A voter-to-voter voting protocol

Stanislaw Baranski
stanislaw.baranski@pg.edu.pl
https://stan.bar

23.10.2023

Internet voting

Motivation

- Voting is one of the most popular mechanisms for collective decision-making; yet, it’s still something we can not do securely online.
- There are several ways of voting; But, internet voting is the most conventional, cheapest, fastest, and safest (e.g., during the outbreak of COVID-19), and hence, a preferred method for conducting voting.

Internet voting

Value

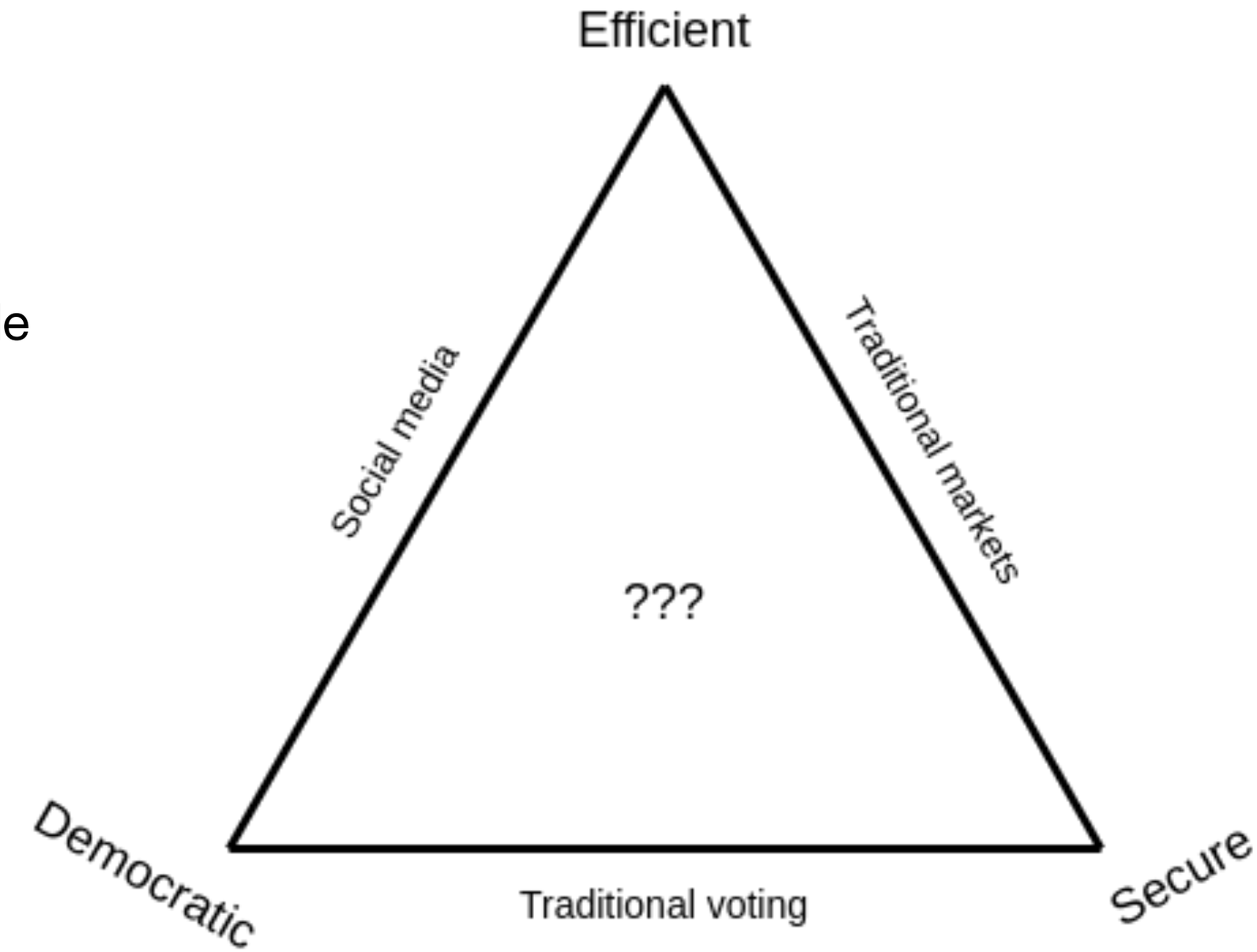
- **Convenience, and safety.** No need to leave your home to participate in voting.
- **Cheap.** No need to print ballot papers or hire people to coordinate the voting process.
- **Trustless, secure, transparent.** Users don't need to trust the authorities that their votes have been included and that the counting process has been correct.
- **Increased turnouts and the frequency of votings.**
- Catalyse the further **development of modern democracy**. Enabling practical applications of direct democracy, liquid democracy, and all other sorts of voting methods like Quadratic Voting, Approval voting, Alternative voting, Score voting, and many others.

Collective decision making

Tradeoffs

Trilemma. Choose only two out of three properties:

- **Democratic**, means that the method ensures easy and equal (egalitarian) decision input for all eligible voters;
- **Secure**, means that the voting is confident, fair, transparent, private, and resistant to attack vectors;
- **Efficient**, means that the method is easy, fast, and cheap.



Source: https://vitalik.ca/general/2021/05/25/voting2.html

Internet voting is hard

Analysis of this area quickly reveals several unsolved issues.

Secure voting requires four main properties:

- **Correctness**, all and only eligible votes are counted.
- **Censorship resistance**, any eligible user that wants to cast a vote can do it.
- **Privacy**, no one can tell which candidate the voters voted for, or even if they voted at all—preventing preliminary results and guaranteeing freedom of choice.
- **Coercion resistance**, voters can not prove to anyone how they voted even if they want to—preventing selling votes as there is no way of verifying if they indeed voted on the paid candidate.

Internet voting is hard

Analysis of this area quickly reveals several unsolved issues.

Secure voting requires four main properties:

- **Correctness**, all and only eligible votes are counted.
- **Censorship resistance**, any eligible user that wants to cast a vote can do it.
- **Privacy**, no one can tell which candidate the voters voted for, or even if they voted at all—preventing preliminary results and guaranteeing freedom of choice.
- **Coercion resistance**, voters can not prove to anyone how they voted even if they want to—preventing selling votes as there is no way of verifying if they indeed voted on the paid candidate.

They are hard to satisfy together.

Internet voting is hard

Analysis of this area quickly reveals several unsolved issues.

Secure voting requires four main properties:

- **Correctness**, all and only eligible votes are counted.
- **Censorship resistance**, any eligible user that wants to cast a vote can do it.
- **Privacy**, no one can tell which candidate the voters voted for, or even if they voted at all—preventing preliminary results and guaranteeing freedom of choice.
- **Coercion resistance**, voters can not prove to anyone how they voted even if they want to—preventing selling votes as there is no way of verifying if they indeed voted on the paid candidate.

They are hard to satisfy together.

And even if they are satisfied, there are more fundamental problems.

A lot of scepticism

- Shankland, S. (2018). **No, blockchain isn't the answer to our voting system woes**. https://www.cnet.com/news/privacy/blockchain-isnt-answer-to-voting-system-woes/
- Lee, T. B. (2018, November 6). **Blockchain-based elections would be a disaster for democracy**. Ars Technica. https://arstechnica.com/tech-policy/2018/11/blockchain-based-elections-would-be-a-disaster-for-democracy/
- Mearian, L. (2019, August 12). **Why blockchain-based voting could threaten democracy**. Computerworld. https://www.computerworld.com/article/3430697/why-blockchain-could-be-a-threat-to-democracy.html
- Jefferson, D. (2019). **The Myth of "Secure" Blockchain Voting. Verified Voting**. Available Online: Https://Verifiedvoting. Org/the-Myth-of-Secure-Blockchain-Voting/(Accessed on 12 October 2020).
- Schneier, B. (2019). **Blockchain and Trust**. Schneier on Security. https://www.schneier.com/blog/archives/2019/02/blockchain_and_.html
- Gaudry, P., & Golovnev, A. (2020). **Breaking the encryption scheme of the Moscow internet voting system**. 32–49.
- Specter, M. A., Koppel, J., & Weitzner, D. (2020). **The Ballot is Busted Before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in {US}**. Federal Elections. 1535–1553.

But

- Buterin, V. (2021). **Blockchain voting is overrated among uninformed people but underrated among informed people**. Vitalik Buterin's Website. https://vitalik.ca/general/2021/05/25/voting2.html
- Golomb, G. (2018, January 3). **Believe It: Cybersecurity is Getting Better, Not Worse.** Infosecurity Magazine. https://www.infosecurity-magazine.com/opinions/cybersecurity-getting-better-worse/
- https://www.electionguard.vote/ **Microsoft (Josh Benaloh) project**

A lot of scepticism

The resistance lies—among others—in insufficient confidence in the technology and a need for trust in the authorities controlling the voting process.

The criticism against internet voting comes down to two arguments:

1. **Device related**. No software is flawless, therefore it can not be trusted.

2. **Trust related**. There is too strong a trust assumption in authorities controlling the voting process.

1. Device related. No software is flawless, therefore it can not be trusted

High-quality software contains on average one defect in every ten thousand lines of code

Table 1: **Four categories of voting systems.** The top row (green) is *software-independent* and far less vulnerable to serious failure than the bottom row (red). The bottom row is highly vulnerable and thus unsuitable for use in political elections, as explained further in §2.

	In person	**Remote**
Voter-verifiable paper ballots[3]	*Precinct voting*	*Mail-in ballots*
Unverifiable or electronic ballots	*DRE*[4] *voting machines*	*Internet/mobile/blockchain voting*

Source: https://people.csail.mit.edu/rivest/pubs/PSNR20.pdf

1. Device related. No software is flawless, therefore it can not be trusted

High-quality software contains on average one defect in every ten thousand lines of code

Table 1: **Four categories of voting systems.** The top row (green) is *software-independent* and far less vulnerable to serious failure than the bottom row (red). The bottom row is highly vulnerable and thus unsuitable for use in political elections, as explained further in §2.

