \qquad X \leftarrow g^{sk}$$

If $h \leftarrow g^s$ there is no way to link signatures issued as

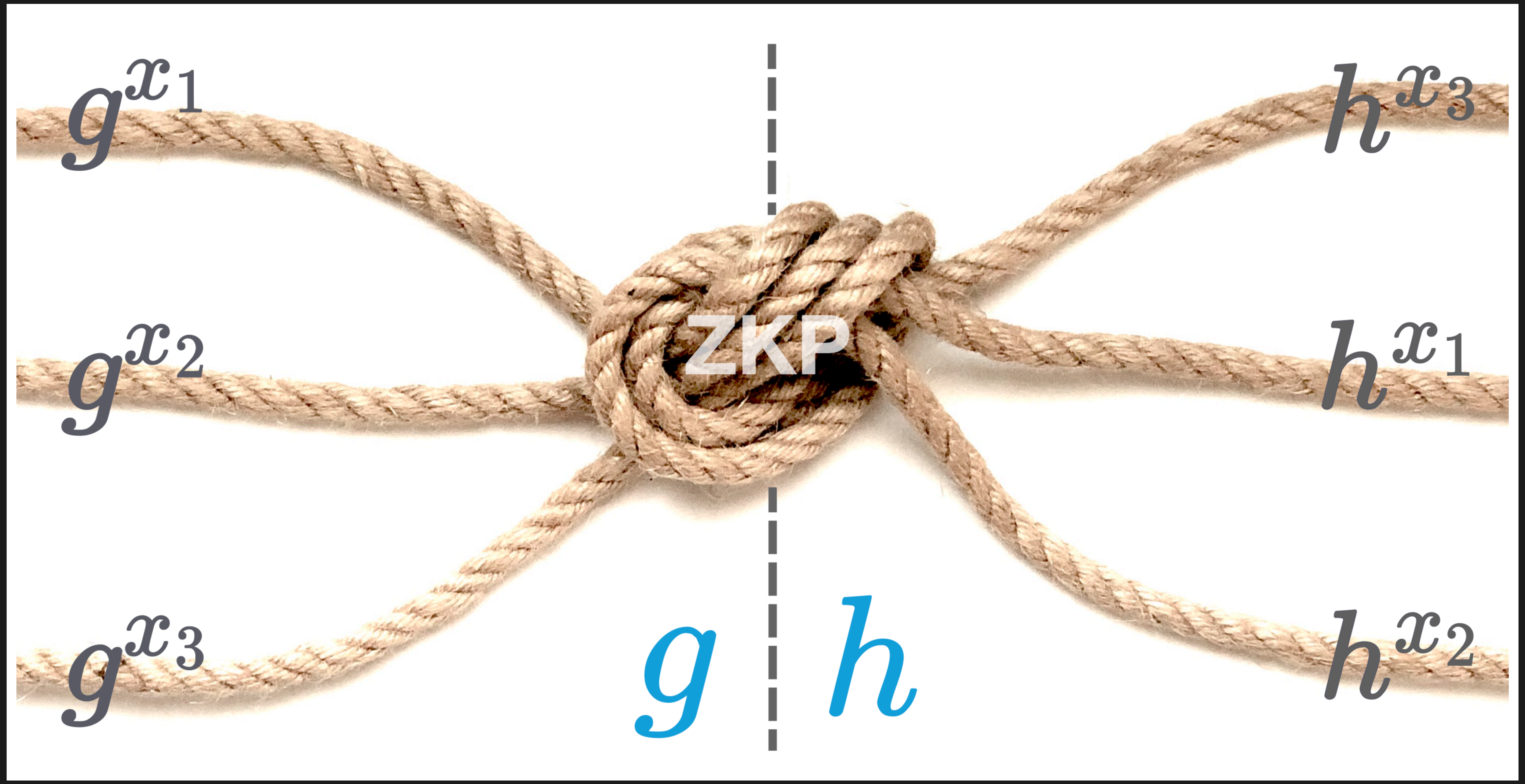
$$sig_h \leftarrow \text{sign}(m', h, sk) \qquad Y \leftarrow h^{sk}$$

Unless s is known to the verifier. Alternatively, zero-knowledge proof of a statement $\log_g(X) = \log_h(Y)$ or $\log_g(h) = \log_X(Y)$ is provided.

