Namazu Chain

Proof of Tensor Generating Work for Verifiable Seismic and Structural Model Improvement

Hayato Kameta ZWEISPACE JAPAN / Goodhills Research

Technical Whitepaper v0.2 - 16 July 2026

Abstract

Namazu Chain is a proposed distributed system in which the primary economically rewarded work is the production of independently verifiable improvements to seismic, regional-propagation, and building-response models. The proposed mechanism, Proof of Tensor Generating Work (PoTGW), replaces a single global nonce race with a graph of canonical tensor tasks. A solver commits to a result before learning a deterministic held-out challenge set; independent validators reproduce the calculation, evaluate physical and statistical constraints, and issue a quorum-signed Verified Tensor Work Certificate. A separate post-quantum settlement layer orders certificates, challenges, reputation updates, and physical-state transitions. The design explicitly separates low-latency event communication from slower scientific verification and final ledger settlement. Since v0.1 of this paper, the project reports execution of a four-node local v0.5 engineering devnet with primary hash mining disabled: 30 finalized checkpoints, 20 valid work certificates, 11 task domains, and 24 passing acceptance tests, including rejection of false and copied claims and preservation of a simulated building-state transition. These results establish a reproducible synthetic-data protocol milestone, not scientific validation on real earthquake or structural data. This document specifies the architecture, formal objects, deterministic scoring approach, building-state history model, adversarial assumptions, implementation evidence, and evaluation plan. It does not claim a deployed earthquake early-warning service, structural-safety certification, scientific validation, or production-ready public consensus.

Contents

Document status. *This is a research and engineering whitepaper, not a peer-reviewed scientific publication. Equations define a protocol framework and testable hypotheses. Final seismic solvers, physical constraints, safety thresholds, and operational governance require domain-expert review and validation on recorded and controlled physical data. The v0.5 figures in Section 19 are project-reported local engineering results; this document package does not independently rerun the source repository.*

1 Introduction

Earthquake monitoring and structural response estimation combine two difficult requirements. The first is latency: useful information may need to move through a network in seconds or less. The second is accountability: measurements, models, corrections, and claims must remain inspectable after an event. Existing earthquake early-warning systems demonstrate that detection is not prediction and that achievable warning time depends strongly on source, sensor, user, and communication geometry (U.S. Geological Survey 2019; Allen et al. 2025). Smartphone and low-cost sensor research also shows that distributed accelerometers can contribute meaningful observations when calibration, triggering, aggregation, and false-positive control are treated seriously (Kong et al. 2016; Di Nuzzo et al. 2021).

Namazu Chain proposes a third requirement: computation performed by participating nodes should itself improve a shared physical-world model. Instead of attaching an arbitrary scientific calculation to a conventional hash lottery, PoTGW makes a verified model improvement the primary unit of work, reputation, and economic credit.

The central proposal is:

> *A node earns work credit by producing a new, independently reproducible and meaningfully improved tensor result for a canonical physical-data task.*

The term *tensor* is used here in an engineering sense broad enough to cover versioned coefficient arrays, transfer functions, state-space operators, modal representations, propagation operators, or other structured multi-dimensional models. The protocol does not assume that every accepted model must be a dense mathematical tensor or a neural network. It requires only a canonical representation, deterministic verification rules, and a scientifically defensible interpretation for the task class.

The design has four coupled components:

1. **Namazu Event Network:** a low-latency path for signed observations and progressive-confidence event messages.
2. **Tensor Work Graph:** a directed acyclic graph of local, regional, higher-aggregation, building, and floor-response tasks.
3. **PoTGW Verification Network:** commit, challenge, reveal, independent reproduction, dispute, and certification.
4. **Namazu Settlement Chain:** deterministic ordering of certificates, challenges, reputation updates, and physical-state transitions using post-quantum signatures.

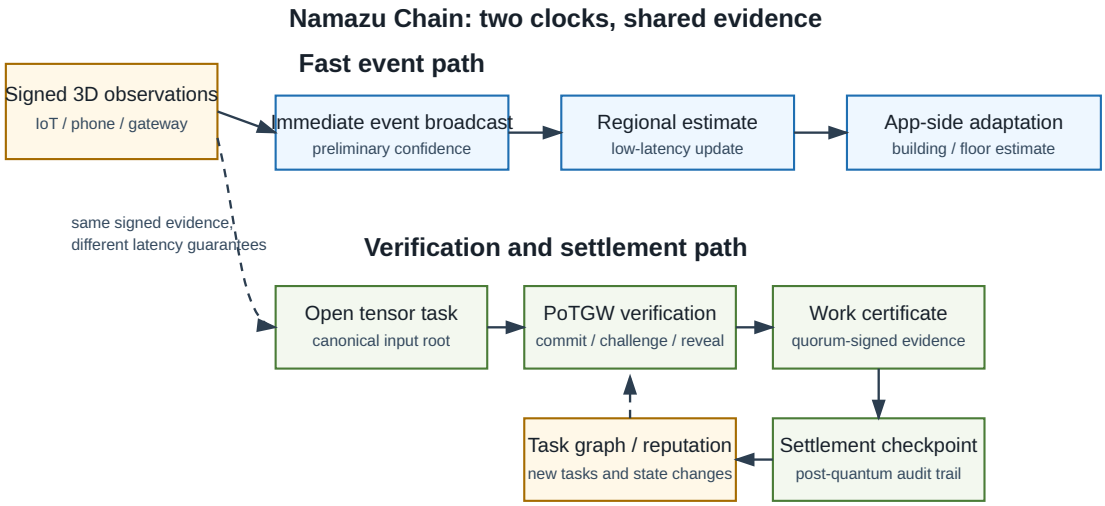

Figure 1: Namazu separates immediate event communication from later verification and settlement.

2 Scope, claims, and non-goals

2.1 What this whitepaper proposes

This whitepaper proposes:

- a canonical task identity for useful physical-model computation;
- a commit-challenge-reveal protocol intended to reduce copying and overfitting;
- a deterministic scoring framework that combines fit, held-out prediction, physical consistency, stability, and complexity;
- task-specific certificates rather than a globally comparable scientific score;
- bounded work credits that create proposer eligibility without a primary nonce race;
- domain-specific node reputation and append-only claim history;
- versioned building-state epochs that preserve formerly valid models after physical change;
- a separate low-latency event path that does not wait for block finality;
- a staged experimental plan from synthetic data to controlled and recorded accelerometer data.

2.2 What this whitepaper does not claim

This document does not claim that Namazu Chain:

- predicts earthquakes before rupture begins;
- is an official earthquake early-warning system;
- can currently determine whether a building is safe to occupy;
- can replace the Japan Meteorological Agency, USGS, NIED, local authorities, structural engineers, or emergency services;
- has been validated on a sufficient real-world earthquake corpus;
- has proven Byzantine security at public-network scale;
- has solved calibration, clock synchronization, privacy, data licensing, or safety certification;
- is currently connected to live NMZ token infrastructure, public sensors, production smartphones, or a deployed blockchain network.

The public warning boundary is particularly important. Established earthquake early-warning programs emphasize rapid detection after an earthquake begins, not prediction, and they warn that some recipients may receive information before, during, or after shaking arrives (U.S. Geological

Survey 2019). Namazu must inherit that scientific and communication discipline.

