MKSTORM:
A Field-Coherent Mesh for Real-Time Perception, Memory, and Synchronization

Enrique Flores

July 28, 2025

Abstract

This document introduces **MKSTORM**—a field-coherent mesh architecture for real-time perception, memory, and distributed synchronization. Anchored by deterministic 128-bit entropy blocks generated by MKRAND, MKSTORM provides universally unique addressing and microsecond-resolved timing across heterogeneous memory media and networked nodes. Rather than relying on centralized clocks or consensus protocols, MKSTORM synchronizes temporal state through shared digital entropy snapshots, enabling decentralized autonomous systems to coordinate, store, and reason about complex, dynamic events in time and space with precision and trust. Designed for edge AI and robotics workloads, MKSTORM acts as a temporal substrate for time-aligned retrieval, cross-agent awareness, and programmable future-state activation—extending from edge devices to the datacenter and battlefield, and across time domains from milliseconds to millennia.

1 Introduction

The growing demand for AI-driven autonomy in drones, vehicles, and industrial robotics requires databases capable of:

- Storing and retrieving high-frequency time- and space-correlated multi-sensor data streams with minimal overhead and latency.
- Providing deterministic timestamps with microsecond precision.
- Ensuring temporal immutability and authenticated provenance across distributed nodes.
- Enabling connected systems to retrieve historical context (e.g., sensor trends) in real time.

Traditional databases lack the combination of chain-of-provenance guarantees, low-latency updates, computational efficiency, and distributed temporal precision required by such edge systems. MKSTORM addresses these gaps by merging principles from digital blockchain, mesh architectures, and time-series data stores.

1.1 Universal Data — Constant-Time Access

MKSTORM enables **constant-time** ($O(1)$) insertions and retrievals for data of any size or type—ranging from high-frequency sensor streams to large binary objects (BLOBs) and spatial telemetry. Each record is spatially addressed via a deterministic 128-bit key, derived from its temporal coordinates, spatial origin, or semantic attributes—eliminating the need for external indexes, lookup tables, or opaque, artificial keys.

In addition, each record is temporally correlated with every other piece of data collected by that node, as well as all other nodes participating in the mesh. By "field-coherent," we refer to the ability to capture not only any piece of data and associate it with spatial attributes, but also to uniformly, repeatably, and immutably bind it to a microsecond-precise point in time—a digital snapshot of the time experienced by all nodes in the mesh. All nodes can refer to this snapshot and compute over it as reliably as they can address memory.

Time and space become first-class citizens in a group-addressable, group-coherent view, with none of the floating-point ambiguity, clock drift, or time-zone confusion experienced by legacy systems using conventional timers.

This universal addressing scheme allows every node in the mesh to compute the storage location and verification hash of any record, regardless of where or when it originated. Although physical access time may vary depending on medium (RAM, SSD, archival) or mesh topology, the logical access model remains flat, deterministic, and globally coherent.

By fusing structured time-series indexing with attribute-driven hashing, MKSTORM delivers an access fabric capable of powering both conventional data workloads and real-time AI pipelines—including future Retrieval-Augmented Generation (RAG) engines, predictive models, and situational-awareness overlays.

2 Core Concepts

2.1 Digital Entropy-Driven Time Indexing

MKSTORM uses 128-bit entropy blocks generated by `MKRAND` to serve as both:

- A digital time anchor for all events occurring during the mesh's operational lifespan.
- A universally unique spatial identifier for mapping data to physical memory or network address space.

Time in MKSTORM is defined numerically by a two-part system:

- The **Long Count** — a 64-bit integer serving as the primary database clock. It increments deterministically at fixed intervals (e.g., every 256 FPGA cycles), supporting centuries of uptime with microsecond or sub-microsecond precision.
- The **Short Count** — a 16-bit sub-clock that discretizes events within each Long Count interval. It provides resolution down to 1 picosecond at high-speed clock domains (e.g., 4 GHz), enabling precise sequencing of intra-block events without ambiguity or collision.

Each interval is anchored by a **PSI index**, a deterministic 128-bit entropy block visually represented as `[<:A0AF48C445ABACC2119DB3529F85237C:>]`, denoting the global state of the mesh during that time slice. Derived from the genesis block, the PSI index provides a cryptographically verifiable fingerprint of all relevant local and remote data captured in that interval.

Together, the PSI index, Long Count, and Short Count form a globally synchronized, digitally verifiable clock framework. This enables distributed systems to reason and communicate about time, causality, and coordination at sub-microsecond resolution—without centralized clocks or consensus protocols.

2.2 Time-Series Data Model

All data entries in MKSTORM are stored as:

```
(originator_id, index, long_count, short_count, data_payload)
```

where:

- `originator_id` is a 128-bit globally unique identifier for the node or process that originated the data.
- `index` identifies the specific 128-bit entropy block in the chain that holds the entry.
- `long_count` is the numeric block height at the time of item capture.
- `short_count` provides high-resolution temporal placement within the block interval.
- `data_payload` contains the binary content (sensor, telemetry, state, etc.).

This model allows ultra-efficient packing and retrieval of high-frequency time-aligned sensor data and supports deterministic event replay, predictive scheduling, and real-time AI inference across the swarm or system.

3 Spatiotemporal Synchronization and Dynamic Reasoning

Figure 1 illustrates a core advantage of the system: the ability to visualize and reason about the evolution of dynamic systems with digitally accurate, temporally ordered identity tracking.

On the left side, a 3D plot displays particle positions on a spherical topology, with one particle highlighted in magenta. This corresponds to a specific `point_id`, whose identity hash—called the `psi_index`—is shown beneath the plot in symbolic form `[<:...:>]`.

