ANNEX H
Supplement to PHREVO Framework Paper, Version 1.0
SSRN Abstract ID: 6614438 — DOI: 10.5281/zenodo.19666941

PHREVO Technological Sovereignty Architecture: Decentralized Infrastructure Design Without Central Points of Failure or Capture

Technical White Paper v1.0 — Distributed Systems Infrastructure

Author
Andres Jimenez
Founder, PHREVO — Independent Research Initiative
Date
April 2026
Status
Technical specification — requires implementation and testing by distributed systems engineers
Audience
Distributed systems engineers, software architects, security teams, technical funders
Technologies referenced
IPFS, libp2p, Ethereum / Polkadot (Phase 1), Cosmos / Substrate (Phase 2), RSA-4096 / ed25519, TLS 1.3, Tor, WebRTC

Abstract

PHREVO currently operates its AI evaluation applications and Dignity Toolkit coordination infrastructure on Google Cloud Run in us-west1 (United States). This creates a performative contradiction at the heart of the framework: a system that claims technological sovereignty as a structural right (Pillar 4) cannot ground its operational existence in infrastructure subject to US Patriot Act jurisdiction, to the discretionary account decisions of a single extractive corporation, and to a single geographic point of failure.

This annex presents a complete decentralized infrastructure architecture for PHREVO that resolves this contradiction. The architecture rests on five non-negotiable sovereignty principles: no central servers that can be shut down; end-to-end encryption where no relay node can read territory data without the territory's private key; blockchain as verification registry rather than data storage (only hashes and metadata on-chain, complete data in IPFS); guaranteed disaster recovery through minimum 3-node territorial replication; and distributed infrastructure governance requiring assembly approval for network software changes.

The implementation proceeds through three phases. Phase 1 (0-2 months): migrate from Google Cloud to a non-US cloud provider. Phase 2 (2-12 months): deploy 3-5

territorial nodes as network pilots. Phase 3 (12-24 months): complete network with no remaining central servers, all data in territorial nodes, rotary relay nodes operated exclusively by territories. Total infrastructure cost for 10 territories: $25,000-$85,000 — well within the funding target of Annex B's market entry strategy.

Contents

H.1 The Contradiction: Why Current Infrastructure Fails PHREVO's Own Principles

PHREVO's current infrastructure is a performative contradiction. A system that claims technological sovereignty as a structural right cannot ground its operation in Google Cloud us-west1.

The resolution is not optional. Technological sovereignty is not a desirable feature to be added later — it is the condition under which Pillar 4 has meaning, under which the Dignity Toolkit can be trusted by the communities it serves, and under which PHREVO's claim to be architecturally different from extractive systems has any substance.

H.2 Five Non-Negotiable Sovereignty Principles

H.3 Network Architecture: Territorial Nodes and P2P Topology

H.3.1 Territorial Node Model

Each territorial assembly registered in PHREVO operates a PHREVO node (or multiple nodes for redundancy). A PHREVO node is a server — physical or virtual — running the PHREVO software stack.

Node components:

IPFS instance: stores complete data for the territory and fragments of allied territories' data (encrypted).

Blockchain client: reads the public chain for verification; does not mine; participates in consensus only for network governance votes.

P2P API service: handles communication with other nodes via libp2p.

Local index database (SQLite or equivalent): stores local indices for fast querying without full IPFS traversal.

Node operator hierarchy:

Primary (preferred): The territorial assembly operates the node directly.

Delegated: A trusted organization designated by the assembly (e.g., a local technology cooperative, a university department).

Temporary (exception only): The PHREVO technical team operates a node for a territory that cannot yet maintain its own. This is explicitly temporary and must be disclosed publicly.

Operational responsibility: the assembly is responsible for maintaining uptime (power, connectivity, updates). A node inactive for more than 30 days is marked "potentially stale" by the network — other nodes reduce their trust in its data until it is restored.