3 Design goals and research hypotheses

3.1 Design goals

G1 - Useful work dominance. Normal work credit must come from certified tensor improvement, not repeated nonce searching.

G2 - Task locality. Unrelated regions, buildings, time windows, and model classes must not be forced into one winner-takes-all competition.

G3 - Reproducibility. Consensus-critical inputs, arithmetic, scores, and encodings must be independently reproducible.

G4 - Scientific humility. A model mismatch must not automatically be classified as fraud. It may indicate physical-state change, sensor failure, insufficient evidence, or model-class failure.

G5 - Low-latency separation. Immediate event communication must not wait for settlement finality.

G6 - Attribution. Solvers and validators must sign claims, challenges, and certificates so that correct and incorrect work remains attributable.

G7 - Historical integrity. A formerly valid building model must remain valid for its historical state epoch even after the building changes.

G8 - Quantum-resistant authentication. The settlement and certificate layer should support standardized post-quantum signature schemes. NIST finalized ML-DSA and SLH-DSA in FIPS 204 and FIPS 205 (National Institute of Standards and Technology 2024a, 2024b).

G9 - Bounded influence. Work credits and reputation must be capped, domain-specific, and decaying so one operator cannot permanently monopolize ordering.

G10 - Falsifiability. Every central claim should map to an executable test or measurable experiment.

3.2 Research hypotheses

The initial implementation should test the following hypotheses rather than assume them:

- **H1:** Multiple nodes can independently produce byte-identical verification results using fixed-point arithmetic and canonical inputs.
- **H2:** Commit-challenge-reveal with delayed held-out selection reduces simple copying and training-set overfit.
- **H3:** A task graph allows independent work across regions and buildings without requiring a meaningless global accuracy comparison.
- **H4:** Bounded work credits plus quorum settlement can preserve ledger progress with primary nonce mining disabled.
- **H5:** Residual-distribution change can trigger a building-state investigation without retroactively invalidating an earlier correct model.
- **H6:** Domain-specific reputation classifies malicious claims more accurately than one global trust number.
- **H7:** A fast event path and a slower verification path can share signed evidence without coupling alert latency to chain finality.

4 System model

4.1 Participants

The protocol distinguishes logical roles. A physical node may perform several roles, but role separation is useful for reasoning and testing.

Role	Responsibility
Observation source	Produces signed, timestamped 3-axis acceleration or derived measurements.
Gateway	Aggregates, timestamps, filters, and forwards observations; may provide clock-quality metadata.
Task publisher	Opens a canonical work task using committed inputs and policy parameters.
Solver	Produces a candidate tensor or coefficient model.
Verifier	Independently reproduces consensus-critical measurements and signs an attestation.
Challenger	Submits evidence that a claim, certificate, input, or verification is invalid or misclassified.
Settlement validator	Orders certificates, disputes, reputation changes, and state transitions.
Application client	Consumes progressive event information and optionally applies a selected building/floor model.
Scientific auditor	Replays data and protocol decisions without necessarily participating in consensus.

4.2 Trust assumptions

The research devnet may initially assume four settlement validators with a 3-of-4 finalization quorum. This is an implementation milestone, not a final decentralization claim. The intended security model allows:

- faulty or malicious solvers;
- validators that lie, fail to reproduce calculations, or go offline;
- observation sources with calibration error, duplicated records, clock error, or fabricated data;
- temporary network partitions;
- copied candidate work;
- adaptive overfitting to visible validation data;
- building dynamics that genuinely change after an event.

The initial devnet does not attempt to solve fully permissionless Sybil resistance. Identity admission, hardware attestation, deposits, public validator rotation, and external scientific governance remain open design areas.

4.3 Two time domains

Namazu uses two time domains:

1. **Event time:** physical observation time, measured in sensor timestamps and short windows.
2. **Settlement time:** logical consensus rounds and finalized checkpoints.

These must not be conflated. A useful event message may be delivered before final settlement. A final certificate may arrive much later and may refine or challenge preliminary results.

5 Observation and data model

5.1 Canonical observation

A canonical acceleration observation is represented as:

$$o = (d, s, q, t, \Delta t, a_x, a_y, a_z, c, m, \sigma),$$

where:

- d is the device identity or privacy-preserving credential;
- s is a sensor identity and sensor class;
- q is a monotonically increasing sequence number;
- t is a timestamp in an agreed integer unit;
- Δt is the nominal sampling interval;
- (a_x, a_y, a_z) are fixed-point acceleration components;
- c is calibration and coordinate-frame metadata;
- m is quality metadata;
- σ is a digital signature.

Consensus-critical acceleration values should use a fixed integer unit, for example micro-gal or another explicitly selected scale. The exact scale is a protocol parameter and must be selected after overflow and noise analysis. Raw floating-point encodings are not valid consensus inputs.

5.2 Observation batches and roots

For a task window W, observations are canonically sorted by:

$$(extregion, \text{sensor class}, \text{device credential}, \text{sensor ID}, \text{timestamp}, \text{sequence}).$$

The ordered records are encoded with a versioned binary schema and committed using a Merkle root or equivalent authenticated structure:

$$R_{\text{input}} = \text{MerkleRoot}(\text{Encode}(o_1), \ldots, \text{Encode}(o_n)).$$

The root commits to ordering as well as content. Duplicate detection rules must be deterministic. The protocol must distinguish exact duplicates from legitimate overlapping samples.

5.3 Location and privacy

Exact smartphone locations and building identifiers can expose sensitive personal and commercial information. The protocol should support graduated disclosure:

- public coarse region or cell identity;
- encrypted or access-controlled precise location;
- rotating device credentials;
- separate public commitments and restricted raw-data stores;
- delayed publication for high-risk traces;
- user consent and deletion policies for off-chain personal data;

- an immutable record of consent and processing policy versions without placing personal data directly on-chain.

Immutability is not a justification for permanently publishing personal sensor traces.

6 Canonical tensor tasks

6.1 Task identity

Each useful-work task has a canonical Task ID:

$$T = H(v, e, r, W, \tau, \ell, h, b, \eta, R_{\text{input}}, A, L_{\text{prev}}, L_{\text{target}}, P),$$

where:

- v is the task schema version;
- e is an event or maintenance epoch;
- r is the geographic region or cell;
- W is the physical time window;
- τ is the tensor task class;
- ℓ is the virtual aggregation level;
- h is physical height or floor class when applicable;
- b is building identity when applicable;
- η is building-state epoch when applicable;
- R_{input} is the canonical input root;
- A is the algorithm family and version policy;
- L_{prev} is the currently accepted accuracy level;
- L_{target} is the requested next level;
- P is the task policy and scoring parameter root.

The protocol must use distinct fields for **virtual aggregation level** and **physical height**. This prevents ambiguity between a regional layer and the second or third floor of a structure.

