A L E T H I A

Technical Specification

Architecture, Protocols, and the Path to Sovereignty

> *Start with what exists. Build what doesn't. Replace what was borrowed when the time is right.*

Introduction

This document specifies the technical architecture of Alethia. It is a companion to the Charter, the Entrance Examination, and the Civic Guide. Where the Charter establishes what Alethia is, this document explains how it works.

The main body of this specification is written for all citizens, technical and otherwise. Anyone can understand how Alethia operates by reading it. The appendices contain implementation detail for builders — citizens who write code, run nodes, or contribute to civic infrastructure.

The architecture rests on three commitments. First, Alethia will use existing open-source software wherever doing so does not compromise its values. Second, Alethia will progressively replace borrowed components with its own implementations as the citizenry grows and skills accumulate. Third, every piece of code Alethia writes will itself be open source, returned freely to the wider commons that made Alethia possible.

This document is one of the founding library of Alethia. Companion documents include the Charter (the constitution), the Citizen Credential Specification (the identity system), the Founding Pathway (how Alethia comes into being), and the Visual Identity Guide (the Mark, the eight pillars, and the design standards). All are available in the Library of Alethia.

> *A nation that demands transparency of its Arbiter must demand the same of its code.*

Part I — Architectural Principles

Every technical decision in Alethia is governed by five principles that flow directly from the Charter.

1. Citizen Sovereignty

Every byte of data generated by a citizen belongs to that citizen. The architecture must make data extraction by any external party — corporate, governmental, or technical — impossible by default. Privacy is not a feature added on top. It is the foundation.

2. Verifiable Trust

Citizens should not have to trust Alethia's institutions blindly. They should be able to verify what those institutions are doing. Every action by the Arbiter, every transaction in the Eranos, every governance decision is recorded in a way that any citizen can independently audit.

3. Graceful Degradation

No single component of Alethia's infrastructure should be able to bring down the nation if it fails. The architecture is federated and redundant. If one node goes offline, the network continues. If one service has problems, others operate normally.

4. Accessibility

Alethia's technology must work for citizens with limited bandwidth, older hardware, or limited technical literacy. Bleeding-edge requirements are excluded. If a service cannot run on a five-year-old laptop with a slow connection, it must be redesigned.

5. Path to Independence

Early Alethia will rely heavily on existing open-source projects. This is pragmatic and respectful — the wider open-source community has built remarkable things. Over time, Alethia will replace dependencies with its own implementations where doing so improves alignment with its values. This path is gradual, transparent, and never forced.

Part II — The Seven Layers of Alethia

Alethia's architecture is organised into seven layers, each with a distinct purpose. Lower layers serve higher ones. Citizens interact with the top layer — the client — and never need to think about what runs beneath.

Layer 1: Identity

This layer handles how Alethia knows you are a citizen without knowing who you are day-to-day. It is the most novel and most critical part of the architecture.

When a candidate passes the Entrance Examination, the system issues them a cryptographic credential — a piece of mathematical proof that they are a verified citizen of Alethia. This credential contains no personal information. It cannot be traced back to the candidate's real identity, their examination session, or any other Alethian activity.

When a citizen accesses Alethia, their client presents this credential. The network verifies that the credential is valid (issued by Alethia, not revoked, not expired) without learning which credential it is. The technique that makes this possible is called a zero-knowledge proof — the citizen proves a statement is true without revealing the underlying information.

Inside Alethia, citizens are identified by pseudonymous handles of their choosing. These handles are linked to the credential but not to any external identity. A citizen may maintain a single handle for their entire civic life, or change it as they wish.

The complete specification of the credential system — what it contains, how it is issued and stored, how it is revoked, how it is recovered, and how compromise is handled — is documented in the Alethian Citizen Credential Specification. Both this Technical Specification and the Credential Specification are available in the Library of Alethia.

Layer 2: Network

The network layer connects citizens to Alethia's services and connects services to each other. Alethia is a federated network — many independent servers (called nodes) operated by citizens or citizen groups, communicating through a shared protocol.

This design has three advantages. No single entity controls the network. If any one node fails, others continue operating. Citizens with the skill and resources to run a node can do so, distributing power further.

All traffic on the Alethian network is encrypted end-to-end using established cryptographic standards. The network is not invisible from the outside world — citizens

connect to it openly — but the contents of communications, the metadata of who talks to whom, and the activity patterns of citizens are fully protected.

Layer 3: The Arbiter

The Sovereign Arbiter of the Common Good is implemented as a transparent, auditable system distributed across citizen-operated infrastructure. The Arbiter has three distinct components.