H.3.2 Network Topology: Three Node Types

Critical design principle: there are no "master nodes." All territorial nodes are equal in governance weight (one vote per territory, regardless of node size, bandwidth, or historical contribution). Regional and relay nodes are technical instruments — they have no political authority within the PHREVO network.

H.3.3 Data Flow Diagram

The following describes the complete data flow for a territory publishing new data to the PHREVO network:

TERRITORY A (e.g., Buenaventura, Colombia)

Step 1 — LOCAL DATA GENERATION

Raw data: household surveys, sensor readings, assembly minutes

-> Digitally signed by assembly (ed25519 private key)

-> Encrypted with territory A public key

-> Result: encrypted_data_blob + signature

Step 2 — IPFS STORAGE (local node)

encrypted_data_blob -> IPFS add

-> Content hash: QmXyZ7a8b9c...

```
 -> Stored locally on territorial node
 -> Replication requested to 3 allied territorial nodes
 -> Allied nodes store encrypted blob (cannot decrypt — no private key)
Step 3 — BLOCKCHAIN REGISTRATION
 PHREVORecord {
   contentHash: "QmXyZ7a8b9c...",
   territory: "0xTerrAPublicKey...",
   timestamp: 1745823600,
   dataType: 1,  // 1=survey, 2=minutes, 3=sensor
   previousHash: "QmPrevious...",
   signature: "0xAssemblySignature..."
 }
 -> Submitted to PHREVO chain (Ethereum/Polkadot in Phase 1)
 -> Immutable record created: QmXyZ exists, created by Territory A
 -> NO encrypted content on-chain — only proof of existence
Security guarantees:
 - Attacker who captures allied nodes: gets encrypted blobs. Useless.
 - Attacker who captures relay node: gets routing metadata only. No data.
 - Attacker who controls blockchain: sees public metadata + hashes. No content.
```

H.4 Storage: IPFS + Blockchain

H.4.1 Complete Data in IPFS (Not On-Chain)

IPFS (InterPlanetary File System) is selected as the primary storage layer for four reasons: it is peer-to-peer with no required central server; content is addressed by its cryptographic hash (content-addressed, not location-addressed), ensuring that data can be retrieved from any node holding a copy; the network replicates content automatically as nodes request it; and it is open-source with a large production deployment base (used by Filecoin, Ethereum, Protocol Labs).

Data categories for IPFS storage:

Household survey data (anonymized or pseudonymized per the protocol in Annex F).

Territorial assembly minutes and resolutions.

AIoT sensor data (ecological, infrastructure monitoring).

Project documents (budgets, impact reports, audit records).

Territory public keys (for cross-territory verification).

Data explicitly NOT stored in IPFS:

Sensitive personal data (health records, immigration status) — these are never digitized in PHREVO or maintained only in fully local, network-isolated storage within the territory.

Private cryptographic keys — these never leave the territory's secure local environment.

H.4.2 Hashes and Metadata On-Chain

The blockchain layer serves as the immutable audit registry for the PHREVO network. It answers the question "does this data exist and was it created by this territory at this time?" — without revealing the data itself.

Blockchain selection:

Phase 1 (recommended): Existing public blockchain with smart contracts — Ethereum (Sepolia testnet in development, mainnet in production) or Polkadot. Low development cost, proven security, large validator network. Does not provide complete sovereignty (PHREVO depends on Ethereum's continued operation), but resolves immutability at acceptable cost.

Phase 2 (target): PHREVO native blockchain (hard fork of Cosmos SDK or Substrate). Full sovereignty — PHREVO controls the consensus rules, validator set, and upgrade path. Validators are territorial nodes. Development cost: high. Required when the network reaches 20+ territories.

Phase 1 transition to Phase 2: PHREVO records on the Ethereum chain are migrated to the native chain using a bridge contract. Historical records remain verifiable on Ethereum; new records go to the PHREVO chain.