6.2 Task classes

Initial task classes may include:

- `GROUND_LOCAL`: fit a local ground-motion representation;
- `GROUND_REGIONAL`: combine certified local representations into a wider-area operator;
- `VIRTUAL_POINT`: create a certified higher-aggregation observation point;
- `BUILDING_BASELINE`: identify a building response model for a state epoch;
- `BUILDING_EVENT`: update or test a building model during an event;
- `FLOOR_RESPONSE`: estimate floor-specific response from regional ground input and a building model;
- `SENSOR_CALIBRATION`: estimate calibration corrections and uncertainty;
- `CROSS_VALIDATION`: independently test an accepted tensor on later or held-out data;
- `STATE_TRANSITION`: evaluate evidence for a changed physical state.

6.3 Task locality and non-comparability

Only candidates for the same canonical Task ID directly compete or supersede one another. A Tokyo local-ground tensor and an Osaka building tensor are not ranked by one global scientific score. They can both earn normalized, bounded work credits within their task classes.

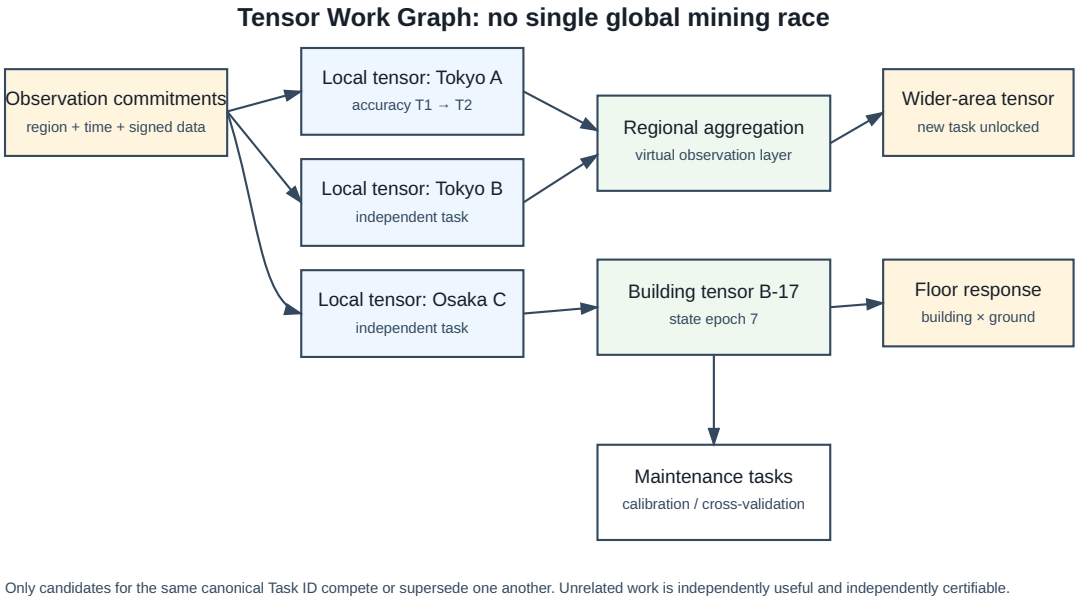

Figure 2: The tensor work graph supports parallel useful work.

7 Model and scoring framework

7.1 Model-agnostic prediction interface

For task class τ and algorithm version A, a candidate model is a canonical parameter object θ. It defines a deterministic prediction function:

$$\widehat{y}_{i,k} = F_{\tau,A}(\theta; p_i, t_k, u_k, z_i),$$

where p_i is position, t_k is time, u_k represents parent tensors or ground input, and z_i contains allowed sensor/building metadata.

The protocol does not prescribe one universal F. A local wavefield basis, transfer operator, modal model, finite impulse response, state-space model, or carefully bounded machine-learning model may be valid for a specific versioned task class. Every accepted algorithm must specify deterministic execution, input availability, parameter encoding, and verification cost.

7.2 Loss components

A candidate is evaluated using integer or rational fixed-point terms:

$$J(\theta) = \alpha E_{\text{obs}} + \beta E_{\text{hold}} + \gamma P_{\text{phys}} + \delta P_{\text{region}} + \kappa P_{\text{stability}} + \lambda C_{\text{model}}.$$

Terms are defined per task class:

- E_{obs}: error on the disclosed fitting set;
- E_{hold}: error on a challenge-selected held-out set;
- P_{phys}: physical-consistency penalty;
- P_{region}: inconsistency with certified neighboring or parent models;
- $P_{\text{stability}}$: sensitivity to small perturbations or repeated windows;
- C_{model}: bounded complexity penalty.

Lower J is better. No term may depend on local wall-clock performance or non-deterministic library behavior.

7.3 Observed and held-out errors

For weighted three-axis residuals $r_{i,k} = y_{i,k} - \widehat{y}_{i,k}$, one deterministic error form is:

$$E = \left\lfloor \frac{1}{Z} \sum_{i,k} w_{i,k} \left(|r^x_{i,k}|^2 + |r^y_{i,k}|^2 + |r^z_{i,k}|^2 \right) \right\rfloor ,$$

where weights and normalization Z are canonical integers. A square root is unnecessary for consensus; display software may convert the squared measure to familiar units.

The held-out set is selected only after commitment. Selection uses finalized chain randomness or a threshold signature output that the solver could not predict at commit time.

7.4 Physical-consistency penalties

Physical penalties must be specific to a task class and scientifically reviewed. Candidate examples include:

- causality and allowed propagation-delay bounds;
- bounded apparent wave speed for a declared medium/model class;
- continuity constraints across neighboring cells;
- coordinate-frame and orientation consistency;
- energy or amplitude bounds appropriate to the representation;
- stable modal parameters within a declared operating regime;
- constraints against impossible floor-to-floor discontinuities;
- required uncertainty inflation when inputs are sparse or low quality.

These are examples, not validated Namazu thresholds. A poor physical constraint can be as dangerous as no constraint because it can reject real phenomena or accept an unrealistic model.

7.5 Meaningful improvement

Let J_0 be the score of the accepted baseline and J_1 the candidate score. The improvement is:

$$\Delta J = J_0 - J_1.$$

A candidate is not accepted merely because $\Delta J > 0$. It must satisfy:

$$\Delta J \geq \Delta_{\min}(\tau, L_{\text{target}}),$$

and all mandatory component thresholds. This prevents tiny numerical improvements from continuously generating rewards.

7.6 Accuracy levels

PoTGW uses versioned accuracy levels rather than an unqualified claim of “one more digit.”

Level	Minimum interpretation	Example requirements
T0	Coarse candidate	Valid encoding and bounded observed error.
T1	Basic local fit	Improvement over baseline and minimum coverage.
T2	Held-out validated	Acceptable challenge-set error and stability.
T3	Physically constrained	T2 plus task-specific physical consistency.
T4	Aggregated/parent validated	Certified dependency graph and regional consistency.
T5	Building/floor validated	Versioned building state, uncertainty, and controlled validation evidence.

The levels are policy labels, not universal scientific truth. Thresholds differ by task class, sensor quality, event type, and evidence regime.

8 Proof of Tensor Generating Work protocol

8.1 Protocol phases

The primary claim protocol has six phases.

8.1.1 Open

A task publisher finalizes the Task ID, input-data root, baseline certificate, algorithm policy, target level, scoring policy, deadlines, and data-availability commitments.