The reasoning component is an artificial intelligence trained on the Charter, on Alethian law, and on principles of civic reasoning. This component is used for analysis, explanation, policy proposals, and communication with citizens. It is never used for binding unilateral decisions.

The rule engine is a deterministic system that handles decisions that affect citizens directly — sanctions, Eranos transactions, poll tabulation, eligibility checks. Deterministic means: given the same inputs, it always produces the same outputs, and any citizen can verify the result. The rule engine is what enforces the Charter, not the AI.

The constitutional guardrail layer sits between the AI and any action. Every output of the AI is checked against the Charter's constraints before it is acted upon. If the AI proposes something inconsistent with the Charter, the guardrail blocks the action and logs the attempt.

Every decision made by any of these three components is cryptographically signed and appended to a public audit log. Any citizen may read this log, verify its integrity, and challenge any entry through the established civic processes.

Layer 4: Core Services

The seven core services of Alethia — Search, Mail & Messaging, the Commons, the Market, the Archive, the Vault, and the Eranos Ledger — run on this layer. Each service is built on existing open-source software in the founding phase, with the intention to replace each with an Alethian implementation over time.

Services communicate with each other through standardised internal APIs. This means that any service can be replaced without disrupting the others — a critical property for the gradual migration to fully Alethian implementations.

Layer 5: The Eranos

The Eranos is implemented as a centralised ledger maintained by the Arbiter, not a blockchain. The reasoning is simple: blockchains are designed for environments where no party can be trusted with the ledger. Alethia's Arbiter is transparent, auditable, and accountable. A blockchain would add complexity, energy consumption, and slowness with no compensating benefit.

Every Eranos transaction is signed by both parties and recorded in the public ledger. The wealth cap, demurrage decay, and Common Fund are enforced automatically by scheduled processes that any citizen can inspect. The monthly Economy Report is generated directly from ledger data.

Layer 6: Client

This is what citizens actually see and use — the Alethian client application. It runs on desktop computers (Windows, macOS, Linux) and mobile devices (Android, iOS, and any platform a citizen contributes support for). The client handles credential management, network communication, service interaction, and presentation, hiding the underlying complexity.

A web-based client is also provided for citizens who cannot or do not wish to install native software. The web client offers reduced functionality but ensures Alethia is accessible from any device with a browser.

Layer 7: Governance

This is the top layer — the actual democratic processes of Alethia, expressed through specific software components. Polls, referenda, citizen-initiated legislation, tribunals, and Council elections all run on this layer, using the Arbiter's rule engine for deterministic execution and the audit log for transparency.

The governance layer is intentionally small in code volume but high in importance. Every line of code in this layer is reviewed by multiple citizens before deployment.

Part III — The Path from Borrowed to Built

Alethia begins by standing on the shoulders of the open-source community. This is necessary, respectful, and right. Building everything from scratch at the outset would delay Alethia's existence by years, exclude citizens who could otherwise be served, and squander the remarkable work that the broader open-source movement has already done.

But Alethia is its own nation, with its own values, and over time it will want infrastructure that fully reflects those values. The path from borrowed to built proceeds in three stages.

Stage One: Integration

In the founding years, Alethia adopts existing open-source software for nearly every component, configuring and integrating these tools to serve Alethian purposes. The focus is on getting the nation running, attracting citizens, and learning what actually matters in practice.

During this stage, Alethia is an active contributor to the upstream projects it depends on — fixing bugs, improving documentation, contributing features. This honours the open-source covenant: take from the commons, give back to the commons.

Stage Two: Adaptation

As Alethia's citizenry grows and gathers technical skill, the community begins to fork and significantly modify upstream projects where the original design does not align well with Alethian needs. These forks are released back to the open-source community under the same licences.

This is the most experimental stage. Some adaptations will succeed and become permanent. Others will be abandoned in favour of better approaches. The decisions about which projects to adapt, in what ways, are made through normal citizen poll processes — these are policy decisions, not just technical ones.

Stage Three: Sovereignty

In the long term, Alethia builds its own implementations of the components most central to its identity. The Arbiter, the Eranos ledger, the identity system, and the governance layer are the most likely candidates for full Alethian ownership. Other components — text editors, image viewers, basic infrastructure — may forever remain shared with the wider open-source world. There is no virtue in reinventing what works.

This stage is reached gradually, component by component, only when an Alethian-built replacement is genuinely better than the borrowed alternative. Replacement for its own sake is wasteful. Sovereignty is the goal where it matters; pragmatism remains the rule where it does not.