PeaceFounder

Braiding

- Exponentiation mix of Haenni & Spycher's proposed construction
- In it, a braider picks a secret factor that exponentiates all input public keys and shuffles them
- Robust zero knowledge ensures the integrity of the braid



Votes signed with relative generator h thus are both anonymous and eligible

PeaceFounder

HistoryTrees.jl

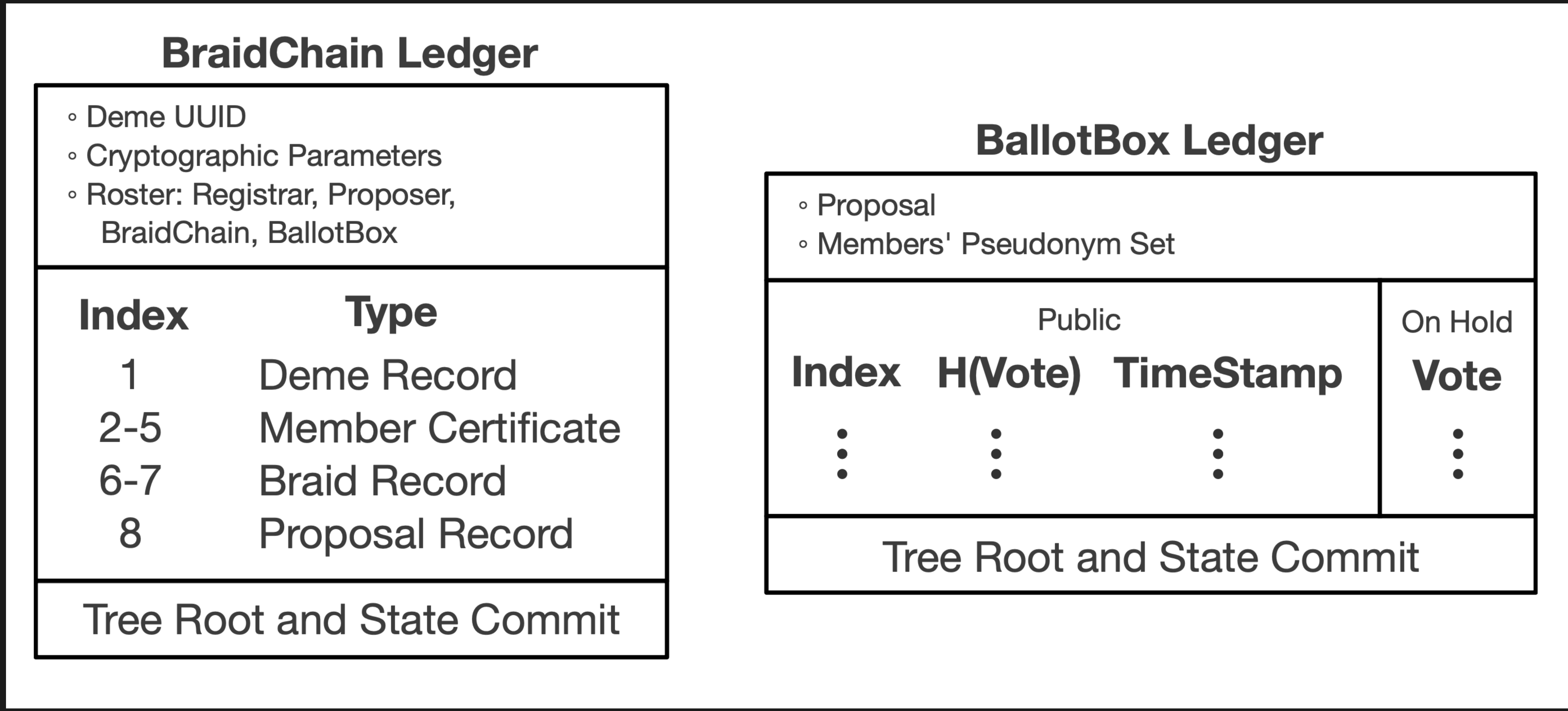
- An extension to Merkle trees with an unbalanced number of entries
- Used for transparency logs to detect malicious certificate authority
- Inclusion proofs are hash chain proofs which link tree roots to the record
- Consistency proofs prove that the current bulletin board commit retains all records from the previous commit

Random queries by thin-voting clients can ensure bulletin board immutability without replication.