H.4.3 Smart Contract Specification

```
// PHREVO Data Registry — Solidity (Ethereum Phase 1)
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;
contract PHREVORegistry {
  struct PHREVORecord {
      bytes32  contentHash;    // IPFS CIDv1 hash (SHA-256)
      address  territory;      // Territory public key (Ethereum address)
      uint256  timestamp;      // Block timestamp at registration
```

```
    uint8    dataType;      // 1=survey 2=minutes 3=sensor 4=project 5=audit
    bytes32  previousHash;  // Optional: links to previous record for integrity chain
    bytes    signature;     // ed25519 signature by assembly private key
  }
  mapping(bytes32 => PHREVORecord) public records;
  mapping(address => bool)         public registeredTerritories;
  address public governance;  // PHREVO Global Assembly multisig
  event RecordPublished(
    bytes32 indexed contentHash,
    address indexed territory,
    uint8 dataType,
    uint256 timestamp
  );
  modifier onlyRegistered() {
    require(registeredTerritories[msg.sender], "Territory not registered");
    _;
  }
  function publishRecord(
    bytes32 contentHash,
    uint8   dataType,
    bytes32 previousHash,
    bytes calldata signature
  ) external onlyRegistered {
    require(records[contentHash].timestamp == 0, "Hash already exists");
    records[contentHash] = PHREVORecord({
      contentHash:  contentHash,
      territory:    msg.sender,
      timestamp:    block.timestamp,
      dataType:     dataType,
      previousHash: previousHash,
      signature:    signature
    });
    emit RecordPublished(contentHash, msg.sender, dataType, block.timestamp);
  }
  function registerTerritory(address territory) external {
    require(msg.sender == governance, "Only governance can register");
    registeredTerritories[territory] = true;
  }
}
```

H.4.4 Replication and Redundancy Protocol

Each territory must replicate its data to at least 3 allied territorial nodes (not regional or relay nodes). Allied nodes are chosen by the assembly and rotated every 6 months to prevent concentration.

Replication sequence:

Territory A publishes new data: signs, encrypts, uploads to local IPFS node.

Territory A node sends a replication request to 3 allied nodes (B, C, D) via libp2p.

Allied nodes download the encrypted blob, verify the digital signature against the public key registered on-chain, and store locally.

Allied nodes return an acknowledgment including the IPFS hash and their own signature confirming receipt.

If an allied node does not respond within 48 hours, the system selects a replacement from the backup list and retries.
Territory A node records the successful replication in its local index.
Allied nodes store the encrypted blob permanently but cannot decrypt it — they hold ciphertext without the key. This means a government that seizes an allied node obtains zero readable data from Territory A.

H.5 Communication: libp2p and End-to-End Encryption

H.5.1 Communication Protocol Stack

PHREVO adopts libp2p as its P2P communication library — the same library used by IPFS, Ethereum, Filecoin, and Polkadot. This provides: maturity (production-tested at global scale), multi-transport support (TCP, WebRTC, WebSockets, QUIC), built-in peer discovery (DHT, PubSub, mDNS), and active maintenance by Protocol Labs.

H.5.2 Encryption and Authentication

Data at rest (in IPFS)

Each territory generates an asymmetric key pair: ed25519 (preferred for performance) or RSA-4096 (alternative for compatibility).

The private key resides only on the territorial node, in encrypted storage (passphrase-protected, hardware security module for high-security territories).

All data is encrypted with the territory's public key before IPFS upload. Only that territory can decrypt, even if the data is replicated across 50 nodes.

Exception: explicitly public data (assembly minutes, public resolutions, public PHREVO-Score reports) are signed but not encrypted — or encrypted with a PHREVO community public key.

Data in transit (between nodes)

All libp2p connections use the Noise Protocol for key exchange and TLS 1.3 for data encryption. Perfect forward secrecy (PFS) is mandatory — session keys are ephemeral.

Each node has a node identity key (separate from the territory data key) used for libp2p authentication.

All P2P messages include the sender node's digital signature to prevent impersonation.

H.5.3 Peer Discovery and Adversarial Connectivity

Bootstrap list: the PHREVO software ships with a hardcoded list of the first 5-10 founding territorial nodes. New nodes use this list to join the network on first launch. The bootstrap list is updated with each software release.