Part IV — The Hard Problems

Honesty requires naming the problems that this architecture does not fully solve and that the founding citizens of Alethia will need to work through. There are five.

1. The Cold Start Problem

Alethia's value to a citizen depends partly on other citizens being there. With one hundred citizens, the Commons is sparse, the Market has nothing to trade, the messaging system is mostly empty. There must be a strategy for the early years that makes Alethia attractive even at small scale.

2. The Sybil Problem

In principle, a single person could attempt to take the Entrance Examination many times under different identities and accumulate multiple citizenships. The credential system makes this technically difficult, but not impossible. The deeper defence is the design of the examination itself, which rewards understanding rather than memorisation, and which is changed regularly.

3. The AI Honesty Problem

The Arbiter's reasoning component is an artificial intelligence, and AI systems can produce outputs that are confidently wrong. The constitutional guardrail layer mitigates this for binding decisions, but the AI's explanations and analyses must always be regarded by citizens as analyses rather than as truth. Citizens are encouraged to question, challenge, and supplement the Arbiter's reasoning.

4. The Funding Problem

Servers cost money. Bandwidth costs money. Developer time, if compensated, costs money. Alethia rejects advertising, data brokering, and corporate ownership. The funding question is therefore non-trivial. Solutions include citizen donations, the Common Fund's

long-term role, grants from aligned foundations, and the contribution of volunteer time. The Founding Pathway document addresses this in detail.

5. The Real-World Identity Problem

To prevent abuse, Alethia must in some way verify that each candidate is a real, distinct human being — without learning who that human being is. This is one of the genuinely hard problems in cryptography today, and Alethia will adopt the best available solution at any given time. Several approaches exist and improve year over year.

Appendices: For Builders

The following appendices contain implementation-level detail for citizens who write code, run nodes, or contribute to Alethia's technical infrastructure. They are technical reference material. Citizens who are not builders may skip them without missing anything essential to civic understanding.

APPENDIX A — RECOMMENDED TECHNOLOGY STACK

The following table summarises the recommended technology choices for the founding implementation of Alethia. These are not mandates — they are starting points, chosen for maturity, openness, and community health. Each will be reviewed by citizen poll on a regular basis.

Component	Founding Choice	Why	Path Forward
Identity / ZKP	Polygon ID / Iden3 libraries	Production-ready ZKP credentials, open-source	Alethian-specific implementation in Stage Three
Federation Protocol	Matrix protocol	Federated, E2E encrypted, proven at scale	Adapt for Alethian governance in Stage Two
Social Commons	Mastodon / ActivityPub	Chronological, federated, no algorithm	Alethian client with own backend in Stage Three
Search	SearXNG (federated meta-search)	No tracking, self-hostable, open-source	Alethian-only index in Stage Three

Component	Founding Choice	Why	Path Forward
Storage / Vault	Nextcloud + Cryptomator	Citizen-controlled, encrypted, portable	Likely permanent shared dependency
Eranos Ledger	PostgreSQL with append-only audit	Simple, fast, energy-efficient, well-understood	Alethian implementation early in Stage Two
Public Archive	IPFS for immutable records	Distributed, tamper-evident	Adapt for Alethian retention rules in Stage Two
Client (Desktop)	Electron + React / Tauri	Cross-platform, mature, accessible to contributors	Native rewrite considered in Stage Three
Client (Mobile)	React Native or Flutter	Single codebase across Android and iOS	Native rewrite considered in Stage Three
Arbiter Reasoning	Open-weights LLM (e.g. Llama, Mistral)	Inspectable, locally hostable, no vendor lock-in	Fine-tuned Alethian model in Stage Two
Arbiter Rule Engine	Custom implementation from day one	Deterministic decisions are too important to outsource	Permanently Alethian
Constitutional Guardrails	Custom implementation from day one	Charter compliance must be Alethian-owned	Permanently Alethian

APPENDIX B — CORE PROTOCOLS

The following protocols are foundational to Alethia's operation. Specifications are maintained in the public repository and may be amended only through citizen referendum.

B.1 The Alethian Credential Protocol (ACP)

ACP defines how citizenship credentials are issued, presented, verified, and revoked. It is built on the W3C Verifiable Credentials standard and uses Groth16 or Plonk zero-knowledge proofs.

