Deploying Blockchain-Based Voting: Lessons from a Nationwide Referendum

Felix Maduakor [1], Jurlind Budurushi [2], and Joerg Mitzlaff [1]

Abstract: opn.vote, a blockchain-backed voting system using blind-signature credentials and a public vote record on Gnosis Chain, was deployed for the first time in ABSTIMMUNG21, a self-organised, nationwide, non-binding referendum, on 13 October 2025 in Germany. Of the 63,064 total votes cast, 12,922 were cast through opn.vote by voters with no previous blockchain experience, without exposing them to wallets, gas fees, or blockchain interactions.

In this article, we report on the use of opn.vote in the ABSTIMMUNG21 referendum. We outline usability, infrastructure, operational, and trust-boundary challenges and derive lessons learned for future deployments. We also report the complete participation funnel from contacted users to recorded votes, the timing and cost of on-chain transactions, and the 1,726 support requests received during the registration and voting periods.

A central finding is that the most consequential practical difficulties did not arise from the cryptographic protocol but at its boundaries with voters, election operations, and blockchain infrastructure, such as credential custody, ballot semantics, relayer funding, and user-support overhead. These findings inform the design of the next iteration of opn.vote for ABSTIMMUNG21 2026.

Keywords: Blockchain voting, Blind signatures, Gasless transactions, Internet Voting

1 Introduction

Germany has no provision for direct-democratic referendums at the federal level. Although Article 20[3] of the German Basic Law states that state authority is exercised by the people through elections and *referendums*, there is no enabling legislation for national referendums. ABSTIMMUNG21[4] is a coalition initiative of civil-society organisations – among them Democracy International, Mehr Demokratie, and Omnibus für Direkte Demokratie – that has established nationwide self-organised referendums since 2021 to demonstrate that such votes are practically feasible. Referendums held by ABSTIMMUNG21 are not legally binding, but are held to a self-imposed standard that is intended to provide the same level of security as postal voting in a federal election[5].

[1] openPetition gGmbH, Berlin, Germany,
maduakor.research@gmail.com, https://orcid.org/0009-0005-9681-1143;
joerg@openpetition.net, https://orcid.org/0009-0003-3177-5503
[2] Baden-Württemberg Cooperative State University Karlsruhe,
jurlind.budurushi@dhbw-karlsruhe.de, https://orcid.org/0000-0002-6732-4400
[3] https://www.gesetze-im-internet.de/englisch_gg/englisch_gg.html#p0112
[4] https://abstimmung21.de/
[5] https://abstimmung21.de/faq/

The initiative has held referendums alongside federal elections since 2021. A 2020 trial postal vote attracted around 46,000 participants across eight topics[6]. In the first full-scale ABSTIMMUNG21 referendum in 2021, approximately 344,556 people registered and about 160,000 voted by post on four topics, with every ballot counted by hand. These numbers demonstrated a strong public interest in direct democracy. However, the postal voting process was costly: printing, mailing, and manually counting the paper ballots cost roughly two euros per ballot, and counting required approximately 1,000 working hours from hundreds of volunteers.

The 2025 referendum marked a turning point. For the first time, participants could choose between the established postal voting channel and a new online voting channel based on opn.vote [MDM25]. Participants could cast their vote on four topics: 1) changes to the German Climate Protection Act, 2) a constitutional amendment concerning the debt brake, 3) a wealth tax on large fortunes, and 4) a halt to weapons deliveries to Ukraine in the event of a ceasefire. Among the 63,064 participants who voted, 50,142 cast their vote through postal voting, and 12,922 through opn.vote.

Introducing an online voting channel served three goals: 1) reducing the logistical and financial burden of postal voting, 2) lowering the barrier to participation, and 3) making democratic participation more direct and contemporary. Furthermore, the deployment had to show that a civil-society organisation can operate an online voting system with security properties comparable to those required in public elections, such as local or federal elections. Hence, the online channel had to ensure vote secrecy, vote integrity and provide a transparent and publicly auditable record of the result, while enabling a usable voting experience for people with no technical background. These requirements motivated the use of opn.vote, which shifts trust from the operator to a public blockchain.

Contributions

We report the first deployment of opn.vote [MDM25], in a nationwide real-world referendum with 12,922 online votes cast by people without prior blockchain experience and without the need to handle blockchain-specific tasks, such as managing wallets or paying transaction fees. Our main contribution is field evidence about practical challenges in such a system.

In addition, we report quantitative deployment data – the participation funnel from registration through credential issuance to vote submission, the timing of voter activity, transaction costs, and support-ticket volumes – as an empirical basis for judging the operational readiness of blockchain-backed voting at medium scale.