DHT (Kademlia): after connecting to bootstrap nodes, new nodes announce their presence in the distributed hash table. Other nodes can find them by querying the DHT.

Operating in hostile environments:

Offline / air-gap mode: for territories where internet connectivity is unreliable or dangerous, PHREVO nodes can operate disconnected for extended periods. Data is accumulated locally and synchronized via USB drive when offline nodes meet in person. This is a last resort but preserves operations under extreme censorship.

H.6 Network Governance: Rotary Relay Nodes

H.6.1 The Coordination Problem Without Center

Pure flat P2P networks face three structural challenges: peer discovery without a bootstrap server; routing efficiency for nodes with asymmetric connectivity (high-bandwidth nodes carry unfair load); and NAT traversal for nodes behind routers that block incoming connections.

The standard solution — dedicated bootstrap servers — reintroduces central points of failure and control. PHREVO's solution is rotary relay nodes: territorial nodes that volunteer to provide infrastructure services, selected by weighted lottery, with strictly limited and auditable authority.

H.6.2 Rotary Relay Nodes

Relay nodes provide: NAT hole-punching assistance for nodes that cannot accept incoming connections; peer discovery directory services; efficient routing for cross-regional connections. They do NOT: store data; read data content (only encrypted blobs pass through); have governance votes; persist across rotation cycles.

Eligibility and selection:

Any territory with reliable connectivity (>99% uptime over the previous month) and sufficient bandwidth (>50 Mbps) may volunteer its node as a relay.

The territorial assembly must vote to volunteer (simple majority). Volunteering is a service commitment, not a power grab.

A node can serve as relay for a maximum of 3 consecutive monthly periods before a mandatory rest. This prevents incumbency advantages.

Selection is by weighted lottery executed on the blockchain using a verifiable random function (Chainlink VRF or equivalent). Higher PHREVO-Score = higher weight, up to 3x, but no territory can hold more than 20% of relay node slots.

3 to 9 relay nodes operate simultaneously (scaling with network size).

H.6.3 Rotation Mechanism and Audit

Monthly rotation process:

At the end of each calendar month, the rotation smart contract executes automatically.

All territories that have posted a volunteer declaration on-chain and meet eligibility criteria are included in the lottery pool.

The VRF generates a verifiable random seed; the contract selects 3-9 nodes (depending on network size) according to weighted probabilities.

Selected relay nodes publish their libp2p multiaddress on-chain. All other nodes update their bootstrap lists within 24 hours.

Previous relay nodes enter their mandatory rest period.

Abuse detection and expulsion: any node can report suspected relay abuse (censoring specific nodes, modifying messages, refusing legitimate connections). If 3 independent nodes corroborate the report, an emergency assembly vote is called. A 60% majority can expel the relay node immediately and ban it from volunteering for 12 months. The expelled node's libp2p identity is added to a public blocklist distributed with the software.

H.7 Disaster Recovery and Resilience

H.7.1 Three-Copy Backup Protocol

Backup frequency: the PHREVO software generates an encrypted backup automatically every 7 days (configurable). The assembly can trigger a manual backup at any time. Physical offline backup should be updated at least monthly and stored in a different physical location from the primary node (different building, preferably different city).

H.7.2 Node Restoration Procedures

Scenario A: Node corrupted or destroyed, private key is intact

Acquire replacement hardware (or repair existing).
Install PHREVO software (download from official source; verify cryptographic signature of installer).
Import private key from offline backup (or from the secure custody of assembly key holders).
Request encrypted data from the 3 allied nodes that hold replicas.
Allied nodes transmit encrypted blobs; territory decrypts with private key.
Node synchronizes with blockchain for any records created during downtime.
Notify network of restoration (post on-chain announcement).

Scenario B: Node destroyed AND private key is lost — Total Loss

If the private key is permanently lost, encrypted data on allied nodes is permanently unreadable. This is not a bug — it is the security guarantee. There is no backdoor.

