

Mesh-Derived Time: Master-Less Consensus and Joint Spacetime Optimization

by [Nick Clark](#) | Published April 25, 2026

Time Synchronization Is Master-Bound and Brittle

Distributed systems treat time as a top-down distribution: an authoritative source (atomic clock, GNSS time, stratum-0 server) broadcasts a reference, and recipients adjust their local clocks toward it. NTP, PTP, IEEE 1588, White Rabbit, GNSS time, and most data-center synchronization schemes follow this pattern.

Master-bound architectures fail predictably when the master is unavailable (network partition), contested (no agreement on which master to trust), spoofed (adversarial GNSS time injection), or compromised (malicious authoritative source). Each failure mode is documented and increasingly exploited, particularly in critical infrastructure and defense contexts.

Existing alternatives — Roughtime, Hybrid Logical Clocks (HLC), TrueTime — partially address this but do not unify with spatial coordinate establishment, do not produce credentialed timestamps suitable for audit, and do not support the kind of governance-bound multi-attester consensus that regulated environments (financial markets, legal evidence, medical records) require.

1. The Primitive: Master-Less Cooperative Consensus

Mesh-derived time establishes time through cooperative consensus among credentialed participants, with no required master. Each participating unit maintains a local clock with a learned drift model, exchanges credentialed time observations with neighbors, and integrates received observations into its own time estimate weighted by credential authority and observation lineage.

Time consensus is a distributed estimation: each unit's time is the weighted aggregate of its own learned drift trajectory and credentialed observations from peers. No participant is privileged as master; if a high-authority participant joins (a regulatory time authority, a GNSS-receiving unit), its observations carry higher weight, but the architecture does not require it.

Master-less operation makes the primitive resilient to partition: when the mesh fragments, each fragment maintains internally consistent time. When fragments rejoin, the consensus mechanism reconciles, recording any required adjustment in lineage.

2. Per-Agent Learned Drift Models

Each participating unit learns its own clock's drift characteristics under operating conditions. The drift model captures temperature dependence, aging, oscillator type, and operational regime, producing a per-unit prediction of how local time evolves between observation events.

The drift model is itself a governed observation. The unit publishes its model, signs it with its credential, and updates it as new drift evidence accumulates. Other participants consume the model to weight observations from this unit appropriately: a unit with a high-quality oven-controlled oscillator gets higher confidence than a unit with a consumer-grade crystal.

Drift models enable accurate time estimates between consensus events, reducing required communication frequency and enabling operation across intermittent connectivity.

3. Ranging-Piggyback Synchronization

Time observations and range observations are produced jointly. When two units exchange a UWB ranging packet, the time-of-flight is the range, and the timestamps in each direction enable mutual time synchronization. The mesh-time primitive captures both contributions in a single observation that updates both the spatial coordinate and the time consensus.

Joint capture eliminates the redundancy of separate time- and position-distribution channels. The same packet that establishes 'you are 47.3 m from me' also establishes 'your clock is 218 μ s ahead of mine.' Both observations are signed, both carry lineage, both feed downstream solutions.

The architectural consequence is that time and space are co-resolved, with cross-correction: a unit with confident time but uncertain position receives both updates from a peer with confident position but uncertain time, with the joint solution converging faster than separate channels would allow.

4. Joint Spacetime Graph Optimization

The full mesh state is a joint spacetime graph: nodes are units (each with a position and time estimate), edges are credentialed range/time observations, and the optimization produces a globally consistent spacetime solution. The optimization is incremental — new observations refine the solution without recomputing from scratch — and lineage-bound — each refinement is recorded with its supporting observations.

Joint optimization captures cross-effects that separate optimizations miss: a clock drift correction implies a position correction (the assumed time-of-flight changes), and vice versa. The joint solution is therefore more accurate than separate spatial and temporal solutions composed after the fact.

The optimization is governance-credentialed: regulatory authorities can sign reference observations that the optimization weights highly, peer authorities contribute lower-weight observations, and the resulting solution carries the lineage of all admitted contributions.

5. Multi-Attester Consensus Timestamps for Audit

Audit-grade timestamping requires that no single clock be the basis for a recorded event time. The multi-attester primitive produces timestamps signed by a quorum of credentialed time authorities: a financial transaction is timestamped by the exchange's time authority and the regulatory time authority and the participant's own time authority, with all three signatures required for the timestamp to be admissible.

Multi-attester timestamps survive single-authority compromise: an attacker who compromises one time authority cannot produce admissible timestamps because the quorum requirement is unmet. The audit trail is structurally tamper-evident.

This serves the regulated-timestamping market: MiFID II microsecond timestamping, SEC Reg SCI requirements, RFC 3161 trusted timestamps, distributed-database commit ordering, and legal-evidence chain-of-custody. The mechanism is the same; the configurations change.

6. Relativistic and Calendar Handling

At sufficient precision and altitude/velocity differences, relativistic effects are non-negligible: GPS satellites, aviation, low-earth-orbit operations, and high-precision financial trading all exhibit relativistic time corrections. The mesh-time primitive

incorporates a relativistic correction layer that adjusts time observations between participants according to declared velocity and gravitational potential.

Calendar handling — leap seconds, time zone transitions, smear strategies during leap-second insertion — is also governance-credentialed: the calendar authority signs the calendar policy, participants apply the policy uniformly, and any deviation is structurally detectable.

These corrections are typically handled separately in current architectures with significant cross-system inconsistency. Unified governance produces consistent treatment across all participants regardless of operating regime.

7. What This Is Not

This is not NTP, PTP, or White Rabbit. Those protocols are master-bound time-distribution. The governed primitive is master-less consensus.

This is not Roughtime. Roughtime addresses GNSS-spoofing detection through randomized server queries. The governed primitive embeds the equivalent guarantees in the credential and lineage structure of every observation.

This is not Hybrid Logical Clocks (HLC). HLC produces causally consistent ordering; the governed primitive produces wall-clock time consensus suitable for audit, regulation, and physical-world coordination.

Conclusion

Mesh-derived time replaces master-bound clock distribution with credentialed cooperative consensus, joint spatial-temporal optimization, and multi-attester audit-grade timestamps. The primitive composes with mesh-derived coordinates (joint spacetime graph), with marker-track transport (markers as time references), and with matched-pair settlement (proximity windows defined in joint spacetime).

Disclosed under USPTO provisional 64/049,409, the primitive serves financial timestamping, legal evidence, distributed databases, regulated-fleet operations, and assured-PNT defense applications under a single architectural mechanism.