The deployment of opn.vote revealed that the most consequential challenges occur within the boundaries between the cryptographic protocol and the operational election environment. These challenges can be categorised as follows: 1) *Credential custody* (PDF downloads, QR

[6] https://www.mehr-demokratie.de/nachrichten/einzelansicht/abstimmung-21-ergebnisse

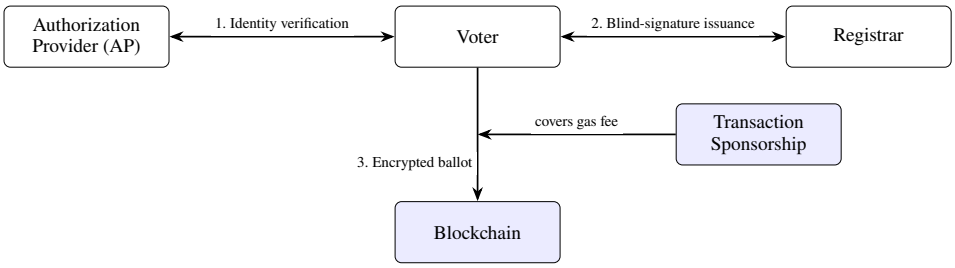

Fig. 1: Architecture of the opn.vote system. The shaded boxes represent blockchain or decentralised components.

codes, browser storage, and recovery) becoming a central usability and security problem; 2) *Ballot semantics* (mapping abstentions, blank answers, and invalid votes onto digital encodings) creating ambiguities inside the system and between the online and postal channels; and 3) *Hidden operational complexity* due to the infrastructure that hid the blockchain from voters shifted that complexity onto the operator, who had to fund, monitor, and maintain it; and 4) *Communication* (mass emails and reminders) introducing system load, cost per vote, and support volume. Finally, we use these findings to directly inform and improve the design of the next version of opn.vote.

2 Foundations

The opn.vote system[7] [MDM25] comprises six principal components. The first component, the *Election Coordinator*, sets up the election, holds the election secret key and concludes the election. Once the voting period ends, the recorded ballots are decrypted, and the result is published so that anyone can recompute the tally. The other five components are directly involved in the three steps of the voting process (see Section 3) shown in Figure 1. More precisely, the voter verifies their identity with the *Authorization Provider* (step 1), then obtains an anonymous credential from the *Registrar* (step 2), and then submits an encrypted ballot that is recorded on the blockchain, with transaction fees covered by the transaction-sponsorship mechanism (step 3). The trust assumptions for the 2025 deployment are discussed in Section 5.5.

A Public Blockchain. A public blockchain is an append-only ledger maintained by many independent nodes rather than by a single organisation [Na08]. Once data have been written to it, it is effectively immutable and publicly readable. In an election context, this property means that any party can retrieve the recorded votes and independently recompute the result, hence confirming the correctness of the tally. For ABSTIMMUNG21 2025, the system

[7] https://github.com/openPetition/opnVote

ran on Gnosis Chain[8], an Ethereum-compatible public blockchain with low transaction fees. Gnosis Chain was selected because it is compatible with standard Ethereum tooling and keeps the on-chain transaction cost per vote well below that of a postal ballot. The blockchain stores the election metadata, the public keys used for ballot encryption, the running counts of authorised and registered voters, and the full record of cast ballots.

Authorization Provider (AP): establishes which voters are eligible to participate in the election. For ABSTIMMUNG21, the openPetition platform served as the AP, verifying voter identity through two channels: 1) the German federal eID and 2) a postal-PIN procedure, in which a unique code was delivered by post. A voter who completed either verification method was confirmed as eligible and could proceed to credential issuance with the *Registrar*.

Registrar: issues exactly one anonymous credential per eligible voter. Its role is architecturally separated from that of the AP. While the AP verifies identity and confirms voters' eligibility, the *Registrar* generates the blind-signature credential [MDM25] for voters. During registration, the voter submits a blinded token, which the *Registrar* signs without learning the underlying value, like signing a document sealed inside an envelope. The voter then unblinds the signed token locally to obtain a credential that is cryptographically unlinkable to any voter-related information observed by the *Registrar* during the signing execution. This unlinkability serves as the system's core anonymity mechanism. Even in the presence of collusion with other parties, the *Registrar* cannot associate a credential later appearing on the blockchain with the blinded token it originally signed. Therefore, it cannot link voters to their ballots. The voter acts as the intermediary, carrying confirmation of eligibility from the AP to the *Registrar* as part of the credential request. For ABSTIMMUNG21, the opn.vote platform served as the *Registrar*.

Voter (Client): interacts with the system through a web application running in their browser. The client generates the voter's secret key material locally, produces the two credential documents (a voting key and an election permit, described in Section 3), encrypts the ballot, and handles all import and export of credentials. All cryptographic operations are performed on the voter's device. No sensitive data is transmitted to any third party. When the voter casts their vote, the browser encrypts the ballot selections and submits the encrypted ballot together with the anonymous credential toward the blockchain.