	In person	**Remote**
Voter-verifiable paper ballots[3]	*Precinct voting*	*Mail-in ballots*
Unverifiable or electronic ballots	*DRE*[4] *voting machines*	*Internet/mobile/blockchain voting*

- Recent advances in cryptography can **guarantee correct program execution** using zero-knowledge proofs [13].

- Generally, it is believed that **cybersecurity is getting better, not worst** [23].

- Moreover, the authors of [24] claim that ”**there is no perfect, infallible way to count votes**. All methods including optical scan, touchscreen, and hand counting—are subject to errors, procedural lapses, and deliberate manipulation.” Therefore, the argument is not about security or lack of it, but how much secure it is, and what are the trust assumptions.

2. Trust related. There is too strong a trust assumption in authorities controlling the voting process

Evidence-based election: Ideally, the whole voting process should be completely trustless, meaning that, there should be no trust assumptions other than in our perception.

- In practice, we rarely monitor the whole process of elections. Rather, we delegate that duty to staff responsible for conducting voting. We believe that at least one person is an honest observer who will alarm if something goes wrong.
- So the evidence-based election [24], in practice uses 1 of N trust model [26], which means that the system is trusted as long as at least one person out of N observers is honest, and in case of a fraud will reveal it.
- However, if at some point, the group of observers drops to a few people, the chance of finding at least one honest observer reduces, and with it the trustworthiness of the whole election. Therefore, a voting process should involve a large number of observers — the larger the N, the more trustworthy the setup is.
- The critique against internet voting system run by centralised authorities is that it requires the strongest assumption on 1 of 1 trust model as there is no way to provide the electorate convincing evidence that the running software is correct. This means that there is a single point of failure, an authority, which if compromised, breaks the trust.

My research

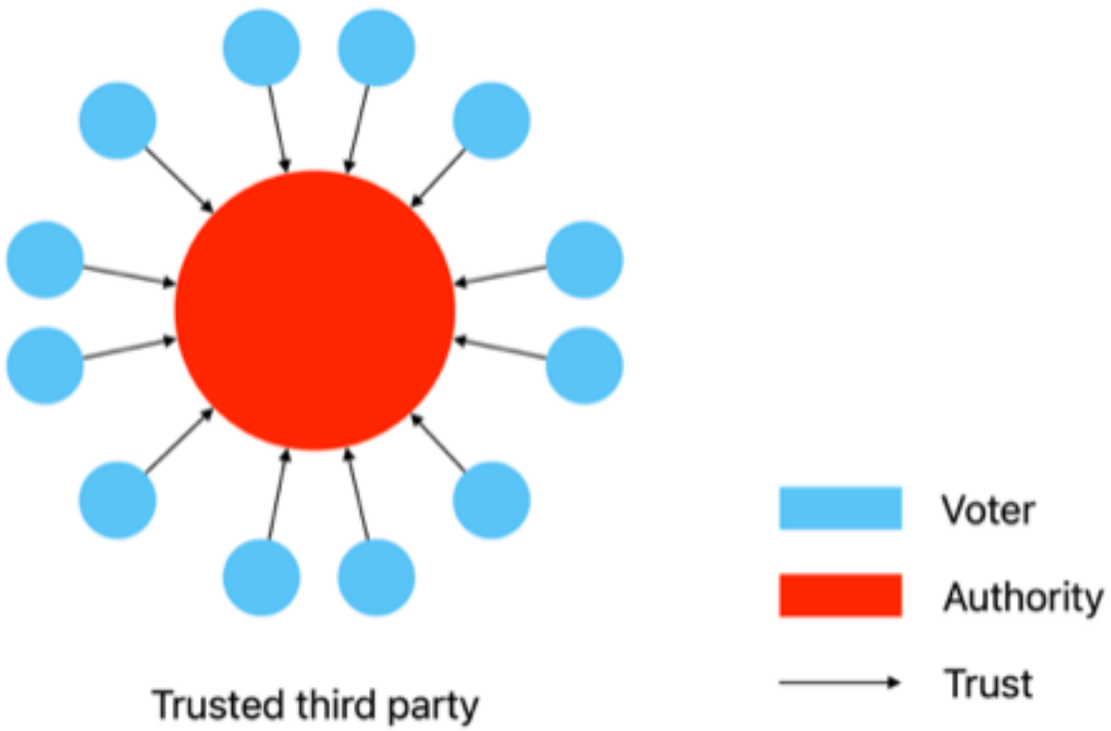
- I don’t want to focus on the device part.
- I want to move the trust part.

Internet voting

Naive solution

- Introduce a trusted authority, which authenticates and authorises voters, collects votes, counts them and publishes a result. Easy!

- Most of the internet protocols rely on a trusted third party. They differ in what the server can or cannot do. The honesty of the trusted third party determines either anonymity, privacy, or coercion resistance properties.



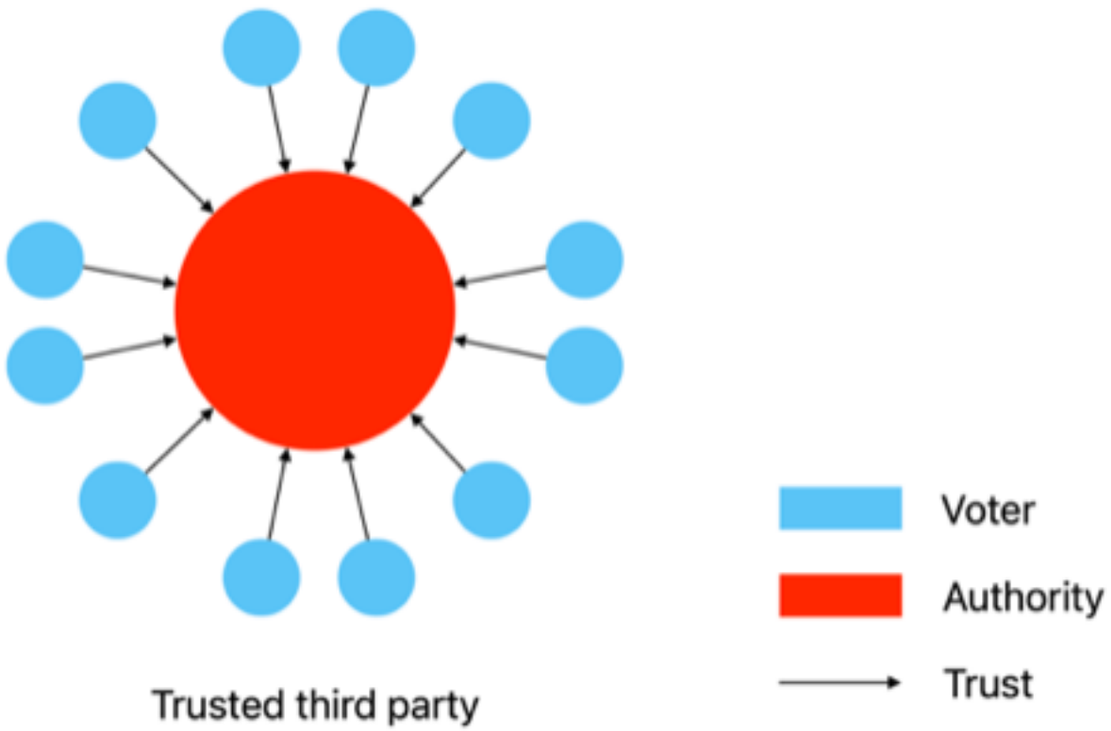
Internet voting

Naive solution

- Introduce a trusted authority, which authenticates and authorises voters, collects votes, counts them and publishes a result. Easy!

- Most of the internet protocols rely on a trusted third party. They differ in what the server can or cannot do. The honesty of the trusted third party determines either anonymity, privacy, or coercion resistance properties.

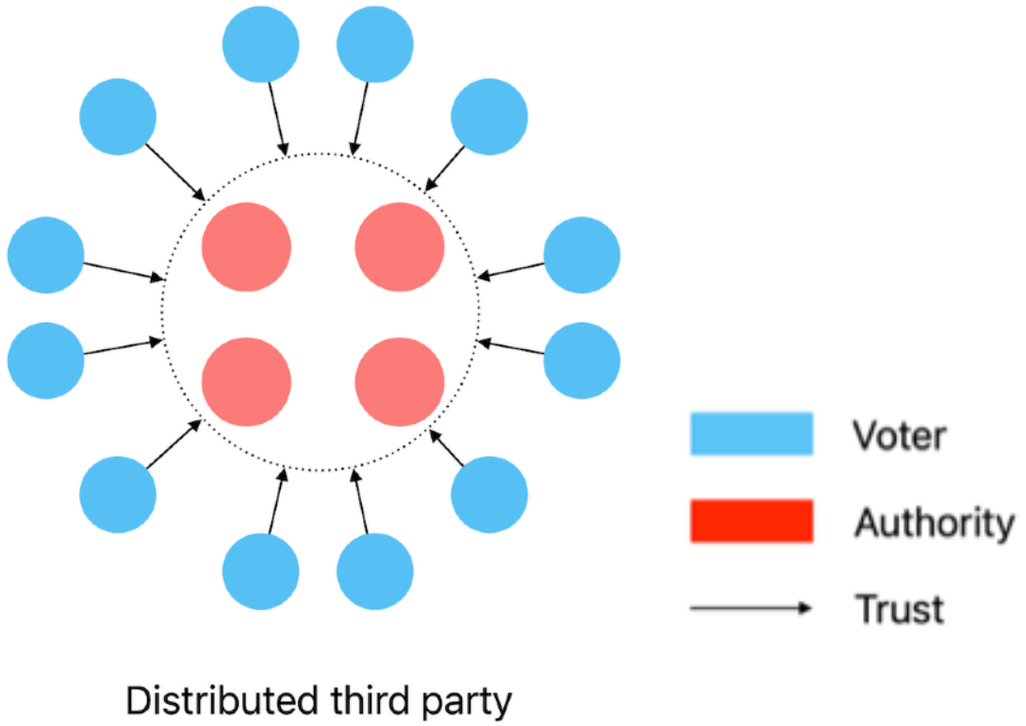
We don’t want the trusted authority, it has too much power!