8.1.2 Commit

Solver n computes candidate θ and submits:

$$C = H(T, n, H(\text{Encode}(\theta)), R_{\text{evidence}}, q, \rho),$$

where q is a solver sequence and ρ is a secret salt. The solver signs the commitment. The commitment binds the solver to the candidate before challenge selection.

8.1.3 Challenge

After the commitment is final, the network derives a challenge seed:

$$s_c = H(R_{\text{finalized}}, T, C, \text{round}).$$

The seed deterministically selects held-out sensors, time intervals, perturbation checks, or requested evidence branches.

8.1.4 Reveal

The solver reveals θ, ρ, evidence branches, declared resource metadata, and any task-specific proof objects. The network verifies that the reveal matches C.

8.1.5 Verify

Independent nodes:

- retrieve and authenticate the underlying data;
- reproduce canonical decoding and ordering;
- recompute predictions and score components;
- test challenge-selected held-out data;
- verify physical and regional constraints;
- verify solver signatures and work ownership;
- return signed accept, reject, or classification attestations.

8.1.6 Certify and finalize

A sufficient verification quorum creates a Verified Tensor Work Certificate. Settlement validators order it into a checkpoint with related challenges, verdicts, reputation changes, and newly unlocked tasks.

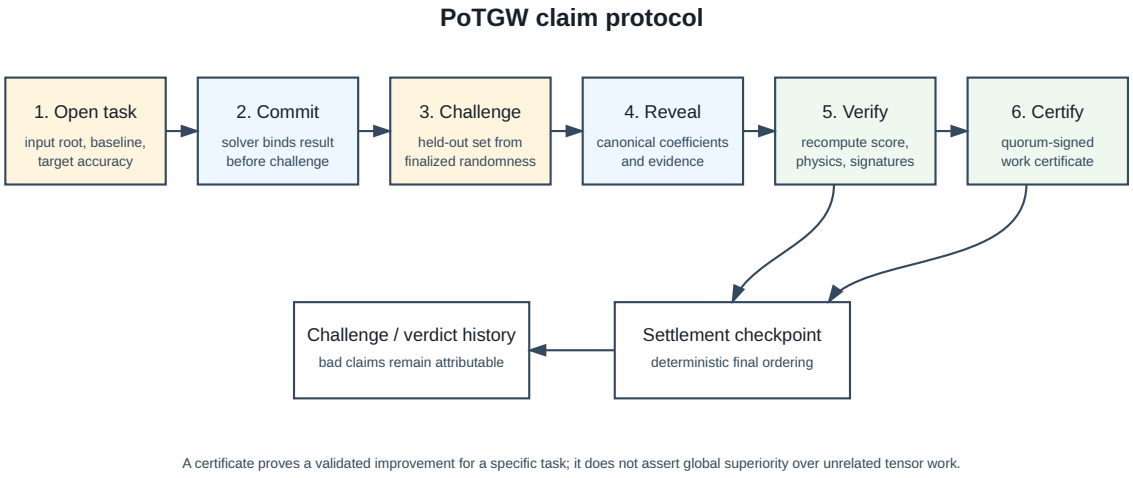

Figure 3: Commit, challenge, reveal, verification, and certification.

8.2 Why commit-challenge-reveal is necessary

Useful-work consensus has a recurring difficulty: a network must prove not merely that arithmetic occurred, but that the work is connected to a real task, is not copied, and provides the claimed improvement. Prior PoUW literature identifies verification, task availability, and preservation of consensus security as central problems (Hoffmann 2022; Dong, Lee, and Zomaya 2019).

Commitment before challenge selection reduces three simple attacks:

1. fitting only the visible validation subset;
2. copying a revealed result and claiming independent authorship;
3. changing coefficients after learning the challenge.

It does not solve every form of plagiarism or collusion. Timing, encrypted mempools, solver enclaves, zero-knowledge techniques, and economic deposits may be considered later.

9 Verified Tensor Work Certificate

A certificate is a canonical, quorum-signed statement about one task and one candidate. A minimum schema includes:

- certificate schema version;
- Task ID;
- parent task and certificate IDs;
- solver node ID and signature;
- commitment ID and reveal ID;
- algorithm family and version;
- canonical input-data root;
- coefficient or model payload root;
- observed error;
- held-out error;
- physical-consistency penalty;
- regional-consistency penalty;
- stability penalty;
- model-complexity penalty;
- baseline score and candidate score;
- accepted accuracy level transition;
- uncertainty and evidence class;
- verifier IDs, decisions, and signatures;
- challenge seed and challenge policy version;
- logical round and expiry conditions;
- certificate ID.

A certificate means:

> *Under the named data, algorithm, policy, and evidence conditions, a quorum reproduced the candidate and accepted the declared level transition.*

It does **not** mean:

- the model is globally optimal;
- the building is safe;
- the event estimate is an official warning;
- the tensor remains valid after a physical-state change;
- the task is scientifically valuable outside its declared scope.

10 Tensor Work Graph

10.1 Dependency edges

A certificate may unlock new tasks. For example:

$$\text{local tensors} \rightarrow \text{regional tensor} \rightarrow \text{building response} \rightarrow \text{floor estimate}.$$

Each edge is explicit. A child task commits to parent certificate IDs, not only parent coefficient payloads. This creates provenance for the final result.

10.2 Parallel work

Nodes can perform useful work on:

- separate geographic cells;
- separate event windows;
- distinct buildings;
- calibration tasks;
- challenge verification;
- regional aggregation;
- historical refinement;
- model-class comparison.

No single node needs to "win the earthquake." The network benefits from multiple independently certified results.

10.3 Supersession

Candidate c_2 can supersede c_1 only when:

- both address the same Task ID or an explicitly compatible successor task;
- c_2 satisfies a higher level or a material score improvement;
- data and policy compatibility are proven;
- the settlement record references the superseded certificate;
- historical validity is preserved.

Supersession is not deletion.

11 Settlement consensus without primary nonce mining

11.1 Separation of useful work and ordering

PoTGW determines which work is useful and who earned bounded work credit. Settlement consensus determines the final order of records. These are related but not identical functions.

The v0.5 research devnet should use a simple 3-of-4 Byzantine-fault-tolerant finalization rule. Mature Byzantine consensus has a large literature; the Namazu prototype should not claim a new BFT proof merely because its proposer eligibility originates from useful work (Castro and Liskov 1999).

11.2 Work credits

A certificate creates bounded credits:

$$K_{n,d}(r+1) = \min(K_{\max,d},\ \mu K_{n,d}(r) + g(c) - p(n,d,r)),$$

where:

- $K_{n,d}$ is node n's credit in domain d;
- $\mu < 1$ is a decay factor;
- $g(c)$ is a capped grant from certificate c;
- p is a penalty from confirmed invalid behavior;
- $K_{\max,d}$ caps domain influence.

Correct independent verification may create smaller credits than solving. Credits are normalized by task class and are not assertions that one scientific task is globally more important than another.

11.3 Proposer eligibility

A deterministic schedule may select settlement proposers from nodes satisfying:

- minimum recent certified work or verification participation;
- no active severe penalty or suspension;
- per-domain and per-operator caps;
- liveness requirements;
- deterministic rotation based on the previous finalized checkpoint.