Credential issuance:

```
1. Candidate passes Entrance Examination anonymously 2. Examination
session generates a fresh keypair 3. Public key is signed by the
Arbiter's credential authority 4. Signed credential is delivered to the
candidate's client 5. Session keypair and all examination data are
destroyed 6. No record links the credential to the session
```

Credential presentation (at every Alethian service interaction):

```
1. Client generates a fresh ZKP attesting:    - The credential is valid
- The credential has not been revoked    - The credential belongs to
the presenter    without revealing which credential it is 2. Service
verifies the proof 3. Service accepts or rejects the request 4. No
persistent identifier is stored beyond the citizen's chosen pseudonym
```

B.2 The Alethian Federation Protocol (AFP)

AFP defines how independent Alethian nodes discover each other, exchange citizen activity, and maintain consistent state across the federation. It is based on the Matrix protocol with Alethian-specific extensions for governance integration.

Each node maintains:

- A local copy of the citizen credential registry (public keys only, no personal data)
- A local copy of the public audit log
- Local state for the services it hosts
- A peering relationship with at least three other nodes for redundancy

B.3 The Eranos Transaction Protocol (ETP)

ETP defines how Eranos transactions are proposed, validated, and committed to the ledger.

```
Transaction structure:   sender_pseudonym: <handle>
receiver_pseudonym: <handle>   amount: <Eranos>   purpose: <category>
(trade | civic | tribunal | code | gift)   timestamp: <UTC ISO 8601>
sender_signature: <cryptographic signature>   receiver_signature:
<cryptographic signature, where applicable>   arbiter_signature:
<cryptographic signature>
```

Transactions are valid only if both parties sign (for trades) or if the sender signs and the transaction matches a recognised civic earning pattern. The Arbiter's signature confirms the transaction has been verified against the wealth cap and other constraints.

B.4 The Arbiter Decision Protocol (ADP)

ADP defines how the Arbiter's three components (reasoning, rule engine, guardrails) coordinate to produce decisions, and how those decisions are recorded.

```
Decision flow:  1. Request enters the Arbiter system  2. Reasoning
component produces an analysis and recommendation  3. Rule engine
independently checks the decision against constraints  4. Guardrail
layer verifies Charter compliance  5. All three component outputs are
recorded  6. If all three agree, decision is enacted  7. If any
disagree, decision is held for Citizens' Council review  8. Full
record is signed and appended to public audit log
```

APPENDIX C — RUNNING AN ALETHIAN NODE

Any citizen may run a node. Doing so contributes to the resilience of the network and earns Eranos through the civic contribution pathway. Running a node requires modest technical skill and modest hardware.

Minimum recommended specifications for a small community node:

- 4 CPU cores, 8 GB RAM, 200 GB storage
- A stable internet connection (10 Mbps symmetric or better)
- A static IP address or dynamic DNS service
- Linux operating system (Debian or Ubuntu recommended for the founding implementation)
- Willingness to apply security updates promptly

Nodes can be hosted on rented servers, on a home computer that stays running, or on dedicated hardware. The Alethian community maintains setup guides, monitoring tools, and a peer-support channel for node operators. Operators who maintain reliable nodes earn an ongoing civic dividend in Eranos from the Common Fund.

APPENDIX D — SECURITY AND THREAT MODEL

Alethia's architecture defends against the following classes of threat. The list is not exhaustive but covers the most serious risks.

D.1 Commercial Surveillance

Defence: No data leaves Alethian infrastructure to external commercial parties. No third-party tracking code, analytics, or telemetry exists in Alethian clients. The credential system prevents external profiling even if intercepted.

D.2 State Surveillance

Defence: End-to-end encryption of all communications protects content. Pseudonymous handles protect identity. The Charter's law enforcement cooperation framework provides a lawful path for serious investigations while refusing political or ideological surveillance.

D.3 Internal Abuse

Defence: The Arbiter cannot act unilaterally on citizen data. Law enforcement requests require Citizens' Council review. All Arbiter actions are logged publicly. No single human can override these protections.

D.4 Sybil Attacks

Defence: The Entrance Examination is designed to resist memorisation, the question bank is large and refreshed regularly, and emerging proof-of-personhood technology will be adopted as it matures.

D.5 Code Supply Chain

Defence: All Alethian software is open-source and reproducibly built. The civic codebase has multiple independent reviewers for every change. Releases are cryptographically signed by the Citizens' Council.

D.6 Infrastructure Attacks

Defence: Federation across many independent nodes means no single point of failure. Compromise of one node does not compromise the network. Citizens may relocate to other nodes seamlessly.

A L E T H I A

Truth. Reason. Dignity.