Internet voting

Trusted institutions

- Decentralise the trusted authority with a set of distrusting parties.
    - Parties can be trusted institutions like universities, banks, candidates, government organisations, and NGOs.



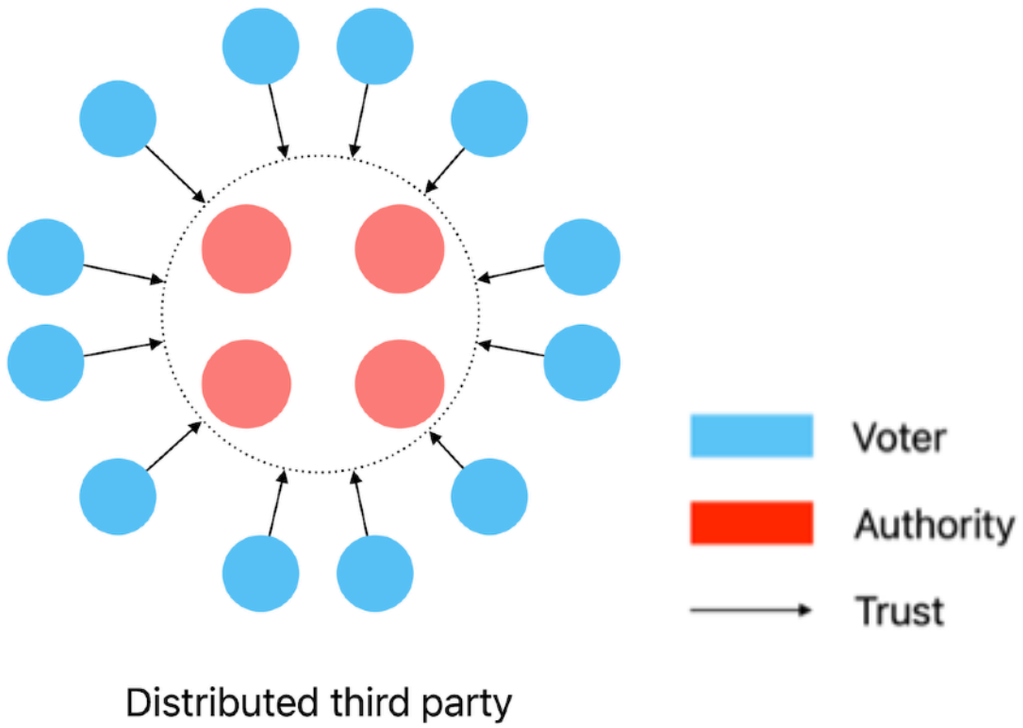
Distributed third party

Internet voting

Trusted institutions

- Decentralise the trusted authority with a set of distrusting parties.
    - Parties can be trusted institutions like universities, banks, candidates, government organisations, and NGOs.

How to achieve Fairness? Integrity? Transparency? Verifiability? Censorship resistance?

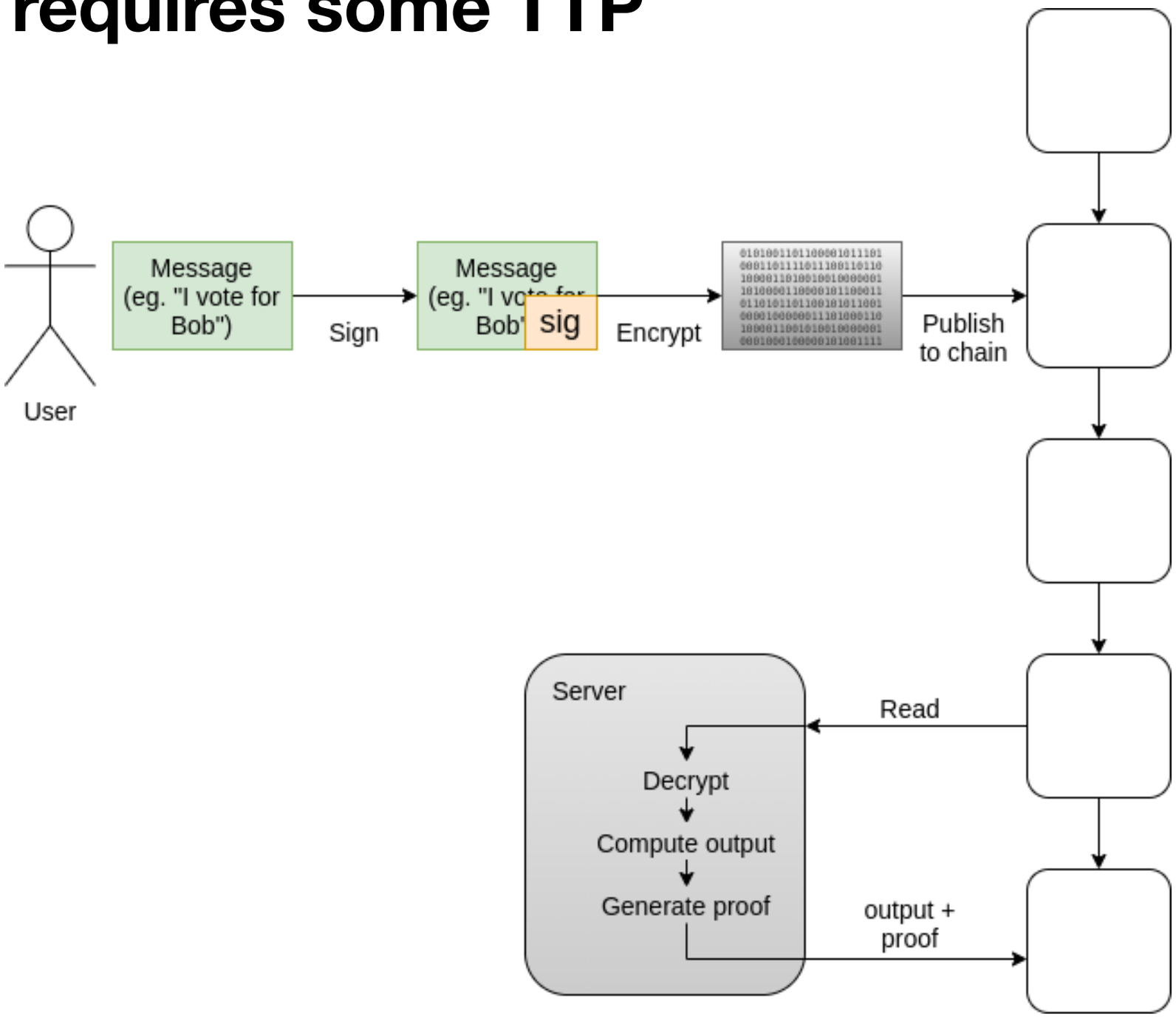


Blockchain

The solution to all problems

Blockchain and Central Server

Usually requires some TTP



Source: https://vitalik.ca/general/2021/05/25/voting2.html

Internet voting

Problems

- Most Internet protocols rely on a **trusted third party**. They differ in what the server can or cannot do. The honesty of the trusted third party determines either censorship-resistance, anonymity, privacy or coercion-resistance properties.

- Some of them use **blockchain for integral and transparent storage** (Voatz, Polys, MACI).

- Some solutions use **MPC to distribute the trusted third party** (Civitas, Swisspost/Scytl, iVoting).

- Moreover, solutions built on **public blockchains** require **voters to pay transaction fees**.

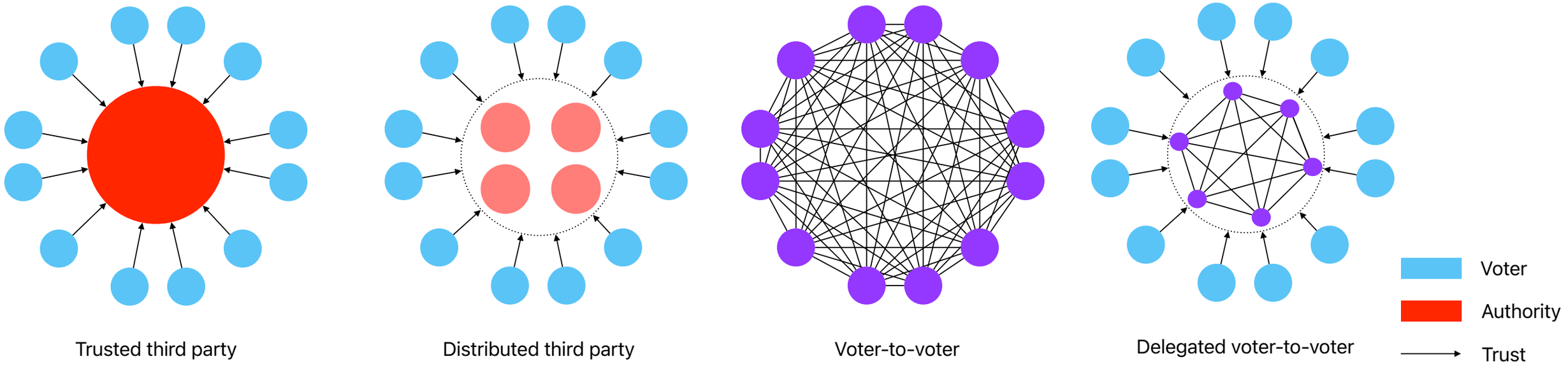
- Solutions built on **private blockchains** still need to be hosted somewhere, which **costs money** or creates a **high entry point** for non-technical people.

My contribution

Voter-to-voter trust model

- Most people think about voting in terms of presidential elections. However, voting is used also in small, local votings like housing associations, board members, contests, hackathons, and all forms of committees.

- We want to go even further and conduct the voting on voters' end devices (PC, laptops, or even smartphones), moving the trust down to the voters and removing the operational fees.