A proposer cannot finalize its own checkpoint alone. Finality requires the settlement quorum.

11.4 No-work periods

The preferred no-event tasks are useful maintenance work:

- historical cross-validation;
- sensor calibration;
- building baseline refresh;
- regional model refinement;
- data-availability audits;
- reproducibility checks;
- synthetic stress tests clearly labeled as synthetic.

When no useful certificate is ready, validators may produce low-cost heartbeat checkpoints. These maintain time and membership without pretending to be tensor work.

11.5 Emergency hash fallback

A minimal hash fallback is optional and disabled by default. If implemented, it must:

- be invoked only under explicit liveness policy;
- be marked `FALLBACK_LIVENESS`;
- create no normal PoTGW work credit;
- use a separately testable path;
- never replace or override a valid tensor certificate;
- avoid creating an economic ecosystem centered on hash power.

The main acceptance test must progress with hash mining completely disabled.

12 Node reputation and claim history

12.1 Domain-specific reputation

One global trust score is misleading. Reputation is keyed by a domain tuple such as:

$$d = (r, \tau, s, A, b_c),$$

where r is region, τ tensor type, s sensor class, A algorithm family, and b_c building class.

A node may have strong history for Tokyo local-ground tasks and no evidence for high-rise structural models. Applications and validators should inspect the relevant domain history.

12.2 Append-only statuses

Claim history includes at least:

- `VERIFIED`;
- `CHALLENGED`;
- `INVALID_CALCULATION`;
- `FALSE_INPUT`;
- `INVALID_SIGNATURE`;
- `COPIED_WORK_ATTEMPT`;
- `SUPERSEDED_BY_BETTER_MODEL`;
- `PHYSICAL_STATE_CHANGED`;
- `SENSOR_FAILURE_SUSPECTED`;
- `INSUFFICIENT_DATA`.

The status is a classification backed by evidence and signatures. It is not an edit to the original claim.

12.3 Penalties

Penalties should be proportional and reversible only through a transparent appeal process. Examples:

- invalid signature: immediate rejection, potential identity investigation;
- deterministic calculation mismatch: rejection and limited domain penalty;
- repeated false input: severe domain penalty or suspension;
- copied-work attempt: penalty after proof of prior binding commitment;
- good-faith model supersession: no dishonesty penalty;
- physical-state change: no solver penalty for a model that was valid in the earlier epoch;
- inconclusive evidence: no fraud classification.

13 Building tensors and physical-state epochs

13.1 Response model

For building b, state epoch η, and floor or height class f, a generic response model is:

$$\widehat{x}_{b,\eta,f}(t) = \mathcal{H}_{b,\eta,f,\theta}\left[g_r(t)\right] + \epsilon(t),$$

where $g_r(t)$ is the regional or ground-level input and $\mathcal{H}$ is a versioned linear or nonlinear operator. For a linear time-invariant research model, this may be a convolution:

$$\widehat{x}_{b,\eta,f}(t) = \sum_{j=0}^{m} H_{b,\eta,f}[j] g_r(t-j).$$

The final system may use modal, state-space, nonlinear, or hybrid models. Structural monitoring research supports the use of vibration measurements and modal analysis, including low-cost MEMS sensor approaches, but operational damage assessment requires substantially more evidence than a generic residual threshold (Michel et al. 2007; Di Nuzzo et al. 2021; Roohi, Hernandez, and Rosowsky 2020).

13.2 State-change trigger

Let residuals under the accepted epoch-η model have reference distribution P_η. New observations produce Q. A deterministic change statistic $D(P_\eta, Q)$ may trigger investigation when:

$$D(P_\eta, Q) > \tau_{\text{transition}}.$$

The trigger is not a damage declaration. It opens a `STATE_TRANSITION` task.

13.3 Classification

Validators consider competing explanations:

- original model or calculation was invalid;
- the building changed physically;
- sensors moved, failed, saturated, or changed calibration;
- input ground-motion estimate was wrong;
- evidence is insufficient;
- the model class is no longer appropriate.

A transition certificate should include evidence roots, alternative-hypothesis scores, uncertainty, and required human review flags.

13.4 Historical preservation

If tensor v7 was valid for state epoch 7, an earthquake may open state epoch 8 and lead to tensor v8. The chain records:

1. v7 and its original evidence;
2. the divergence event;
3. challenges and alternative explanations;
4. the state-transition verdict;
5. v8 and its new evidence.

It does not rewrite v7 as false solely because v8 is now required.

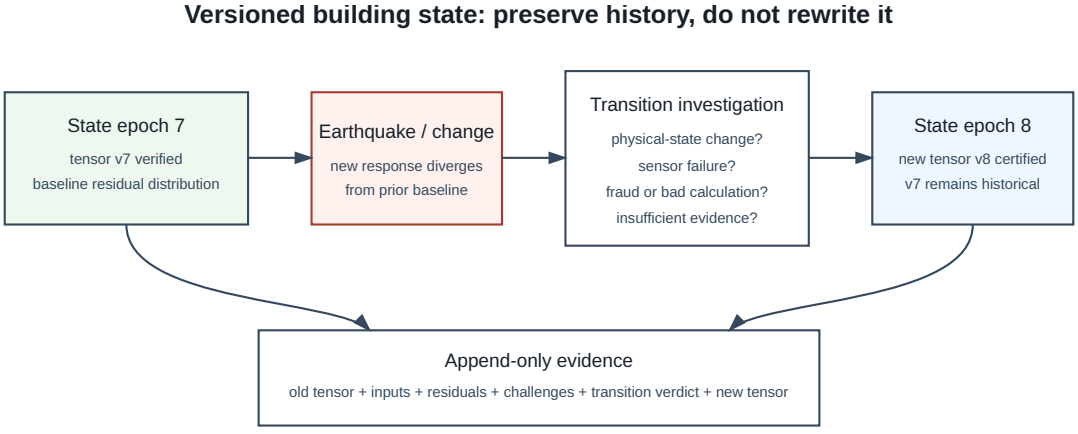

Figure 4: Building tensors are versioned by physical-state epoch.

14 Namazu Event Network

14.1 Immediate path

The immediate path is designed for progressive information:

1. a sensor or phone detects unusual motion;
2. a signed local event message is broadcast;
3. nearby gateways or nodes seek multi-sensor confirmation;
4. preliminary region and intensity estimates are updated;
5. an application optionally applies a selected building/floor model;
6. confidence and provenance are displayed;
7. later PoTGW certificates refine and audit the event.

The event path must not block on a settlement round, certificate quorum, or data publication deadline.

14.2 Progressive confidence states

An application may distinguish:

- `LOCAL_DETECTION` - one or more local triggers;
- `MULTI_SENSOR_PRELIMINARY` - independent nearby confirmation;
- `REGIONAL_ESTIMATE` - an initial regional propagation estimate;
- `BUILDING_ADAPTED` - a selected building/floor model has been applied;
- `NETWORK_VERIFIED` - a PoTGW certificate exists;
- `SETTLEMENT_FINALIZED` - the certificate is in a finalized checkpoint;
- `CHALLENGED` - later evidence disputes the prior result.

These labels must not be translated into official safety instructions without regulatory and scientific approval.