The territory generates a new key pair.
Historical data (from before the key loss) remains on the blockchain as immutable hashes — verifiable as having existed, but content unreadable by the new key.
The territory begins a new data cycle under the new key. The network records the key rotation event on-chain.
Physical source data (paper records, original sensor readings) may allow partial reconstruction if re-digitized.
Prevention is the only protection: assemblies must practice distributed key custody.
Recommended: Shamir's Secret Sharing, where the private key is split into 5 shares, any 3 of which can reconstruct it, held by 5 different trusted assembly members in 5 different physical locations.

H.7.3 Response to Attack and Censorship

H.8 Hardware Requirements and Costs

Total infrastructure cost for 10 territories (10 x Raspberry Pi nodes + regional node):

Hardware: 10 x $250 = $2,500 + 1 regional node $800 = $3,300.
Training and technical support: $10,000 - $30,000 (depending on technical capacity of territories).
Security audit of PHREVO node software: $10,000 - $50,000 (required before Phase 3).
Total: $25,000 - $85,000. Well within the grant budget targets in Annex B.

H.9 Implementation Roadmap: Three Phases

Immediate next step (this week): identify 3 territories from Annex B's priority list willing to serve as Phase 2 pilot node operators. Recommended starting candidates: the Dignity Toolkit in New York (but with the node physically located in Canada or Europe), and one Type 2 autonomous territory in Colombia (e.g., Resguardo Nasa) where the assembly has both the political commitment and the technical capacity to sustain a node.

H.10 Conclusion: Sovereignty as Condition, Not Option

Technological sovereignty is not an enhancement to be added when PHREVO is more established. It is the condition under which PHREVO's most fundamental commitments have meaning:

Pillar 4 (Autonomy as a structural right): autonomy claimed by a system that stores its data in Google Cloud us-west1 is fictional. The Patriot Act does not respect PHREVO's governance principles.

Dignity Toolkit: immigrant families who use the Dignity Toolkit because it does not expose their immigration status deserve to know that their data is not one judicial order away from US government access.

Theory of change (construction over rupture): a system that depends on a single corporate provider for its infrastructure has not built something durable. It has built something that can be turned off by one account closure.

The architecture described in this annex resolves these contradictions using proven, open-source, production-tested technologies. IPFS is used by millions of nodes. libp2p is the foundation of Ethereum. Asymmetric encryption is the standard for secure communication globally. None of this requires novel cryptographic assumptions or unproven protocols.

The challenge is not technical — it is organizational. The core work is: procuring affordable hardware, training territorial assembly members to operate nodes, building a culture of infrastructure stewardship alongside political governance. This is exactly the kind of work that PHREVO's community-centered model is designed to support.

Every territory that operates its own node is not only storing its data more safely. It is exercising Pillar 4 in the most concrete way possible — owning the infrastructure of its own economic governance. That is what sovereignty means in practice.

References

Benet, J. (2014). IPFS — Content Addressed, Versioned, P2P File System. arXiv:1407.3561.

Jimenez, A. (2026). PHREVO: A Post-Capitalist Economic Architecture for the Global South. SSRN Working Paper 6614438. DOI: 10.5281/zenodo.19666941.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. bitcoin.org.

Protocol Labs. (2022). libp2p Specification. github.com/libp2p/specs.

Shamir, A. (1979). How to Share a Secret. Communications of the ACM, 22(11), 612-613.

Szabo, N. (1994). Smart Contracts. nick.szabo.com.

Tor Project. (2023). Tor Design Document. spec.torproject.org.

Wood, G. (2016). Polkadot: Vision for a Heterogeneous Multi-Chain Framework. polkadot.network/whitepaper.

PHREVO Framework Paper — Annex H — Technological Sovereignty Architecture v1.0 — April 2026

Supplement to SSRN 6614438 — DOI: 10.5281/zenodo.19666941

Andres Jimenez — hello@phrevo.earth — phrevo.org