Aim

- We **don’t** want to build
  - A large-scale voting system for presidential elections;
  - A voting system for crypto space only;
  - A perfectly secure, coercion-resistant protocol.
- Rather, we **do** want to build
  - A voting protocol for small to medium size voting like 1000 voters;
  - Applications: Student Council Elections, Corporate Board Decisions, Homeowners Association Voting, NGOs, Town Hall Meetings, Union Voting, Talent Competitions, Student communities, or Boardroom.
  - A voting protocol for people, without crypto background;
  - A decentralised, privacy-preserving, secure protocol that works.
  - App that is easy and free to use

Internet voting

Problems

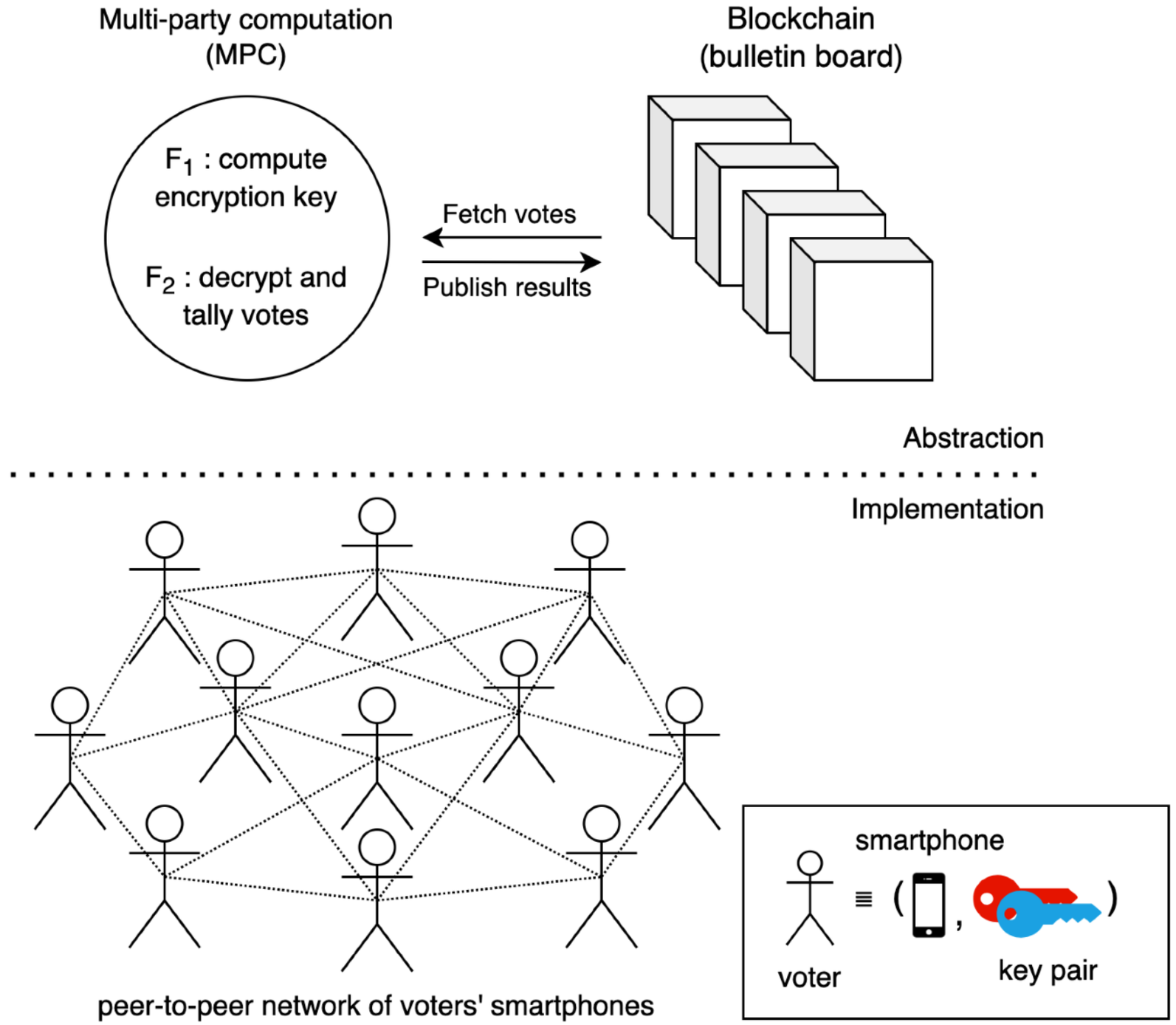
Property	Centralised server	Private network	Public blockchain	Voter-to-Voter network
Transaction fees	No	No	Yes	No
Service costs[1]	Medium	High	No	No
User-Friendliness	High	High	Low	Medium
Trust to[2]	Central authority	Authorities[3]	Miners	Voters

Objectives

- Private, based on honest-majority assumption
- Distributed, there is no central authority that guarantees any security property
- Convenience, three optional rounds
  - 1. Optional DKG
  - 2. Optional voting
  - 3. Optional tally
- Lightweight both in terms of memory and time so it can be executed on voters’ laptops or (ideally) smartphones.
- Free, zero-fee participation using peer-to-peer ad-hoc network or ERC4337 paymaster