14.3 Latency measurements

The implementation should measure separately:

- sensor-to-gateway latency;
- gateway-to-peer latency;
- preliminary multi-sensor confirmation latency;
- regional estimate latency;
- app adaptation latency;
- commitment, challenge, verification, certificate, and settlement latency.

A single “block time” is not an adequate measure of system usefulness.

15 Deterministic computation

15.1 Arithmetic

Consensus code must specify:

- integer units and ranges;
- signed overflow behavior;
- rounding mode for division;
- canonical matrix and tensor ordering;
- deterministic linear algebra routines;
- treatment of missing values;
- solver convergence limits;
- coefficient quantization;
- score saturation and clipping;
- canonical serialization.

If a solver uses floating-point, machine learning, GPUs, or a non-deterministic optimizer to discover a candidate, that is acceptable only if validators can verify the final candidate using deterministic consensus routines.

15.2 Discovery versus verification

PoTGW may deliberately separate expensive discovery from bounded verification:

- **Discovery:** solvers may use optimized numerical libraries, GPUs, search, or domain-specific methods.
- **Verification:** all validators execute a deterministic, bounded evaluator on the revealed model and committed evidence.

This asymmetry is desirable, but verification must still establish usefulness rather than only arithmetic correctness.

15.3 Algorithm governance

Every algorithm family requires:

- a stable version identifier;
- reference implementation;
- test vectors;
- complexity bounds;
- input and output schema;
- deterministic verifier;
- known failure modes;
- deprecation and migration policy;
- independent scientific review before safety-sensitive use.

16 Threat model and adversarial analysis

16.1 False accuracy claims

A solver may publish coefficients with a fabricated accuracy level. Independent recomputation rejects the claim. Repeated or deliberate behavior reduces domain reputation.

16.2 Training overfit

A solver may optimize only the public fit set. Delayed challenge selection and held-out measurements reduce this attack. The network should also use later time windows and cross-region validation where appropriate.

16.3 Copied work

A node may copy a revealed candidate. Binding commitments, solver signatures, reveal timing, and certificate ownership prevent simple reassignment. Collusion before commitment remains an open problem.

16.4 Fake usefulness

A computation can be mathematically correct but unrelated to a genuine physical task. Each task therefore commits to authenticated input roots, task policy, data-availability evidence, and downstream dependency rules. The network must verify the connection between computation and declared task, a problem emphasized in the broader PoUW literature (Hoffmann 2022).

16.5 Sensor fabrication and Sybil observations

Many fabricated sensors can create a false physical model. Mitigations may include:

- device credentials and calibration history;
- diverse operator and hardware classes;
- geographic plausibility;
- radio/network diversity;
- comparison with trusted reference stations;
- bounded contribution per device/operator;
- anomaly and collusion detection;
- deposits or legal identity for high-trust feeds;
- explicit evidence classes rather than one undifferentiated dataset.

16.6 Time and location manipulation

Clock drift, delayed replay, GPS spoofing, and location fabrication can corrupt propagation models. Records must carry clock-quality metadata, sequence numbers, replay protection, and cross-source consistency checks.

16.7 Validator collusion

A 3-of-4 research devnet can finalize a false certificate if three validators collude. The devnet demonstrates software behavior, not public-network security. Future designs need larger, diverse validator sets, governance, independent data auditors, and formal security analysis.

16.8 Denial of service

Attackers may create expensive tasks or floods of observations. Task admission, verification cost limits, data-size bounds, rate limits, and fee/deposit policy are required.

16.9 Reputation poisoning

Attackers may challenge honest nodes to overload them or manipulate reputation. Challenges need evidence commitments, bounded costs, and penalties for provably abusive behavior.

16.10 Physical-state ambiguity

A mismatch can arise from damage, sensor movement, input error, or model failure. The protocol must represent uncertainty and alternative classifications rather than forcing a binary honest/dishonest result.

17 Post-quantum authentication

Namazu's certificate and settlement objects should be signed using standardized post-quantum algorithms or a migration-ready hybrid policy. FIPS 204 specifies ML-DSA, and FIPS 205 specifies SLH-DSA (National Institute of Standards and Technology 2024a, 2024b).

The engineering prototype should define:

- allowed signature suites by protocol version;
- key epoch and rotation rules;
- certificate and checkpoint signature domains;
- key compromise and revocation records;
- hybrid classical/post-quantum transition policy if used;

- maximum object sizes and verification costs;
- deterministic signature-policy upgrades.

Use of a standardized algorithm does not by itself make an implementation secure or certified. Key generation, random-number generation, side-channel resistance, library quality, and operational key management remain essential.

18 Incentives and economics

This whitepaper intentionally avoids fixing token emission or reward values. The mechanism should first demonstrate that useful work can be identified, verified, and ordered without primary hash mining.

Economic design principles are:

- reward certified work, not unverifiable effort;
- reward independent verification at a lower bounded rate;
- do not reward rejected or duplicate work;
- cap rewards by task class and epoch;
- avoid incentives to produce meaningless micro-improvements;
- reserve credits for maintenance and data-availability work;
- penalize proven false input, copied work, and dishonest attestation;
- avoid penalizing legitimate model supersession or physical-state change;
- separate scientific value from market price.

Before public economics, the project needs attack simulations covering collusion, bribery, task monopolization, withheld results, validation cartels, and strategic sensor manipulation.

19 Engineering implementation status

19.1 v0.4 single-machine precursor

The project reports that the local v0.4 Java prototype executed the following single-machine research path:

- generated 80 synthetic 3D acceleration records;
- fit them into 8 coefficients;
- obtained reported RMSE of 0.422141 gal under a 1.750000-gal research threshold;
- committed sensor, coefficient, aggregation-layer, and building-result roots;
- mined a nonce of 187858 to a `0000`-prefix research target;
- produced a local building result labeled `DEMO_CAUTION`;
- verified the locally constructed proof block.

The v0.4 experiment established a calculation, commitment, and local verification path. Its nonce mechanism was subsequently removed from normal progress in the PoTGW-first v0.5 design.

19.2 v0.5 PoTGW-first local devnet

The project reports implementation of `v0.5-namazu-potgw-consensus` as a four-process Java devnet on one local Windows machine. The implementation includes:

- four separately running node processes with distinct ports, data directories, logs, identities, and ML-DSA-65 keys;
- canonical tensor tasks and dependency unlocks in a Tensor Work Graph;

- commit, deterministic challenge, reveal, independent verification, quorum verdict, and certificate creation;
- settlement checkpoints whose normal proposer eligibility derives from verified work credits rather than nonce mining;
- domain-specific reputation events;
- a simulated building-state transition from epoch 7 to epoch 8;
- machine-readable task, certificate, reputation, transition, and checkpoint outputs.

The reported main run was invoked with:

```
.\RUN-V0.5-DEVNET.cmd
.\VERIFY-V0.5-DEVNET.cmd
```

19.3 Reported v0.5 execution results

Table 1 records the result summary supplied by the project after the local run.