Each data record is assigned its own individual blockchain. The PSI index of a selected point changes over time because we access it via its genesis block, and can deterministically compute any future state. By sharing the genesis block or descendant hashes, we can refer to any object state or history across the mesh.

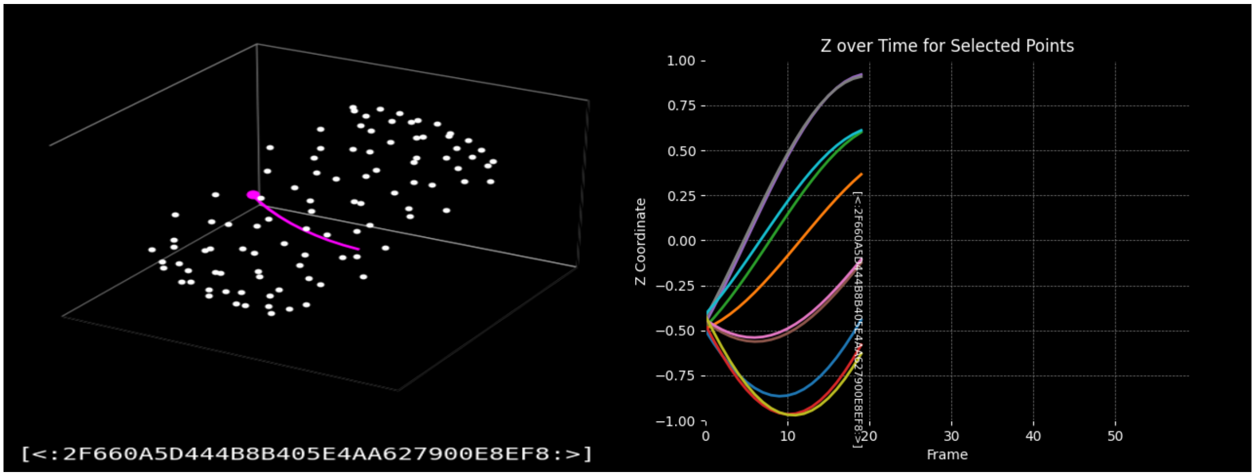

Figure 1: Combined visualization of particle positions (left) and Z-coordinate history (right), with identity-linked tracking hash for reasoning over distributed dynamic systems.

The right panel shows the Z-coordinate evolution of particles over time—presenting a scalar lens on motion, correlation, or causality.

The power of this representation lies in its ability to unify spatial, temporal, and identity information. Any computing node in the mesh can transmit or receive data about a specific particle—across arbitrary spatial distance—without ambiguity or loss of temporal context.

By deriving all time references from deterministic entropy, systems can reason about causality and update order even under asynchronous or lossy conditions. This makes the system ideal for distributed, real-time reasoning over time-evolving, high-frequency data with absolute precision.

3.1 Deterministic Data Sharding

By using the high bits of the 128-bit entropy block (the **PSI index**) as a shard key, MK-STORM enables deterministic distribution of data across multiple storage layers and nodes in a cluster. This supports tiering across memory hierarchies (RAM, SSD, HDD, or remote nodes) while preserving globally consistent and collision-free addressing.

Every record, event, or particle state in the system is inherently mapped to a globally unique address derived from its PSI index. This address is not merely a hash—it is a functional coordinate in a 128-bit address space that enables one-hop, direct retrieval without the need for routing tables or lookup indices.

This vast address space (2^{128}) allows seamless, indefinite scalability. New storage can be added either locally (within the same rack) or globally (via IPv6 or decentralized object stores), and any node respecting the deterministic scheme can instantly locate the data it needs.

This architecture supports exascale system design. It eliminates traditional coordination mechanisms—such as DHTs or consensus-based directories—by turning every PSI index into both a fingerprint and a locator. The result: a flat, verifiable, causally sound substrate for high-performance computing at planetary scale.

4 AI and RAG Integration (Optional Extension)

To support LLM-based agents and real-time reasoning, MKSTORM can integrate a Retrieval-Augmented Generation (RAG) layer. The interface supports natural language queries such as:

> "What is the average altitude and yaw change over the past 34.5 seconds?"

Such queries are translated into entropy-derived time windows, records are retrieved from the distributed system, and structured responses are composed for the model.

5 Applications

- Drone swarm coordination and real-time airspace modeling.
- Autonomous robotics with sensor fusion and historical awareness.
- Industrial monitoring and predictive maintenance.
- Military situational awareness systems with encrypted data logging.
- Edge AI nodes requiring low-latency RAG-style temporal memory.

6 Patent Claims

1. A method for deterministic data retrieval in a distributed computing environment, comprising:
   (a) generating a 128-bit entropy block representing a globally synchronized timestamp;
   (b) associating said entropy block with one or more data records captured during the corresponding time interval;
   (c) computing a shard key from a portion of the entropy block to determine a physical or logical storage location;
   (d) retrieving said data records using the entropy-derived address without reliance on intermediate lookup tables or consensus protocols.
2. A method of storing time-series data across distributed nodes using a cryptographically synchronized chain of entropy blocks, wherein each block encodes both temporal and spatial context.
3. A system for assigning globally unique addresses to multi-sensor data streams, wherein each record is mapped to a 128-bit entropy block generated by a deterministic function.
4. A data ingestion mechanism incorporating FPGA-based logic to generate entropy blocks and associate them with real-time incoming data streams, thereby enabling low-latency timestamping and indexing.

5. A distributed database architecture wherein entropy blocks serve simultaneously as:
   (a) timing signals;
   (b) record locators;
   (c) replication keys for sharded storage.
6. A method for enabling AI agents to perform natural-language queries over temporally indexed data, comprising:
   (a) translating user queries into entropy-derived time windows;
   (b) retrieving corresponding data records from distributed nodes;
   (c) returning a structured response suitable for consumption by a language model.