PeaceFounder

Buletin Board Structure

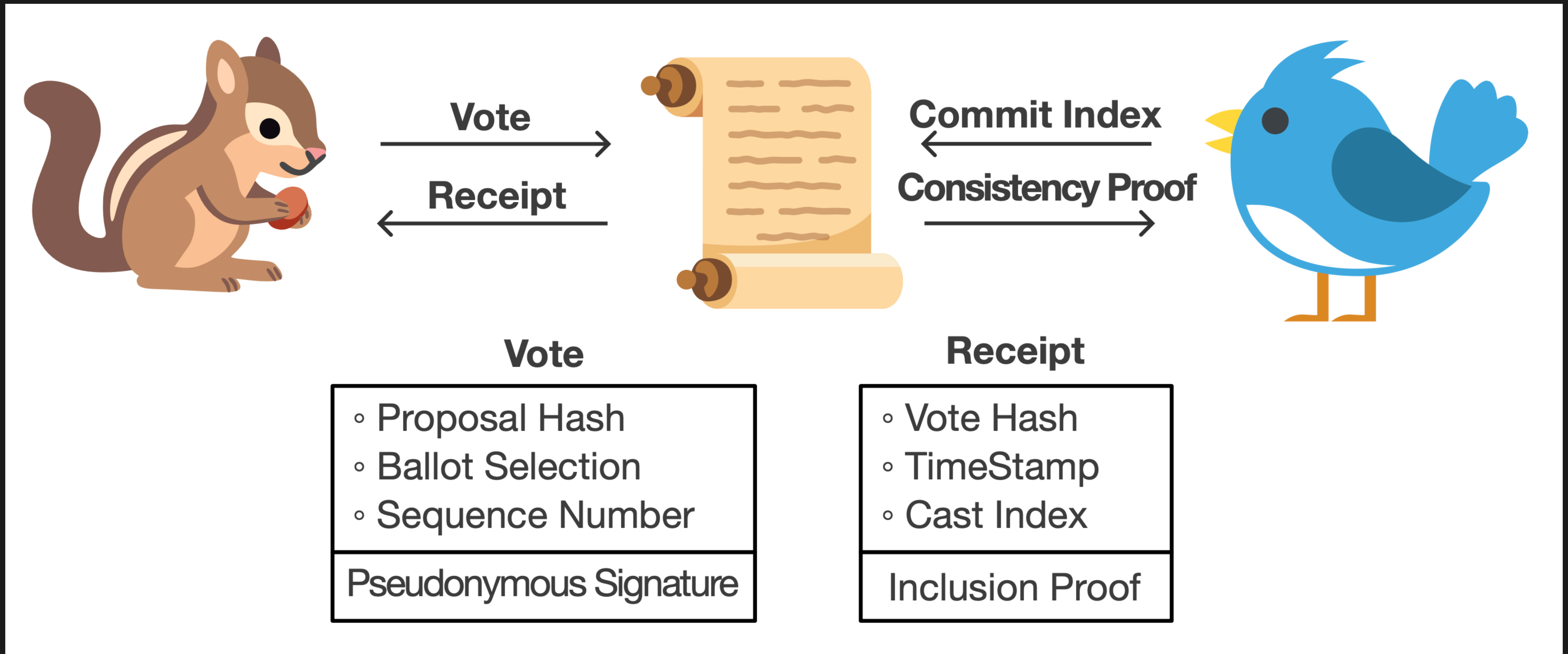
- PeaceFounder is designed around asynchronicity, the unavailability of braiding resources, and hence long-lived instances (demes).
- The bulletin board is split into BraidChain and BallotBox ledgers;
- For a BraidChain record to be included, it needs to be verified and consistent with the current ledger state;
- A proposal record contains an anchor to the BraidChain ledger's state, which sets a relative generator;
- A BallotBox ledger is initialised with a proposal and corresponding members' pseudonym set, which is set by the anchor index in the proposal.