Transaction Sponsorship: removes blockchain transaction costs from the voter. Each write to a public blockchain incurs a fee in the network's native currency, commonly termed *gas*. Ordinary voters do not hold this currency and cannot be expected to acquire it. Rather than charge voters, the system pays these fees from a pool pre-funded by the

[8] https://docs.gnosischain.com/

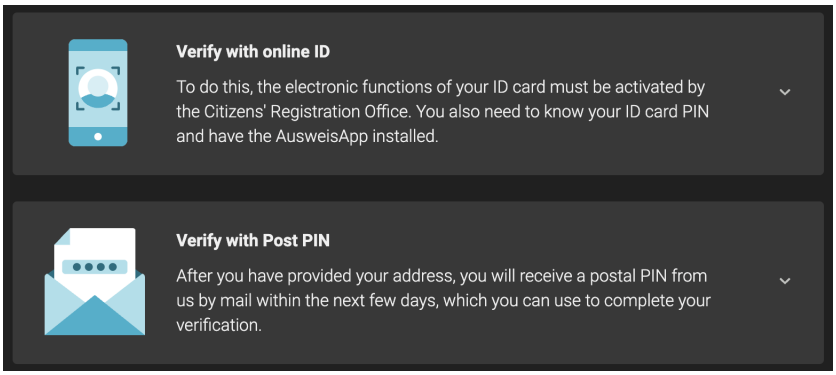

Fig. 2: The two identity-verification methods offered on the openPetition platform.

election organiser, released under publicly verifiable rules that specify who may submit a ballot and how often it may be recast, and open to any client rather than a single provider. Voters therefore need no wallet, no cryptocurrency, and no awareness of fees. This pattern is referred to as gasless voting. It is essential for accessibility, but transfers a substantial operational burden from the voter to the organiser.

3 The Voter Experience

This section describes the process from the voter's perspective. The digital voting process comprised four stages: 1) verifying voter identity, 2) receiving anonymous credentials, 3) casting a vote, and 4) tallying. Registration and voting were separated in time because metadata common to both phases, such as the timing of a voter's actions or the network address they connect from, could otherwise link an anonymous ballot to the identity established at registration. Thus, separating phases reduces timing correlation.

3.1 Verifying Voter Identity

To participate, a voter required a verified openPetition account, which was independent of opn.vote. Voters could verify their identity with the German federal eID or through a postal-PIN procedure (see Figure 2), in which a code was delivered by post.

3.2 Receiving Anonymous Credentials

After identity verification, voters could register via opn.vote for the ABSTIMMUNG21 2025 referendum. In this step, voters obtained the anonymous credentials required to cast a

vote. The registration produced two documents that the voters were required to keep (see Figure 3). The steps of the registration were the following: First, the voters created a *voting key*. This is a secret generated locally in the browser and saved as a PDF containing a QR code. Note that the voting key is not linked to the voter's identity, should not leave the device, and if lost, it cannot be reissued. The voters then obtained an *election permit*, the anonymous credential itself, delivered as a PDF with a QR code, along with the election details and a link to the voting page. Note that the election permit is not linked to the voter's identity. At most one permit was issued per voter [MDM25].

Fig. 3: The two credential documents: *voting key* (left), and *election permit* (right).

These two documents served different purposes. The voting key is the secret seed from which the election permit is derived, while the election permit is the anonymous credential used to cast a vote. The election permit alone is sufficient to cast a vote; the voting key is required only to re-create the permit if it is lost. Once registration closed, no further credentials could be issued or re-created.

3.3 Casting a Vote

During the voting phase, a voter returned to the opn.vote platform and loaded their election permit, either still present in the browser from registration or re-imported from the PDF. All operations were performed locally in the browser.

The voter was presented with the ballot (see Figure 4), which listed the four referendum questions, each offering the options to *approve*, *reject* or *abstain*. Upon confirmation, the browser encrypted the selections and submitted the ballot together with the anonymous credential. The submission was validated and written to the Gnosis Chain, and the voter received a confirmation containing a link to their recorded vote on the blockchain (see Figure 5).

Voters could update their vote up to ten times during the voting period, and only the last submission was counted. Recasting beyond that limit would have required a separate client and self-funded transaction fees.

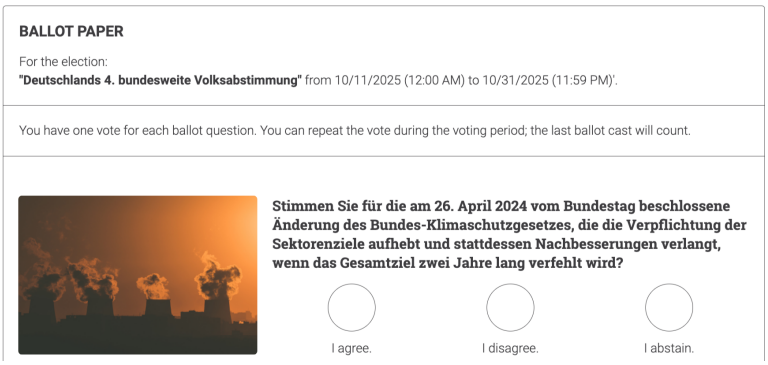

Fig. 4: The digital ballot for Abstimmung21 2025 referendum.