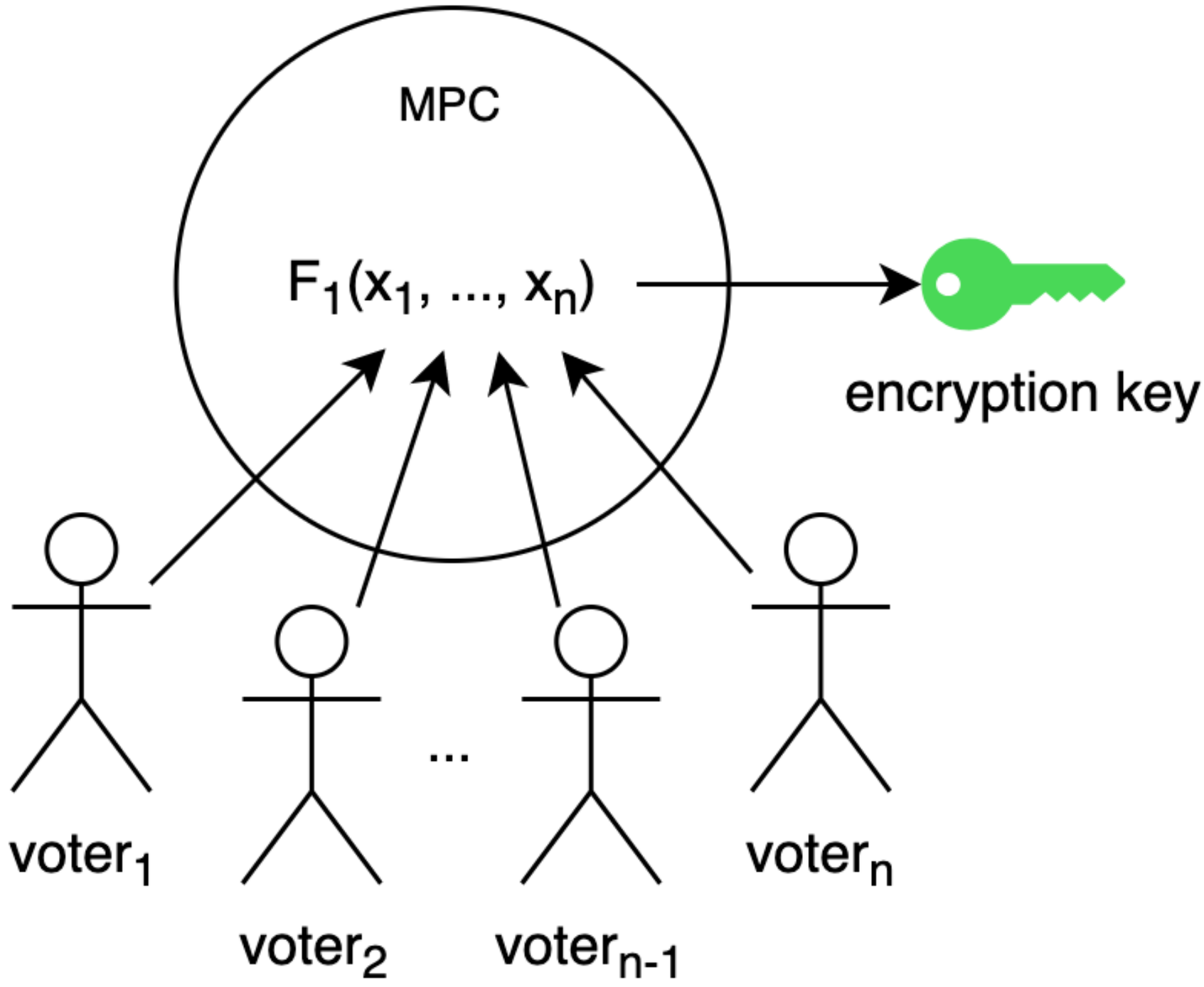
Technical Vision

Architecture



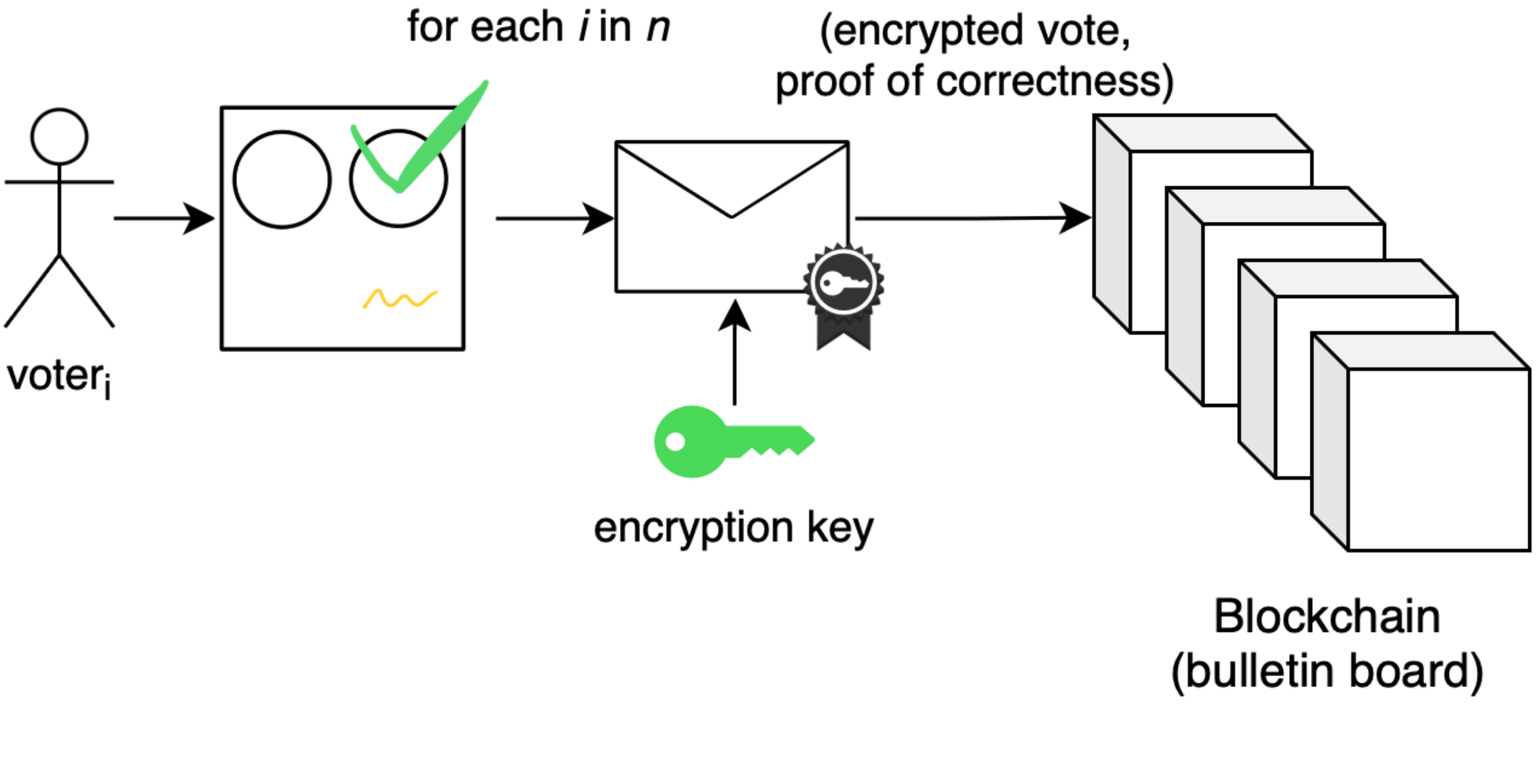
Technical Vision

Setup



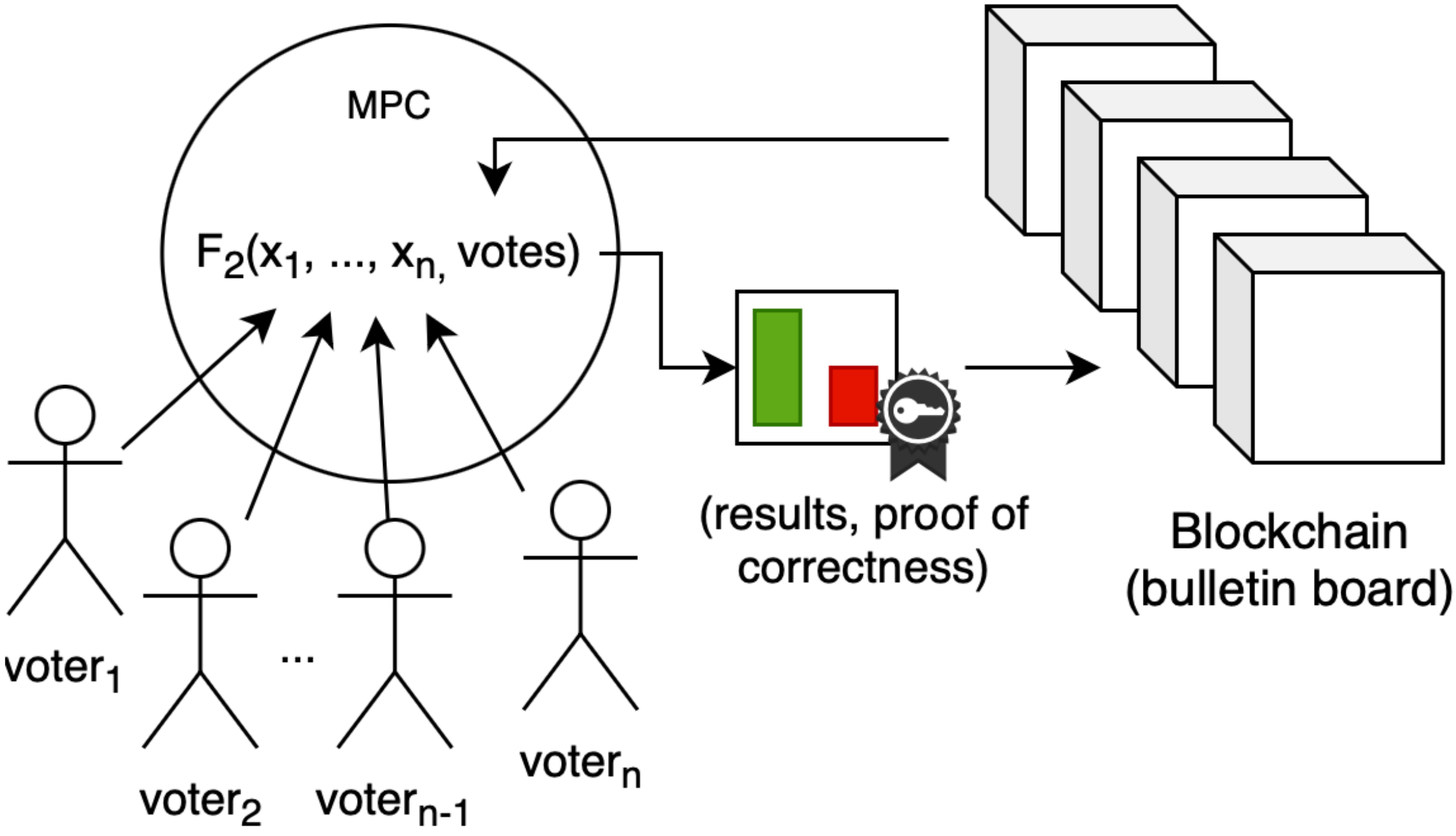
Technical Vision

Voting

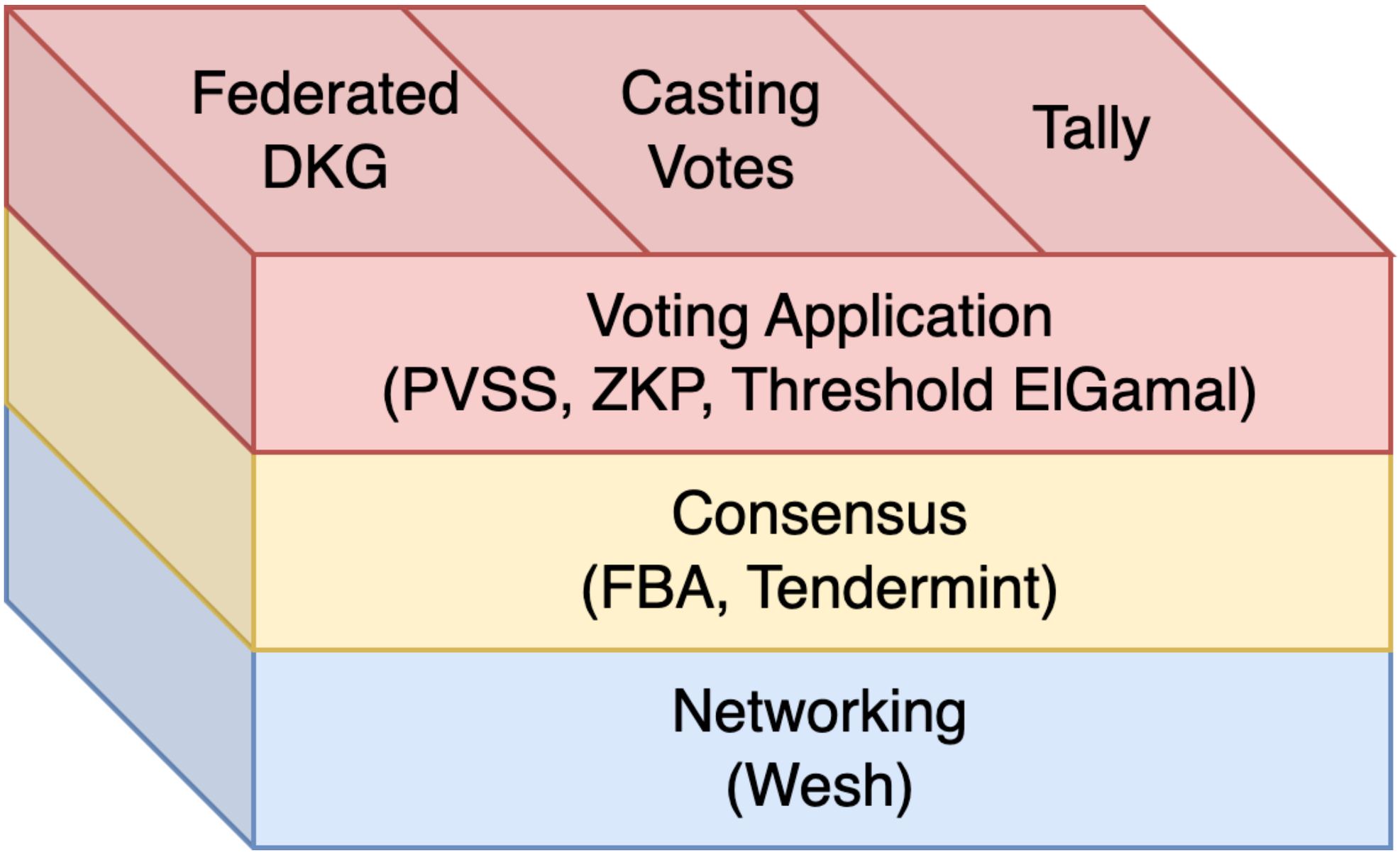


Technical Vision

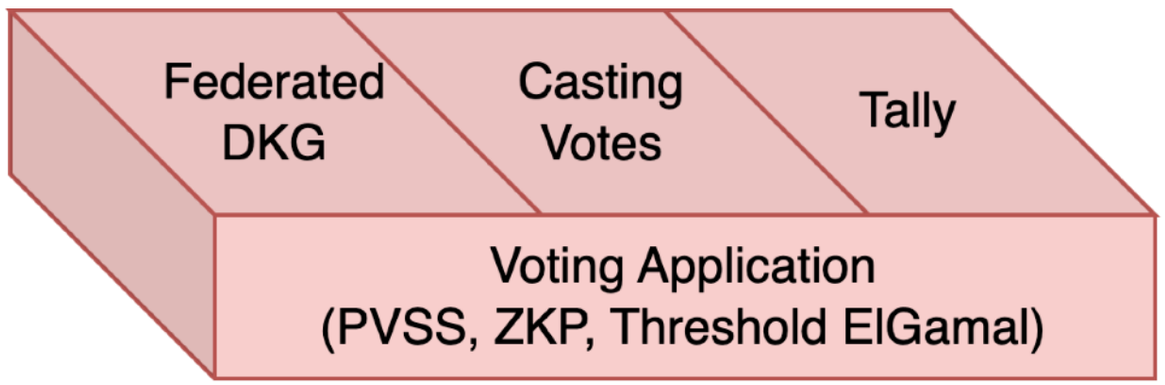
Tally

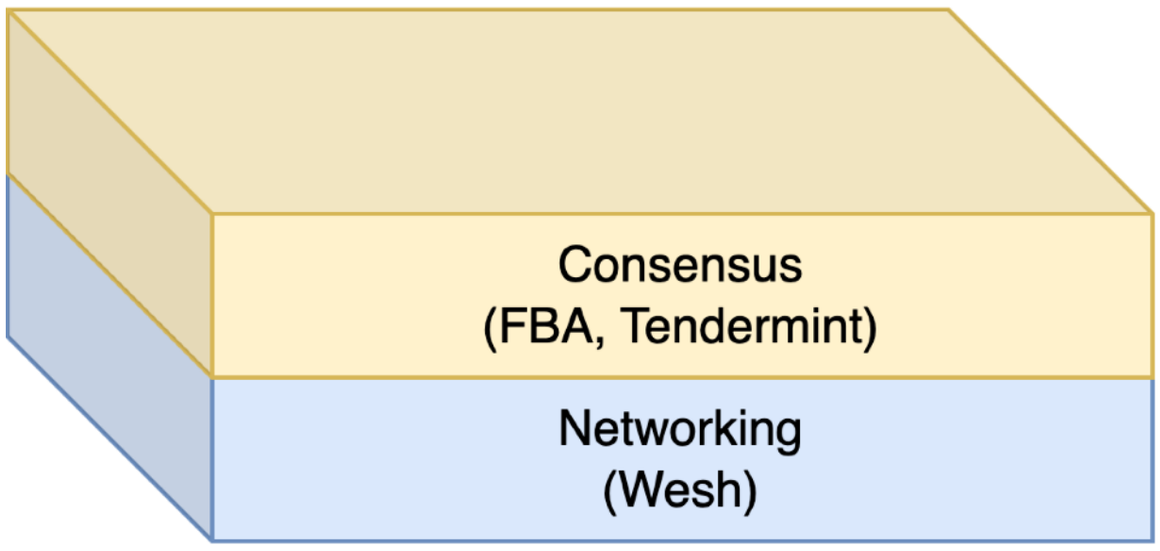


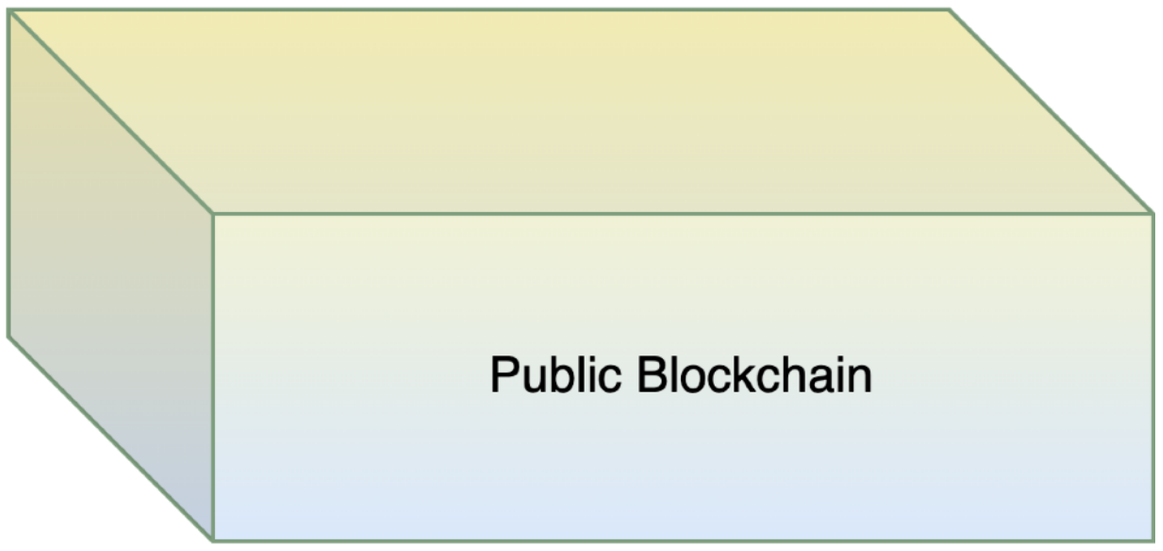
Architecture



Architecture





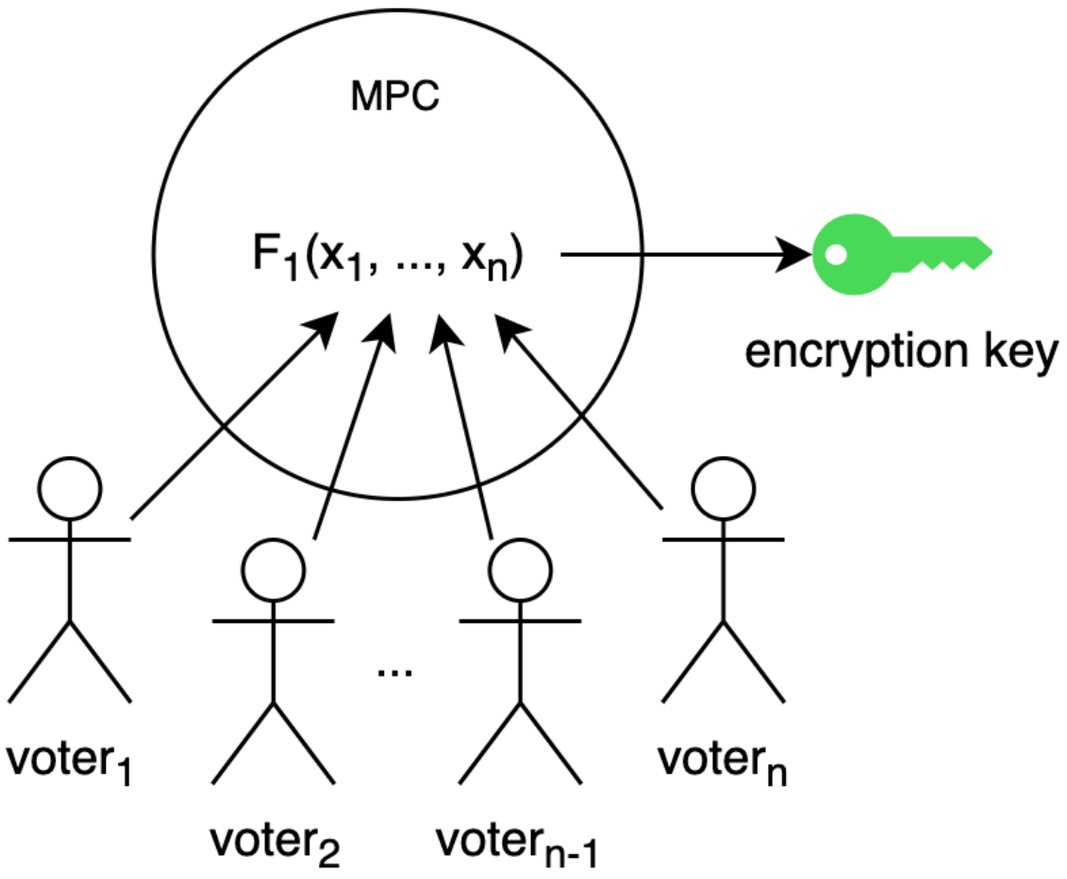


Voting Application

Assumptions

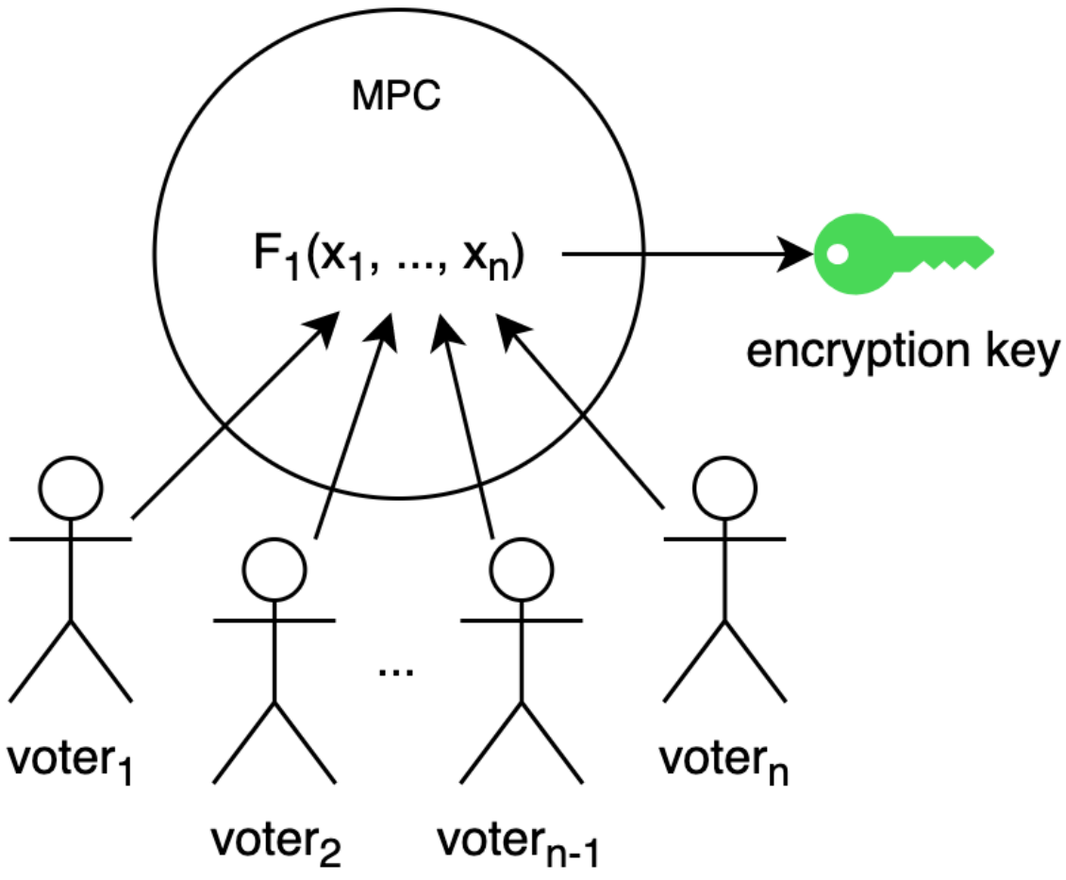
- The set of all n participants $\mathbb{P} = \{P_1, \ldots, P_n\}$ is publicly known.
- Each participant P_i consists of key pair (sk_i, P_i), where $sk_i \in_R \mathbb{Z}_q$ is a randomly selected secret key and $P_i = sk_i \times G$ the corresponding public key. We use the same notation for a party and its public key, P_i, as parties are identified by their public keys only.

Distributed Key Generation



- We don’t want any party to see the secret decryption key.
- We want parties to jointly compute key-pair.
  - Secret $d = x_1 + x_2 + \ldots + x_n$
  - Public $E = dG$
- S.t. no one learns $d, x_1, x_2, \ldots, x_n$

Distributed Key Generation