PeaceFounder

Voting

- Every vote signed by a valid pseudonym and associated with a valid proposal hash gets recorded in the BallotBox ledger, even if it is superseded or malformed.
- Upon vote recording, a receipt containing an inclusion proof is returned; if the same vote is already recorded, a receipt for it is returned instead.
- A voter keeps a consistency-proof chain and conducts incremental follow-up queries until votes are finalised. This ensures their vote's inclusion as well as votes made by others.
- The BallotBox ledger publicly displays vote hashes for integrity while concealing actual votes for fairness. This can be extended as a coercion/bribery resistance measure as the system is receipt-free during this period.
- A timestamp ensures that malware cannot show a receipt linked to someone else's vote. Meanwhile, a cast index helps locate the specific vote on the ledger.



https://eprint.iacr.org/2024/1040

PeaceFounder Demo

http://peacefounder.camdvr.org

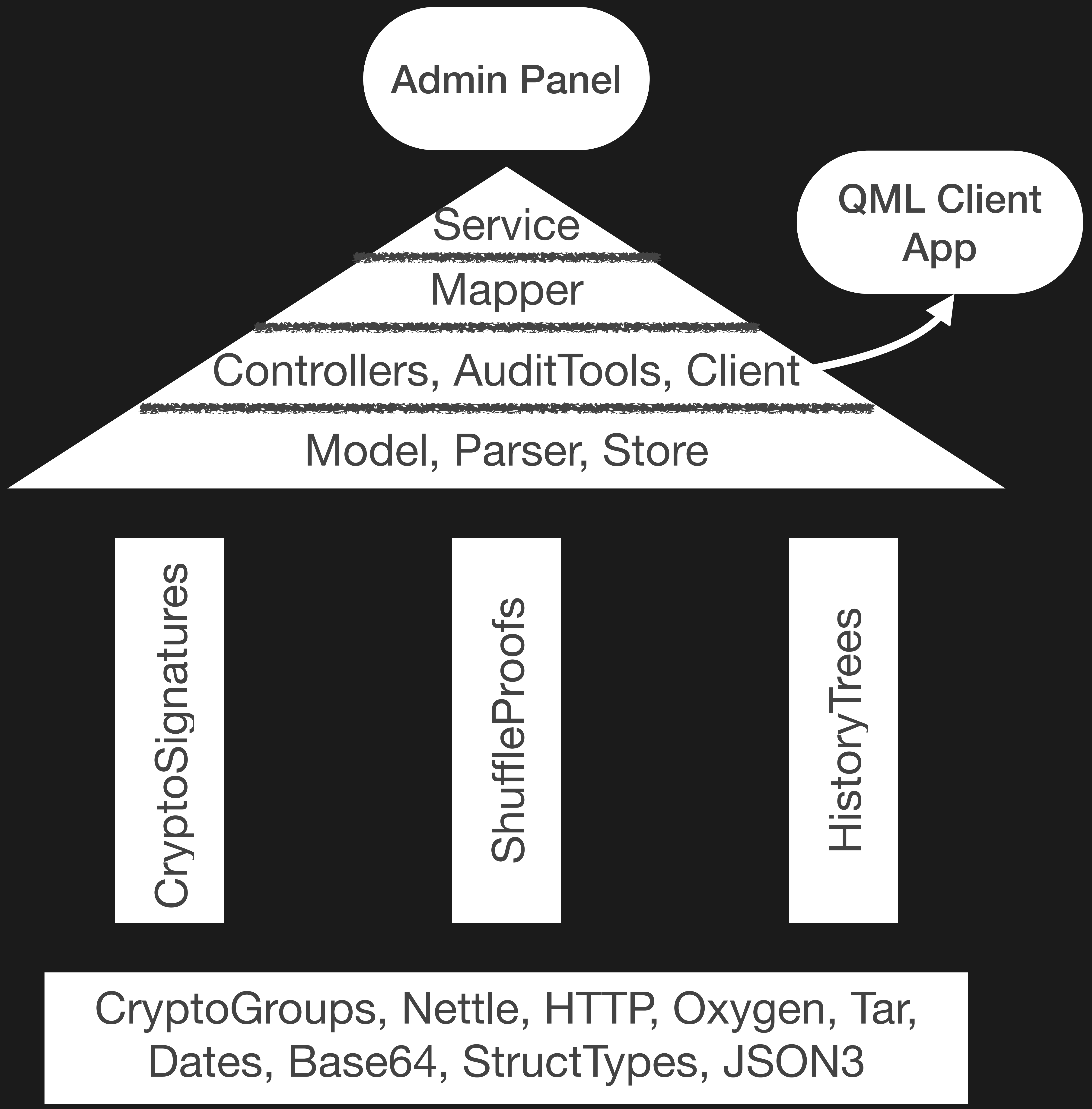
Stack Overview

Backend, Admin Panel and Client

- 15k lines of Julia code and 3k lines of QML, some ChatGPT-generated Javascript and CSS tricks
- The backend is built as a modular monolith
- The admin panel is layered on top of the backend and defines a separate service

Module	src	test
PeaceFounder	5574	867
CryptoGroups	1989	596
ShuffleProofs	1236	578
CryptoSignatures	153	122
HistoryTrees	300	234
PeaceFounderAdmin	1286	31
PeaceFounderClient	417	246

Generated via PackageAnalyzer



Why Julia

What have made it great for the project

- Memory safety and garbage collector allow to focus on the problem;
- Sensible and rich hierarchical type system with zero cost abstractions;
- Multiple dispatch. isbinding(x, y[, hasher]) has 37 methods!
- As a developer I can use dependencies with binaries without ever having to compile anything on my machine or know anything about the zoo of build systems;
- Manifest.toml reproduces the same environment among all platforms;
- Modules, macros and globals surpass class singletons with ubiquitous self;
- Interactive workflow with Revise, Infiltrator and root projects with local dev

Short Feedback Loops