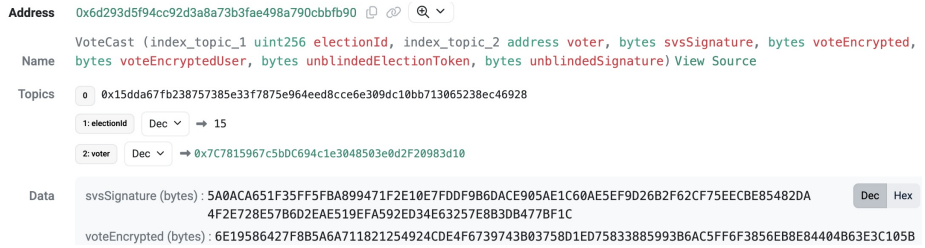

Fig. 5: The recorded vote on the blockchain, viewed as the `VoteCast` event log on a block explorer. The event signature lists all stored fields.

3.4 Tallying and Verifying the Election Result

The tallying phase followed the voting phase. The *Election Coordinator* decrypted the recorded ballots using the election's secret key, validated each, and computed the result. The result and the decryption key were then published on the blockchain.[9]

Voters, or any other interested party, could independently verify the result, or any individual ballot, by retrieving the recorded ballots from the blockchain, decrypting them with the published key, and recomputing the tally. A standalone verification tool for this purpose was published after the election.[10]

[9] On-chain election result for the four questions, together with the published decryption key; the *invalid* tally consists of abstentions and blank answers (Section 5.2): https://gnosisscan.io/tx/0x00132311fac3e4bbb95cbd0e1c4b5a03e648da5c0b913237b59c429c4e1dd119#eventlog

[10] https://github.com/shadmau/opnVote-verifier

4 Deployment Results

Each phase of the online voting process was driven by an e-mail campaign. Table 1 presents three campaign waves: 1) a verification campaign, inviting openPetition users to verify their identity; 2) a registration campaign, inviting verified users to obtain their credentials; and 3) a voting campaign, asking credential holders to cast their vote. The online channel produced 12,922 of the 63,064 total votes. The remaining 50,142 were cast by post.

Tab. 1: E-mail campaigns. Open and click rates are relative to messages delivered.

Campaign	Date	Recipients	Opened	Clicked
Verification				
Invite & reminder	Jun–Jul	≈3.3M	≈38%	≈3%
Registration				
Pilot	28 Aug	4,000	82%	49%
Invite	04 Sep	30,808	81%	48%
Reminder 1	25 Sep	22,420	65%	26%
Reminder 2	06 Oct	16,581	60%	19%
Voting				
Invite	13 Oct	20,477	85%	64%
Reminder	21 Oct	20,408	67%	25%

4.1 Data Sources and Methodology

The results presented in the following subsections are based on operational data collected during the deployment, namely openPetition logs (the number of users who verified their identity, the split between eID and postal-PIN verification, and the number of voters who began but did not complete credential issuance), opn.vote logs (the number and timing of successful blind-signature credential issuances), and on-chain events (the number and timestamps of recorded votes and recasts). E-mail campaign metadata provided send dates, open rates, and click-through rates for each campaign wave. Each wave produced a concentrated spike in the corresponding phase, which in turn shaped infrastructure load and cost (see Section 4.3). Sponsorship-wallet records supplied the total sponsored cost, cost per vote, and the pricing events that drove the wallet outage (see Section 4.4). Finally, the 1,726 support messages received during the registration and voting periods were classified into the categories discussed in Section 4.5.

Note that the voter-facing web client did not include client-side analytics, e.g., tracking of page views, clicks, or user flows. Hence, we do not have counts of how many voters opened the voting page.

4.2 Participation Funnel

Because authorisation, credential issuance, and recorded votes are all logged on-chain, their counts can be checked for consistency, since each stage must not exceed the one before it. This held throughout the deployment. Table 2 shows the participation funnel for the online channel. Thereby, three observations stand out.

First, we identified a significant drop-off at identity verification. Of the 3.3 million contacted users, about 99,000 clicked the verification link, and of those about 45% (44,251) completed verification using eID (26%) or postal PIN (74%).

Second, 23,533 voters started the credential-issuance process, but only 20,583 completed it (88%). The 12% drop-off corresponds to voters who began the multi-step workflow on opn.vote (clicking start, creating a voting key, downloading the PDF) but abandoned before completing all steps. Because the process requires several clicks and two separate PDF downloads, each intermediate step was a point where voters could drop out.