- Each party pick a random polynomial $f_i(X) \in \mathbb{Z}_q[X]$, and then defines the final polynomial $\mathbf{f}(X) = \sum_{i=1}^{n} f_i(X)$;
- $\mathbf{d} = \mathbf{f}(0)$,
- $\mathbf{E} = \mathbf{d} \times G$.
- To prevent misbehaviour of parties (sending arbitrary values) we use a more sophisticated version of SS called Publicly Verifiable Secret Sharing (PVSS) which involves zero-knowledge proofs attesting that the correct relation between values holds.

Dynamic Distributed Key Generation

Problem

- **We want the DKG and Tally rounds to be optional**
- Threshold Encryption works on Shamir Secret Sharing (SSS)
- SSS works on polynomials
- Polynomials have to have a defined fixed degree t
- We want the DKG phase to be optional, so the total number of participants is unknown, and so the t is also unknown
- Therefore, **we can not define the polynomial of unknown size**

Dynamic Distributed Key Generation

Solution

1. Dynamic DKG requires all parties to be online for the duration of the DKG (possibly a few hours). It's done by reconstructing the key by current participants and resharing it again with the new participant.

    1. Unpractical assumption. We want the protocol to be non-interactive, meaning that the party sends only one message and then can leave (YOSO)

2. Introduce the registration round where parties can signal their interest in participating in DKG so that the number of participants is known

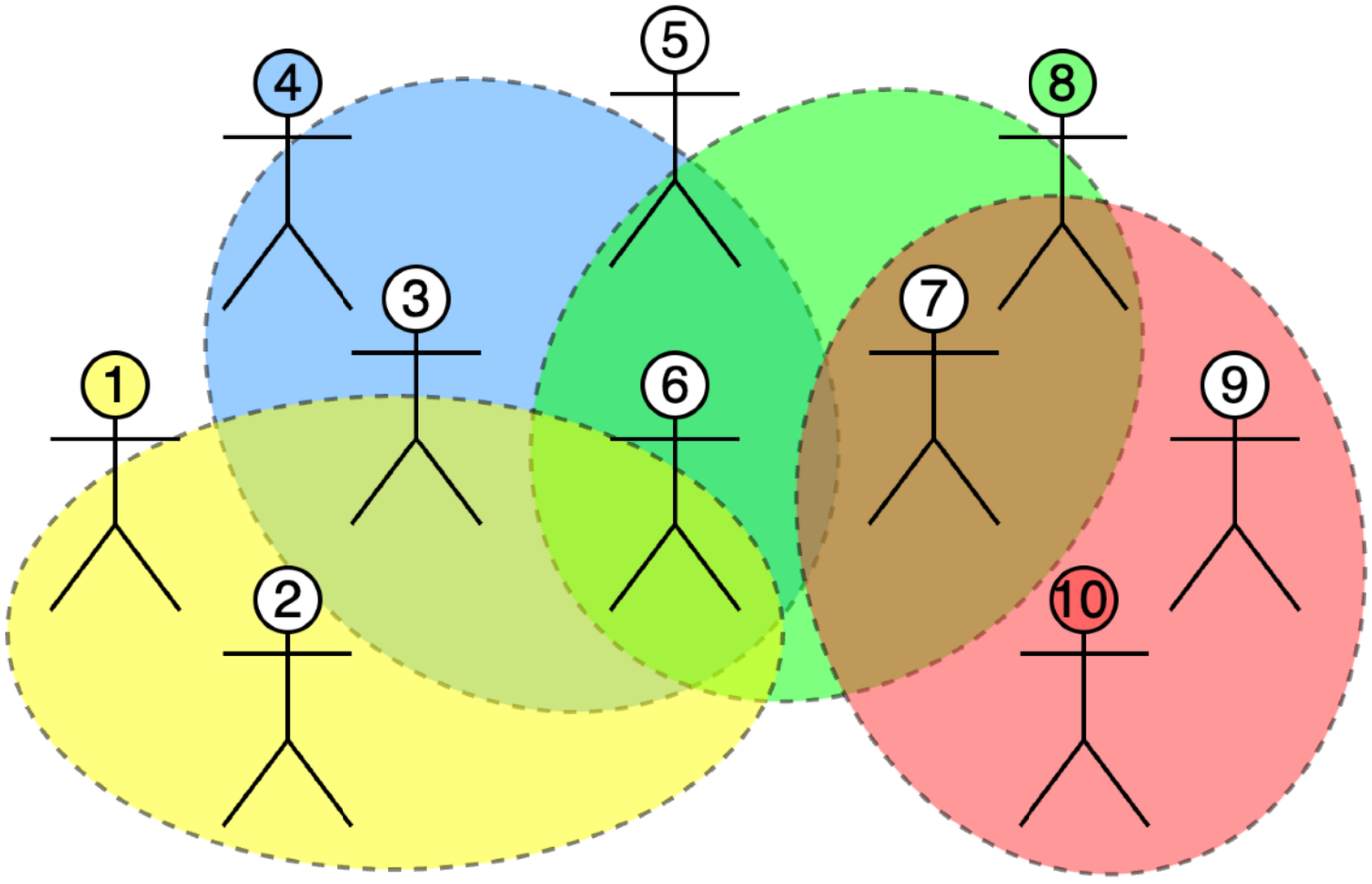
    - We don’t want to introduce another round

    - How can we know that registered members will actually participate in DKG round?

3. Virtual registration, based on social assumptions

    1. A novel technique for dynamic DKG that works similarly to the Federated Byzantine Agreement (Stellar Consensus Protocol).

Federated DKG



DKG is a set $\mathbb{D}$ = { 1 4 8 10 }

1 's Guardian Set $\mathbb{G}_1$ is { 2 3 6 }

4 's Guardian Set $\mathbb{G}_4$ is { 3 5 6 }

8 's Guardian Set $\mathbb{G}_8$ is { 5 6 7 }

10 's Guardian Set $\mathbb{G}_{10}$ is { 7 8 9 }

- $P_1 : \mathsf{SSS}(\mathbb{G}_1, x_1) \rightarrow \{s_2, s_3, s_6\}$
- $P_4 : \mathsf{SSS}(\mathbb{G}_4, x_4) \rightarrow \{s_3, s_5, s_6\}$
- $P_8 : \mathsf{SSS}(\mathbb{G}_8, x_8) \rightarrow \{s_5, s_6, s_7\}$
- $P_{10} : \mathsf{SSS}(\mathbb{G}_{10}, x_{10}) \rightarrow \{s_7, s_8, s_9\}$

- Later, during Tally round, d is recoverable if at least 2-of-3 secret shares of each $\{\mathbb{G}_1, \mathbb{G}_4, \mathbb{G}_8, \mathbb{G}_{10}\}$ is published.
- Definition(Decipherability). A _ enjoys Decipherability iff at least 2-of-3 of each Guadrian Set publishes its partial decryption.