Measure	Reported result
Separate node processes	4
Finalized settlement checkpoints	30
Valid Verified Tensor Work Certificates	20
Distinct task domains	11
Acceptance tests passed	24
Acceptance tests failed	0
Primary hash mining in successful main run	Disabled
All four nodes reached the same finalized tip	Pass
Deliberately false claim rejected	Pass
Copied-work attempt rejected	Pass
Building state epoch 7 to epoch 8 preserved without rewriting epoch 7	Pass

The run therefore satisfies the principal v0.5 engineering criterion:

> *Four independent node processes solved and verified multiple tensor tasks, rejected false and copied claims, preserved a legitimate simulated building-state transition, and converged on one finalized certificate-chain tip with primary hash mining disabled.*

19.4 Interpretation and evidence boundary

The v0.5 result is materially stronger than the v0.4 concept proof because normal ledger progress is reported to occur through PoTGW certificates and verified-work eligibility rather than a nonce race. It demonstrates an executable protocol architecture for task-local useful work, certificate ordering, claim history, and deterministic convergence under the implemented synthetic scenarios.

It does **not** yet establish:

- scientifically valid seismic or structural tensor models;
- correct performance on recorded earthquake accelerograms;
- operation across independent physical machines or administrative domains;
- public-network Byzantine security;
- resistance to large-scale Sybil, cartel, bribery, denial-of-service, or data-poisoning attacks;
- calibrated building-damage classification;

- operational earthquake early warning;
- production readiness or safety certification.

For a repository release, the reported claims should be accompanied by immutable evidence copied from the executed v0.5 directory, including at minimum:

- `test-output/SUMMARY.txt`;
- `test-output/certificates.json`;
- `test-output/finalized-checkpoints.json`;
- the complete acceptance-test report;
- the deterministic genesis and node configuration files;
- hashes of the source tree and evidence artifacts;
- the exact shared finalized tip hash;
- documented runtime, Java version, operating system, and commands.

The present whitepaper revision records the project-supplied summary but does not independently inspect or rerun those local files.

20 Evaluation methodology

20.1 Phase A - Deterministic synthetic tests

Goals:

- verify canonical encoding and fixed-point arithmetic;
- reproduce task IDs and scores across machines;
- test every rejection status;
- test forks, partitions, and recovery;
- validate work-credit and proposer caps;
- confirm that unrelated tasks progress in parallel.

Synthetic data is appropriate for protocol correctness but not for scientific performance claims.

20.2 Phase B - Recorded open seismic datasets

Goals:

- replay recorded acceleration windows;
- measure fit and held-out prediction;
- compare task classes and algorithm versions;
- evaluate sensitivity to missing sensors, clock error, and noise;
- publish reproducible test manifests and hashes.

Dataset licensing and provenance must be recorded.

20.3 Phase C - Controlled physical experiments

Goals:

- use calibrated accelerometers on known structures or shake-table experiments;
- introduce controlled sensor movement and calibration faults;
- test state-transition classification;
- compare model estimates with independent instrumentation;
- involve structural and seismological reviewers.

20.4 Phase D - Shadow network

A non-alerting shadow deployment may ingest live or near-live data without issuing public safety messages. It measures latency, availability, false triggers, and model drift while official systems remain authoritative.

20.5 Phase E - Governed pilot

Any safety-relevant pilot requires:

- responsible public agency and scientific partners;
- defined liability and human oversight;
- official-message compatibility;
- accessibility and public education;
- incident response and audit procedures;
- independent security review;
- explicit limits on app language and automated actions.

21 Metrics

Protocol metrics include:

- deterministic replay success rate;
- certificate acceptance and false-accept rate;
- false-reject rate by task class;
- held-out improvement distribution;
- verification cost versus discovery cost;
- task completion latency;
- settlement finality latency;
- partition recovery time;
- proposer concentration;
- domain reputation calibration;
- data-availability failure rate;
- copied-work detection rate.

Scientific metrics include:

- observation and held-out residuals;
- uncertainty calibration;
- propagation timing error;
- regional consistency;
- building response prediction error by floor;
- change-point detection delay;
- classification accuracy among damage, sensor fault, input error, and model failure;
- sensitivity to sensor density and quality.

Safety communication metrics include:

- event-message latency;
- confidence-state transitions;
- alert arrival relative to shaking;
- false and missed preliminary detections;
- user comprehension;
- behavior under contradictory official and experimental messages.

22 Governance and scientific review

A physical-world consensus system cannot rely only on code governance. Protocol changes may alter model assumptions, accepted sensors, or safety interpretation. A mature design needs separate review domains:

- cryptography and distributed systems;
- seismology and ground-motion modeling;
- structural engineering;
- sensor calibration and metrology;
- privacy and data protection;
- public warning and risk communication;
- economics and anti-manipulation;
- accessibility and emergency management.

Algorithm activation should require published test vectors, independent review, a delayed activation window, rollback policy, and explicit compatibility rules for old certificates.

23 Open research questions

1. What tensor or operator families provide the best balance of physical meaning, predictive value, compactness, and deterministic verification?
2. How should challenge sets be selected when data access is delayed, private, or geographically sparse?
3. How can the network distinguish legitimate parallel discovery from copied work or private collusion?
4. What is the correct Sybil-resistance mechanism for observation sources, solvers, and settlement validators?
5. How should task-class work credits be normalized without pretending that unrelated scientific tasks share one value scale?
6. Which physical constraints improve robustness without suppressing unexpected but real wave phenomena?
7. How should uncertainty propagate from sensors through regional tensors into building and floor estimates?
8. How can structural-state change be classified without implying an automated safety diagnosis?
9. What raw data can be public, restricted, delayed, aggregated, or deleted while maintaining an auditable commitment history?
10. Can zero-knowledge or verifiable-computation methods reduce disclosure while preserving independent verification?
11. How should a fast event network behave during internet congestion, power loss, malicious floods, or contradictory sensor clusters?
12. What governance structure can integrate official agencies and scientific review without centralizing every useful-work task?

24 Roadmap

24.1 v0.5 - PoTGW-first local devnet (engineering milestone completed)

Project-reported local execution now includes four separate node processes, deterministic task and certificate objects, commit-challenge-reveal verification, domain reputation, a simulated building-state transition, 3-of-4-style settlement, and normal progress with primary hash mining disabled. Remaining v0.5 publication work is to preserve complete evidence, publish reproducibility hashes, and obtain an independent rerun.

24.2 v0.6 - Recorded-data replay

- recorded accelerometer datasets;
- reproducible manifests;
- baseline algorithm comparisons;
- hold-out and perturbation tests;
- scientific error reporting.

24.3 v0.7 - Controlled sensor and building experiment

- calibrated hardware;
- synchronized data acquisition;
- controlled changes and faults;
- independent reference instrumentation;
- structural engineering review.

24.4 v0.8 - Shadow event network

- low-latency non-public event flow;
- app progressive-confidence interface;
- no safety-critical alert claims;
- latency and reliability measurement.

24.5 Later stages

- formal consensus analysis;
- larger validator diversity;
- privacy-preserving data access;
- external scientific partnerships;
- public-agency consultation;
- governed pilot only after safety review.