Third, a major actionable loss was between the issuance of credentials and the recorded votes. Thus, 20,583 credentials were issued but only 12,922 votes were recorded, a drop of 37%. Since no client-side analytics were collected, we cannot determine how many of these voters opened the voting page, attempted to import their credential, or encountered errors. However, the support data (see Section 4.5) suggest that credential custody problems (lost PDFs, confusion between the two documents, incorrect file format) account for at least part of this loss.

Tab. 2: Participation funnel and verification-method split.

Stage	Count	Retained
Contacted (verification campaign)	≈3,300,000	–
Followed verification link	≈99,000	3%
Identity verified	44,251	45%
via eID	11,621	26%
via postal PIN	32,630	74%
Credential request started	23,533	53%
Credentials issued	20,583	88%
Online vote recorded	12,922	63%

4.3 Timing of Voter Activity

Voter activity was strongly concentrated around the e-mail campaigns. Figure 6 shows the issuance of credentials per day during the registration phase, and Figure 7 shows recorded votes per day during the voting phase; in both, the campaign send dates are marked, and each send is followed by a pronounced spike. More than 50% of voting activity occurred within 24 hours of an e-mail send, confirming that a mass e-mail is as much an infrastructure

load event as a communication one. Furthermore, a minority of voters used recasting: of 12,922 voters, 294 submitted at least one recast, for a total of 304 recast transactions.

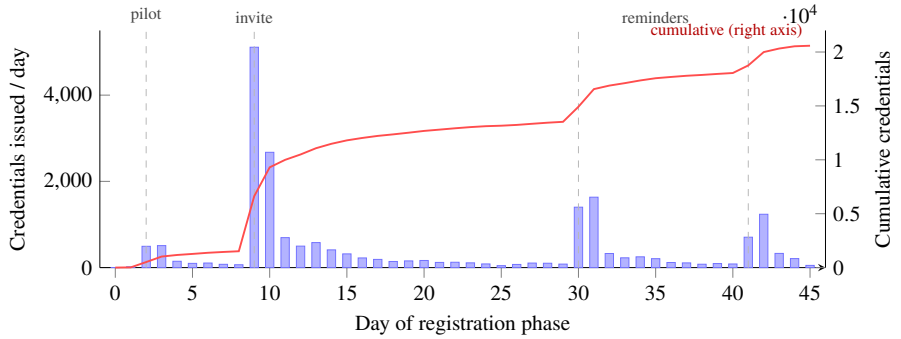

Fig. 6: Credential issuance per day during the registration phase. Bars show daily issuances (left axis); the line shows cumulative issuance (right axis). Dashed lines mark e-mail sends (pilot, invitation, and reminders).

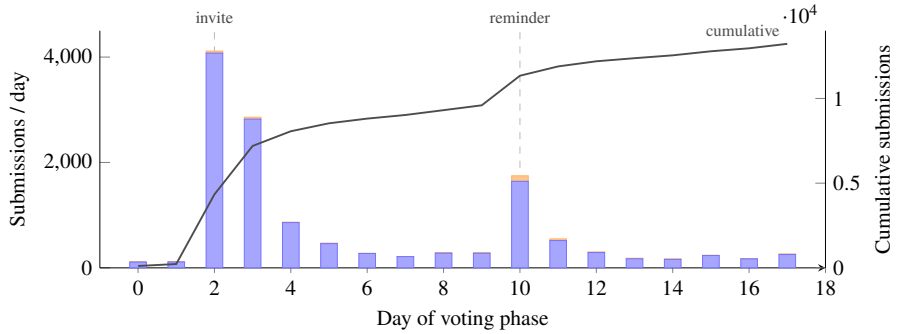

Fig. 7: Daily online submissions during the voting phase: first-time votes (blue) with recasts stacked on top (orange), left axis; the line shows cumulative submissions (right axis). Dashed lines mark e-mail sends (invite, reminder).

4.4 Transaction Outcomes, Cost, and the Wallet Outage

The transaction sponsorship was realised as a sponsorship wallet, pre-funded by the organiser and a commercial *relayer service* (Gelato)[11], which submitted each ballot to the Gnosis Blockchain and drew the corresponding fee from that sponsorship wallet. Before reaching the relayer, every submission first passed through an off-chain *Signature Validation Server* (*SVS*), which checked the voter's credential and forwarded only valid ballots. Both the single relayer and the off-chain check were operational choices specific to this deployment. Of 14,084 vote-submission requests reaching the SVS, 13,226 were validated and recorded on-chain (12,922 first-time votes and 304 recasts), while 858 were rejected.

[11] https://docs.gelato.network/

A wallet outage. The 858 rejected submissions were caused by a single incident. The sponsorship wallet had been funded for the expected per-vote cost of approximately 0.02 cent. When the first voting-invitation e-mail was sent, the resulting traffic surge pushed the relayer into its auto-scaling pricing tier, where each transaction cost approximately 3.0 cent, roughly 150 times the normal rate. The wallet was drained within minutes and for approximately 30 minutes all submissions were rejected because there were no funds to pay gas fees. After the wallet was manually refilled, voting resumed normally. The 858 count may not represent 858 unique voters, as some voters likely retried during the outage. Of the affected voters, 102 contacted the support by e-mail; all were answered within one day, and some confirmed that voting worked after the outage ended.