Test Driven Development

- The PeaceFounder backend is built in a layered form, separating the Model, Mapper and Service layers and includes a separate Client backend;
- There is an integration test for each layer that allows one to spot errors at the lowest abstraction level and not any level deeper;
- The service layer is tested directly with the Router, which is passed to the client, allowing it to follow the whole stack trace;
- Time is mockable, so tests do not need to wait for events or fail because precompilation has taken too long.
- Revise and Infiltrator help tremendously with the global state.

Short Feedback Loops

How I use Revise with HTTP

HTTP Middleware

```
function ReviseHandler(handle)
    req -> begin
        Revise.revise()
        invokelatest(handle, req)
    end
end
```

The very next HTTP request reflects the new codebase, making debugging the Admin panel a breeze.

Short Feedback Loops

How I use Revise with QML

QML Middleware

```
function ReviseHandler(handle)
    return function(args...)
        Revise.revise()
        invokelatest(handle, args...)
    end
end
```

QML Middleware

```
function set_qmlfunction(f::Function; name::Symbol = nameof(f), middleware = [])

    handler(args...) = ErrorMiddleware(f; name)(args...)
    qmlfunction(string(name), reduce(|>, [handler, middleware...]))

    return
end

for func in [setDeme, setProposal, castBallot, refreshHome, refreshDeme,
             refreshProposal, resetBallot, addDeme]
    set_qmlfunction(func; middleware)
end
```

See the changes from the backend in the UI immediately without restarts.

Issues with Julia

Some nuances I wish were addressed

- Null safety. It is burdensome for the program to compile and run where, at some point, nothing could happen unchecked at runtime;
- Modular numbers Mods. It is natural to put modulus as a type parameter and write function signatures for its equality. But it shouldn't be compiled. Also BigInt;
- Reusing client backend code for mobile would be fantastic;
- Explicit mutability at the call site (Rust, Swift, Kotlin, Mojo);
- Ability to shadow getindex at a module level so some crypto code could be reimplemented from specs in verbatim.

The End