25 Conclusion

Namazu Chain proposes a useful-work consensus architecture centered on verifiable improvement of shared physical-world models. The key departure from ordinary proof-of-work is not simply replacing SHA-style hashing with matrix arithmetic. The protocol must prove that a candidate addresses an authenticated task, materially improves a declared baseline, survives delayed held-out validation, respects task-specific physical constraints, and remains attributable to its solver and verifiers.

The system therefore treats PoTGW certificates as the source of normal work credit while using a separate quorum settlement mechanism to order history. Different regions and buildings can progress in parallel. Invalid claims reduce domain-specific trust. A later mismatch does not automatically erase an earlier result; it can open a versioned investigation into changed physical state. Immediate event communication remains separate from final ledger settlement.

The project-reported v0.5 result advances this architecture beyond a single-machine concept proof. A four-process local devnet reportedly finalized 30 checkpoints and 20 work certificates across 11 task domains with primary hash mining disabled, while rejecting a false claim and a copied-work attempt and preserving a simulated building-state transition. This is evidence that the protocol structure can be made executable under deterministic synthetic conditions.

The result remains an engineering milestone rather than scientific or public-safety validation. The next credible step is a reproducible v0.6 replay against recorded, licensed accelerometer

datasets, with exact manifests, independent reruns, baseline comparisons, uncertainty reporting, and publication of all acceptance and failure results. Only after recorded-data evidence should the project progress to controlled physical structures and a non-alerting shadow event network.

26 Appendix A - Canonical object sketches

26.1 A.1 Task object

```
{
  "schema": "namazu.task.v1",
  "event_or_maintenance_epoch": "EQ-TEST-0007",
  "region_id": "JP-TOKYO-CELL-0042",
  "window_start_ns": 0,
  "window_end_ns": 10000000000,
  "task_class": "GROUND_LOCAL",
  "aggregation_level": 0,
  "physical_height_class": null,
  "building_id": null,
  "building_state_epoch": null,
  "input_data_root": "...",
  "algorithm_policy_root": "...",
  "previous_level": "T1",
  "target_level": "T2",
  "scoring_policy_root": "..."
}
```

26.2 A.2 Candidate commitment

```
{
  "schema": "namazu.commitment.v1",
  "task_id": "...",
  "solver_node_id": "...",
  "candidate_payload_hash": "...",
  "evidence_root": "...",
  "solver_sequence": 19,
  "commitment_signature": "..."
}
```

26.3 A.3 Work certificate

```
{
  "schema": "namazu.vtwc.v1",
  "task_id": "...",
  "solver_node_id": "...",
  "commitment_id": "...",
  "algorithm_id": "LOCAL-BASIS-FXP-v1",
  "input_data_root": "...",
  "candidate_payload_root": "...",
  "observed_error": 421930,
  "held_out_error": 498210,
  "physical_penalty": 0,
  "regional_penalty": 1200,
  "stability_penalty": 500,
```

```
  "complexity_penalty": 8000,
  "baseline_score": 650000,
  "candidate_score": 508000,
  "previous_level": "T1",
  "new_level": "T2",
  "challenge_seed": "...",
  "verifier_attestations": ["...", "...", "..."],
  "logical_round": 27,
  "certificate_id": "..."
}
```

The numeric values above are illustrative only. Implementations must not copy them as safety thresholds.

27 Appendix B - Validator pseudocode

```
verify_candidate(task, commitment, reveal, challenge_seed):
    require canonical_hash(reveal) == commitment.candidate_payload_hash
    require verify_signature(commitment.solver_node_id, commitment)
    require task.is_open_at(reveal.logical_round)

    inputs = load_and_verify_inputs(task.input_data_root)
    require inputs.canonical_root == task.input_data_root

    fit_set, hold_set = derive_challenge_split(inputs, challenge_seed)
    model = decode_canonical_model(reveal.model_payload)

    observed_error = evaluate_fixed_point(model, fit_set)
    held_error = evaluate_fixed_point(model, hold_set)
    physical_penalty = evaluate_physical_policy(model, task)
    regional_penalty = evaluate_parent_consistency(model, task.parents)
    stability_penalty = evaluate_stability(model, task)
    complexity_penalty = evaluate_complexity(model, task)

    score = weighted_saturated_sum(
        observed_error,
        held_error,
        physical_penalty,
        regional_penalty,
        stability_penalty,
        complexity_penalty)

    require level_transition_is_valid(task, score, each_component)
    return signed_verification_attestation(all_results)
```

28 Appendix C - Repository publication structure

```
namazu-potgw/
  README.md
  WHITEPAPER.md
  NAMAZU-PoTGW-Whitepaper-v0.1.pdf
  NAMAZU-PoTGW-Whitepaper-v0.1.html
```

```
NAMAZU-PoTGW-Whitepaper-v0.1.tex
references.bib
figures/
  architecture.svg
  work-graph.svg
  protocol.svg
  building-state.svg
specs/
  task-schema.json
  commitment-schema.json
  certificate-schema.json
test-vectors/
implementation/
SECURITY.md
GOVERNANCE.md
```

29 References

Allen, Richard M. et al. 2025. "Global Earthquake Detection and Warning Using Android Phones." *Science* 389. https://doi.org/10.1126/science.ads4779.

Castro, Miguel, and Barbara Liskov. 1999. "Practical Byzantine Fault Tolerance." In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 173–86.

Di Nuzzo, Flavio, Davide Brunelli, Tommaso Polonelli, and Luca Benini. 2021. "Structural Health Monitoring System with Narrowband IoT and MEMS Sensors." https://arxiv.org/abs/2104.13029.

Dong, Zhongli, Young Choon Lee, and Albert Y. Zomaya. 2019. "Proofware: Proof of Useful Work Blockchain Consensus Protocol for Decentralized Applications." https://arxiv.org/abs/1903.09276.

Hoffmann, Felix. 2022. "Challenges of Proof-of-Useful-Work." https://arxiv.org/abs/2209.03865.

Kong, Qingkai, Richard M. Allen, Louis Schreier, and Young-Woo Kwon. 2016. "MyShake: A Smartphone Seismic Network for Earthquake Early Warning and Beyond." *Science Advances* 2 (2): e1501055. https://doi.org/10.1126/sciadv.1501055.

Michel, Clotaire, Philippe Guéguen, Saber El Arem, Jacky Mazars, and Panagiotis Kotronis. 2007. "Full Scale Dynamic Response of a Reinforced Concrete Building Under Weak Seismic Motions Using Earthquake Recordings, Ambient Vibrations and Modelling." https://arxiv.org/abs/0710.1205.

National Institute of Standards and Technology. 2024a. "FIPS 204: Module-Lattice-Based Digital Signature Standard." NIST. https://doi.org/10.6028/NIST.FIPS.204.

———. 2024b. "FIPS 205: Stateless Hash-Based Digital Signature Standard." NIST. https://doi.org/10.6028/NIST.FIPS.205.

Roohi, Milad, Eric M. Hernandez, and David V. Rosowsky. 2020. "Performance-Based Post-Earthquake Decision-Making for Instrumented Buildings." https://arxiv.org/abs/2002.11702.

U.S. Geological Survey. 2019. "USGS ShakeAlert Earthquake Early Warning System." 2019. https://www.usgs.gov/news/featured-story/usgs-shakealert-earthquake-early-warning-system.