Cost structure. The total sponsored on-chain transaction cost for all 13,226 digital votes was approximately 53.13 USD, excluding the relayer's fixed subscription fee. The cost had two distinct regimes. Under normal load, the Gelato relayer processed transactions in its standard pricing tier at approximately 0.02 cent per vote[12], comfortably below the cost of a postal ballot (approximately 2 Euros per postal vote). However, the on-chain transaction costs outlined above are not directly comparable to the postal vote costs, as they exclude identity verification costs, server and subscription costs, support costs, and other overhead costs. During high-volume bursts (immediately after e-mail sends), the relayer's throughput was exceeded and it switched to an auto-scaling tier, where each transaction cost approximately 3.0 cent. Although only 12.7% of submissions fell into this tier, those submissions accounted for approximately 50.67 USD of the total 53.13 USD cost. In other words, about one-eighth of the votes caused 95% of the cost. Figure 8 plots the costs per vote along with the submission rate during the voting phase. The two spikes in cost (days 2–3 and day 10–11) coincide precisely with the e-mail sends, confirming that communication timing directly drives cost.

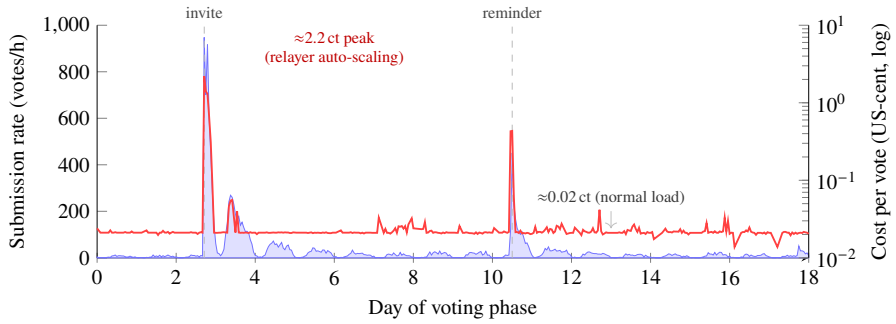

Fig. 8: Submission rate (left axis) and cost per vote (right axis, log). Cost per vote is measured on Gnosis Chain and reported in US-cent. Dashed lines mark e-mail sends (invitation, reminder).

[12] This was the cost for the vast majority of votes.

Tab. 3: Support requests by category.

Category	Count	Share
Identity verification (eID / postal PIN)	781	45%
Content-related criticism (topics / referendum format)	407	24%
Import and submission problems	194	11%
Credential creation (e.g. document lost or not saved)	192	11%
Submission failures (reported)	102	6%
Other (privacy/security questions, usability)	50	3%
Total	1,726	100%

4.5 Support Volume and Voter Difficulties

The deployment generated 1,726 support requests, each answered individually. A recurring challenge was to reproduce issues because credential material was stored only locally, and sometimes not at all, thus support staff could not inspect the voter's state.

Table 3 groups requests into categories. The "Share" column shows each category's percentage of the 1,726 total. The working hours spent on answering support requests have not been recorded; thus a direct comparison with hand counting paper ballots of the postal channel is not possible.

Two observations are noteworthy. First, the two largest categories, identity verification (45%) and content-related criticism (24%), lay entirely outside the opn.vote system. eID and postal-PIN problems arose before voters reached opn.vote, and content-related messages were mostly objections to the referendum topics or format rather than usability issues. Together, 69% of support requests concerned issues that a voting system cannot address. Second, the remaining 31% of requests split into credential handling (creation and import, 22%), submission failures caused by the wallet outage (6%, Section 4.4), and other issues (3%). Credential handling was thus the main recurring opn.vote-specific difficulty.

5 Lessons Learned

The cryptographic core performed as designed: credentials were unlinkable, the on-chain record was consistent and publicly verifiable, and no integrity violations occurred. The central finding of the deployment is that the most relevant practical challenges arise at the boundaries between the protocol, voter behaviour, election operations, and blockchain infrastructure. We organise our lessons learned into five themes.

5.1 Credential Custody

The largest single cluster of practical challenges was the handling and storage of voter credentials. The protocol issues each voter two artefacts, a *voting key* (the master seed) and an *election permit* (the anonymous credential), and after issuance, custody of both rests entirely with the voter (see Section 3.2). This introduced a usability and security challenge that the system could not control once a credential left the browser. Support tickets and operator experience revealed several recurring problems.