Voting Application

Federated Distributed Key Generation

- For each party $P_i \in \mathbb{D}$, where $\mathbb{D} \subseteq \mathbb{P}$ is a subset of parties participating in DKG:
  - Chose a guardian set of k parties $\mathbb{G}_i = \{P_{i_1}, P_{i_2}, \ldots, P_{i_k}\} \subseteq \mathbb{P}/P_i$
  - Sample a random polynomial $f_i(X) \in_R \mathbb{Z}_q[X]$ of degree $t-1$
  - Compute decryption (secret) key $d_i = f_i(0)$ and encryption (public) key $E_i = d_i \times G$
  - Create a t-of-k access structure for d_i using Publicly Verifiable Secret Sharing (PVSS)
    - For each guardian $P_{i,j} \in \mathbb{G}_i$, $1 \leq j \leq k$, create a secret share $f_i(j)$, encrypt it $C_{i,j} = \text{Enc}_{P_j}(f_i(j))$ and create a zero-knowledge proof π = "I know f_i s.t. given j, P_j, and $C_{i,j}$, the $C_{i,j}$ is an encrypted value of a polynomial $f_i(\cdot)$ applied to j"
  - Broadcast $(E_i, C_{i,j}, \pi)$

Voting Application

Private channels

Encryption$_P(s)$

- $k \leftarrow_{\$} \mathbb{Z}$
- $r \leftarrow_{\$} \mathbb{Z}$
- $C1 = k \cdot G$
- $M = G \cdot r$
- $C2 = P \cdot k + M$
- $\Delta = s - M\,.\,x$
- return $(C1, C2, \Delta)$

Decryption$_{sk}(C_1, C_2, \Delta)$

- $M = C2 - sk \cdot C1$
- $s = M\,.\,x - \Delta$

Voting Application

Voting

- For each voter $P_i \in \mathbb{V}$, where $\mathbb{V} \subseteq \mathbb{P}$ is a subset of parties participating in voting:
  - Select a vote $v_{ij} \in \{0,1\} \simeq \{$"no", "yes"$\}$for each candidate $j \in \{1 \ldots l\}$.
  - Compute a ballot using ElGamal encryption for $B_i = (r_i G,\ r_i \mathbf{E} + v_{i1} H_1 + \ldots + v_{il} H_l)$, where
    - $r_i \in_R \mathbb{Z}_q$ is a blinding factor for user i, and
    - $H_1, \ldots, H_l$ are independent generators (one for each candidate).
- Compute a zero-knowledge proof π = “I know $r_i, v_{i1}, \ldots, v_{ij}$ s.t. $\sum_{j=1 \ldots l} v_{ij} = 1$ and $\forall_{j=1}^{l} v_{ij} = 1 \vee v_{ij} = 0$ and B_i is a correctly encrypted ballot using those values”
- Broadcast (B_i, π).

Voting Application

Online Tally

- For each party $P_i \in \mathbb{T}$, where $\mathbb{T} \subseteq \mathbb{D}$ is a subset of parties participating in the Threshold ElGamal Decryption. $\mathbb{T}$ must include at least $t \leq k$ parties from every set of guardians $\mathbb{G}_1, \ldots, \mathbb{G}_{\mathbb{D}}$.
  - Sum the first part of the ballots $A = \sum_{i \in 1 \ldots \mathbb{V}} C1_i = \sum_{i \in 1 \ldots \mathbb{V}} r_i \times G$, where $(C1_i, C2_i) = B_i$
  - Calculate the share of the decryption key $d_i = \sum_{f_j(i) \in S_i} f_j(i)\lambda_{j,i}$, where S_i is a set of shares $f_j(i)$, where P_i is in P_j's guardian set $\mathbb{G}_j$, and $\lambda_{j,i} = \prod_{k \in \mathbb{G}_j \setminus \{i\}} \frac{k}{k-i}$ denotes Lagrange coefficients.
- Calculate partial decryption $A_i = A \cdot d_i$.
- Compute a zero-knowledge proof π = "I know all shares $f_j(i)$ from encrypted shares $C_{j,i} = \text{Enc}_{P_i}(f_j(i))$ s.t. $A_i = A \cdot d_i$"
- Broadcast partial decryption (A_i, π)

Voting Application

Offline Tally

- Anyone can calculate the voting results:
  - Sum the first part of the ballots

    $C2 = \sum_{i \in 1 \ldots \mathbb{V}} C2_i = \sum_{i \in 1 \ldots \mathbb{V}} r_i \mathbf{E} + v_{i1} H_1 + \ldots + v_{il} H_l$, where

    $(C1_i, C2_i) = B_i$

  - Sum the partial descriptions

    $Z = \sum_{i \in 1 \ldots \mathbb{V}} A_i = \sum_{i \in 1 \ldots \mathbb{V}} A \sum_{f_j(i) \in S_i} f_j(i) \lambda_{j,i} = A \cdot d$

  - The decryption is $M = C2 - Z = x_1 H_1 + \ldots + x_l H_l$

Voting Application

Offline Tally

$M = C2 - Z = x_1 H_1 + \ldots + x_l H_l$ because

$$
\begin{aligned}
M &= C2 - Z \\
&= \sum_{i=1}^{k} (r_i \times \mathbf{E}) + H_1 \times \sum_{i=1}^{k} v_{i1} + \ldots + H_l \times \sum_{i=1}^{k} v_{il} - Z \\
&= \sum_{i=1}^{k} (r_i \times \mathbf{E}) + H_1 \times \sum_{i=1}^{k} v_{i1} + \ldots + H_l \times \sum_{i=1}^{k} v_{il} - \sum_{i=1}^{k} r_i \times \mathbf{E} \\
&= H_1 \times \sum_{i=1}^{k} v_{i1} + \ldots + H_l \times \sum_{i=1}^{k} v_{il} \\
&= x_1 H_1 + \ldots + x_l H_l
\end{aligned}
$$

To extract $x_c$we have to solve Elliptic-Curve Discrete Logarithm Problem. However, because x_c is a small number $0 \leq x_c \leq \quad \mathbb{V}$ it is feasible. To extract each x_i we use the technique described in Anonymous voting by two-round public discussion

Voting Application

zkSNARKs

Prover times

Groth16				
PVSS			Encrypt ballot	Partial decryption
1 of 2	2 of 3	3 of 4		
1.388s	2.135s	2.414s	0.747s	0.506s/share

PLONK				
PVSS			Encrypt ballot	Partial decryption
1 of 2	2 of 3	3 of 4		
67.753s	71.902s	146.026s	16.822s	8.602s/share

Voting Application

zkSNARKs

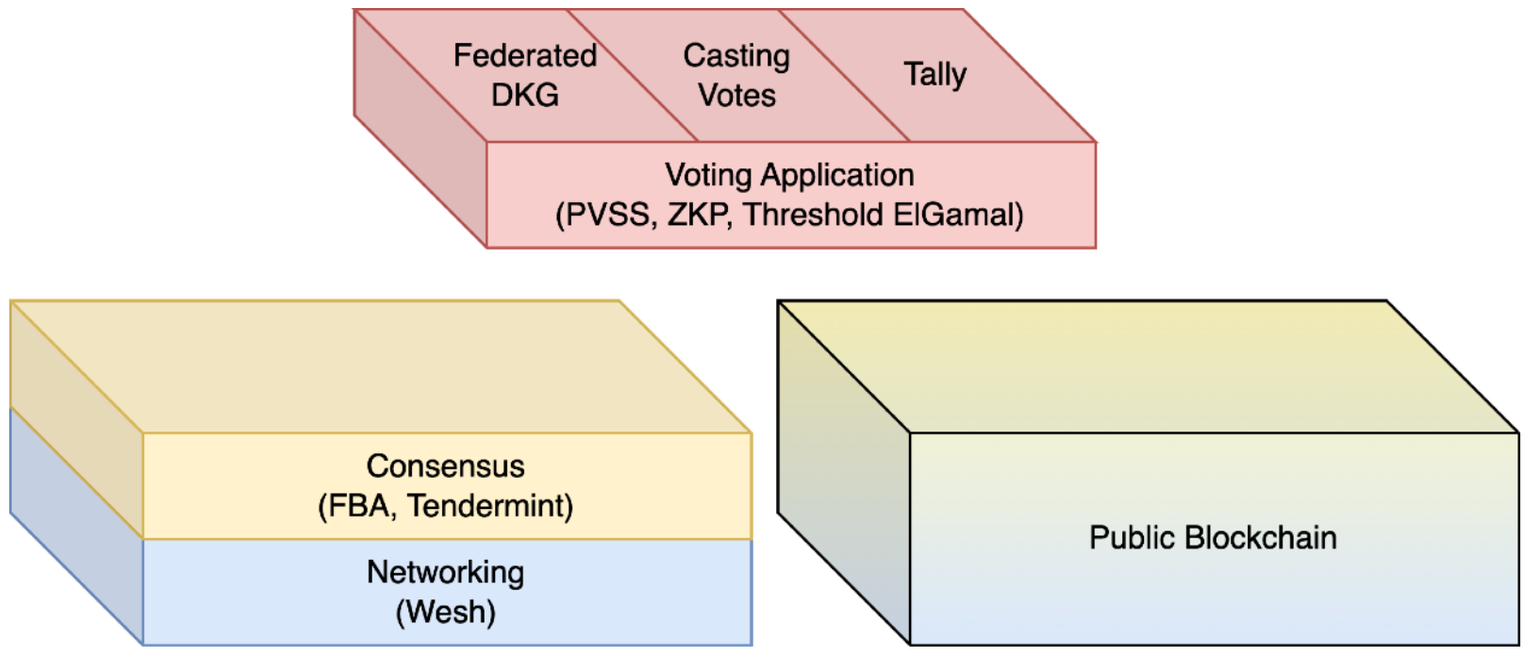
Message sizes

Groth16				
PVSS			Encrypt ballot	Partial decryption
1 of 2	2 of 3	3 of 4		
6476 KB	8928 KB	11412 KB	2.310 KB	2.548 KB/share

PLONK				
PVSS			Encrypt ballot	Partial decryption
1 of 2	2 of 3	3 of 4		
7.825 KB	10.332 KB	12.817 KB	3.671 KB	3.920 KB/share

Consensus and Networking

- Ad-hoc blockchain network
  - p2p networking via async mesh network https://wesh.network/
  - FBA or Tendermint
- Or, a public blockchain with ERC4337 paymaster to cover the transaction costs



Future work and open questions

- Finish p2p implementation.
- Create blockchain-based implementation on ETHIstanbul.
- Where and how to do an experiment?
- Publish paper.
- How to build MACI on top of this architecture?

Questions?

Stanislaw Baranski

stanislaw.baranski@pg.edu.pl

https://stan.bar

23.10.2023