First, the two-artefact model itself was a source of confusion. Support tickets showed that some voters attempted to import the voting key rather than the election permit, or saved only a link rather than the election permit PDF. The similar names in German (*Wahlschlüssel* vs. *Wahlschein*) compounded this.

Second, some voters scanned the QR code of the election permit with their phone camera instead of using the opn.vote client's import function. The camera may interpret the QR content as a URL and open it in a browser or search engine, which could send the credential data to an external service or store it in the browser history. Any party who obtains the credential from such a side channel could cast or recast a vote. This threat extends to any credential-import path outside the voting client.

Third, the behaviour of the device and browser further complicated the custody. On some devices, the PDF was downloaded automatically; on others, the voter had to locate and save it manually. As a fallback, the system retained credentials in local browser storage, which reduced drop-off but made the custody model less transparent, since voters were unaware of the stored credential, and clearing the cache or switching devices silently removed it. Usability and custody support thus worked against each other.

This cluster of challenges is consistent with the 37% drop-off between credential issuance and recorded votes observed in the funnel (see Section 4.2). A substantial fraction of voters who obtained credentials never voted. While the lack of client-side analytics does not allow full attribution of these drop-offs, credential custody is the most visible contributing factor. Hence, we derive that the system designer cannot treat credential custody as the voter's responsibility. Future deployments must therefore treat credential custody as a crucial design concern.

5.2 Ballot Semantics and Tallying Consistency

Another challenge emerged between the legal semantics of a ballot and its technical encoding on the digital interface. This was further emphasised by the parallel paper process, whose results had to be reconciled with the digital ones. Each of the four questions offered approval, rejection, or abstention, but voters did not always map their intent onto these options as expected by the system.

The clearest example involved blank options. Of the 12,922 voters, 155 left one or more options with no selection rather than actively choosing abstention. Leaving *all* questions blank produced a submission error, whereas leaving *some* blank was accepted, which was an unintended inconsistency. The interface did not encode a blank answer as abstention or as invalid, and thus it was treated as undefined in the tallying process. Since voters were not informed how a blank answer would be treated, even a default mapping to abstention might not be a voter's intention. More broadly, paper and digital ballots expose fundamentally different voter behaviours. On paper, invalid votes arise from ambiguous or stray marks, whereas in a digital interface such cases must be represented explicitly or prevented by design, and both channels' rules must agree for a combined count.

Hence, we derive the following lessons. First, every possible voter action, including inaction, such as leaving an option blank, must map to a defined ballot state before the system is deployed. Allowing an undefined state creates ambiguity that can be challenged in any post-election review, especially when digital and postal results are combined. Second, a single *ballot-state model* covering approval, rejection, abstention, and technically invalid encodings must be agreed with all election stakeholders and enforced consistently across the frontend, backend, vote encoding, and tallying logic.

5.3 Operational Complexity of the Gasless Infrastructure

The gasless design successfully concealed wallets, gas, and blockchain interaction from voters but relocated that complexity to the election operator. This is a structural consequence of the gasless pattern. Any blockchain-based voting system that sponsors transaction fees will face analogous responsibilities because the underlying blockchain still requires gas, and someone must fund, monitor, and manage it.

In practice, the relayer availability, funding, and throughput became election-critical responsibilities. An underfunded wallet or an overloaded relayer directly translates to voters being unable to submit their ballots. This availability failure is invisible in the protocol specification but immediately visible to voters. The wallet outage described in Section 4.4 demonstrates this fragility. The wallet was provisioned for normal-tier cost but drained within minutes when the post-invitation traffic surge triggered the relayer's auto-scaling tier, rejecting all submissions for approximately 30 minutes. While sponsorship restrictions limit the number of votes an individual voter can send, the sponsorship layer is critical for security, as its unavailability directly impacts the availability of the voting system.

The cost data (see Section 4.4) reinforced this point. The total expenditure was driven less by the number of votes than by their temporal concentration, itself a consequence of mass e-mail sends. Staggering e-mail delivery over hours rather than sending all invitations at once could therefore reduce cost. Occasional outages during pre-election testing did not affect the final voting period but exposed availability assumptions that are not considered in the protocol.

We derive that gasless voting improves voter usability at the cost of a hidden operational layer that must be provisioned, monitored, and maintained like any other election-critical infrastructure component. For election operators and practitioners evaluating blockchain-based voting, this means that the operational overhead of gasless submission should be budgeted alongside the voting system security guarantees.

5.4 Election Operations and Communication

Another challenge was between the technical system and the human operation of the election. The patterns observed here, such as communication-driven load spikes, the need for runtime observability, and the fragility of concentrated operational knowledge, are relevant to any practitioner running an online voting system, because they will recur regardless of the specific system.

As shown in Section 4.3, mass e-mails acted as infrastructure load events. Therefore, communication had to be planned together with the relayer's capacity, sponsorship budget, and real-time monitoring. Moreover, schedule changes compounded this challenge. The voting period was adjusted after some credentials had already been issued, which left an outdated, later end date on a number of already-downloaded election permits. Two voters relied on that cached date and tried to vote after the actual end time, receiving a generic error message. Further, runtime observability was also insufficient. Some failures could be diagnosed only after the fact because the client did not emit informative error messages, and support staff could not inspect credential state stored only on the voter's device.

We derive that election communication and operations must be treated as part of the technical deployment, with campaign scheduling planned according to infrastructure capacity. Furthermore, adding privacy-preserving drop-off analytics, treating runtime observability as crucial, and enabling distributed operational access for the support team.

5.5 From Concept to Practice: Design Decisions and Trade-offs

The opn.vote design targets a decentralised system in which election-critical checks are transparently enforced on-chain, ballot submission is open to any client, and no trusted server is required for voter anonymity. The 2025 deployment was a deliberate first step toward this goal, but it simplified its surroundings in three known ways.

Anonymous channel. The protocol assumes that votes reach the blockchain over a channel that hides the voter's network identity and timing of submission. Providing such a channel to non-technical voters at scale is non-trivial. Hence, this first deployment omitted it and relied on the blind signature's cryptographic unlinkability alone, accepting that the operator

could observe IP addresses and submission timestamps. Anonymity in this deployment thus rested partly on the operator not collecting and correlating network-level meta-data of voters with voter identities.

Off-chain credential verification. The protocol verifies each blind-signature credential on-chain. The deployment verified it instead in an off-chain *Signature Validation Server (SVS)*, because a defect in a server can be fixed while voting is underway, whereas a defective smart contract cannot. This introduced a new trust assumption. While the SVS cannot alter or forge ballots because every submission is signed on the voter's device, it must be trusted to forward every valid ballot.

Single relayer, off-chain sponsorship rules. The protocol provides for open submission and publicly verifiable sponsorship rules. The deployment routed all submissions through one commercial relayer (Gelato), with the sponsorship rules (including the ten-recast limit) publicly announced but enforced in the backend. This outsourced infrastructure and left the sponsorship rules unverifiable from the outside.

While none of these trade-offs affected vote secrecy or integrity, each shifted the protocol's trust assumptions. Thus, we derive that these simplifications need to be removed in further iterations.

6 Conclusion

The first opn.vote deployment in ABSTIMMUNG21 2025 produced 12,922 online votes and demonstrates that blockchain-backed voting can be made accessible to non-technical voters without exposing wallets, cryptocurrency, or gas management. No integrity violations occurred and the on-chain record remained publicly verifiable throughout. The hardest practical challenges lay not inside the cryptographic protocol, but at its boundaries with real voters and real-world election operations, in particular credential custody, ballot semantics, the hidden operational layer behind gasless submission, and the coupling between communication and infrastructure load. Support requests proved to be an informative diagnostic source, surfacing failure modes that were invisible in protocol-level logs.

While the deployment uncovered multiple operational difficulties, most of the findings can be addressed through process changes and technical iterations. The results of the deployment directly feed into the next iterations of opn.vote. Several infrastructure improvements have already been implemented for the ABSTIMMUNG21 2026 referendum. Credential validity is now checked on-chain, removing the off-chain SVS entirely; ballot submission can now be routed through interchangeable or self-hosted providers rather than one commercial relayer, so that independent clients can also submit votes; and sponsorship rules, including the recast limit, are enforced in publicly verifiable logic.

Furthermore, a single ballot-state model that covers all voter actions, including inaction, will be defined before deployment, enforced across the frontend, encoding, and tallying, and agreed with the paper-channel stakeholders. Campaign scheduling will be planned according to the capacity of the infrastructure. Since pre-election testing cannot surface every failure mode in heterogeneous voter environments, privacy-preserving funnel analytics and structured client-side error reporting will be added.

Moreover, to address credential custody and submission privacy, a dedicated application is being developed targeting the ABSTIMMUNG21 2027 referendum. This application stores credentials securely on the voter's device, removing the need for manual PDF and QR-code handling. The same application provides a controlled network stack for adding network-level submission protections. As a signed and open-source app, it also strengthens client integrity through public auditability. Every credential-import path is treated as a security-sensitive interface, evaluated for whether it exposes credentials to external services, browser history, or cloud synchronisation.

Finally, the findings presented in this paper, derived from a single medium-scale deployment of a non-binding civil-society referendum, show the practical challenges a blockchain-backed voting system faces once real voters are involved. While the deployment revealed a number of practical issues, several have already been addressed in the current version of opn.vote, as described above, and the remaining lessons continue to inform subsequent iterations of the system.

References

[MDM25] Maduakor, F.; Doan, T. V. T.; Mitzlaff, J.: opn.vote: A Publicly Verifiable Blockchain-Based eVoting System for Democratic Participation. In: VOTING'25 Workshop, Financial Cryptography and Data Security 2025 (FC '25). 2025.

[Na08] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. 2008.