

DBM-SI Structural Messaging

Payload-Structure Communication Model for Structural Intelligence

Under constraint, structure becomes the carrier of meaning.

Version: v0.1.1

Project: DBM-SI Structural Messaging

Status: Conceptual Specification

Scope: Structural Intelligence Runtime Communication Model

1. Introduction

Structural Messaging in DBM-SI defines the communication model through which runtime payload and structural evidence are jointly transmitted, validated, and frozen into reproducible system states.

Unlike conventional messaging systems that transport data alone, Structural Messaging carries payload together with structural context — including evidence chains, invariant hashes, execution status, and snapshot references — ensuring that structural intelligence execution remains auditable, reproducible, and verifiable across runtime boundaries.

This model serves as the bridge between execution payload flow and structural knowledge formation in Structural Intelligence systems.

The architectural motivation of Structural Messaging — including constrained payload signaling scenarios such as bridge signaling, trigger-based observation, and the quantum-entanglement analogy — is explained in:

****DBM-COT ITEM #266 — Structural Messaging Architecture in DBM-SI****

This specification focuses on the communication model itself.

Short Definition

Structural Messaging is the payload-structure communication model of DBM-SI runtime systems, enabling payload transmission together with structural evidence, execution auditability, and snapshot-level reproducibility.

One-Line Definition

Structural Messaging enables payload transmission with structural evidence in Structural Intelligence runtime systems.

2. Structural Messaging in the Structural Intelligence Stack

Structural Messaging occupies the communication layer between runtime execution and structural knowledge formation.

Structural Intelligence Stack

Structural Algorithms

↓

EvidenceChain

↓

Structural Messaging

↓

Snapshot / Knowledge State

Responsibilities include:

- payload transmission
- structural context propagation
- execution auditability
- snapshot reproducibility
- validator-aware communication

3. Payload-Structure Communication Model

Structural Messaging treats a message as the combination of runtime payload and structural context.

$$\text{StructuralMessage} = \text{Payload} + \text{StructuralContext}$$

Payload examples:

- IR objects
- alignment results
- candidates
- runtime parameters
- execution outputs

StructuralContext examples:

- EvidenceChain reference
- invariantHash
- ExecutionStatus
- snapshotId
- validator state
- convergence evidence

Payload alone does not constitute a Structural Message.

Note — Constrained Communication Origin

Structural Messaging originates from communication scenarios where payload cannot be transmitted directly and must be encoded through shared structural conventions.

This specification defines the messaging abstraction that generalizes such communication patterns into Structural Intelligence runtime systems.

The motivating narrative examples are documented in ITEM #266.

4. Structural Messaging Objects

Structural Messaging defines three primary communication objects.

4.1 EvidenceMessage

Minimal transferable structural evidence unit.

Conceptual fields:

timestamp
producer
payload
evidenceHash
invariantHash

Purpose:

- transmit structural evidence
- bind payload to invariant context
- provide validator input

4.2 ExecutionReceipt

Verifiable record of structural execution.

Conceptual fields:

status
evidenceChainHead
snapshotId
mode
cost
reason
timestamp

Purpose:

- record execution outcome
- provide audit trail
- connect runtime events to structural state

4.3 SnapshotMessage

Frozen structural knowledge state.

Conceptual fields:

baselineId
eventHashes
mode
validatorState
convergenceState
timestamp

Purpose:

- freeze runtime structural state
- enable reproducibility
- support audit and recovery

5. Messaging State Model

Structural Messaging propagates runtime state explicitly.

ExecutionStatus

OK
SAFE_MODE
STOPPED
COOLDOWN
QUARANTINE

Status must be included in:

- ExecutionReceipt
- SnapshotMessage
- validator reports

6. Propagation Model and Medium Independence

Structural Messaging defines communication at the level of structural state representation rather than physical transport mechanisms.

6.1 Payload Transmission

Structural Messaging operates over classical computing media:

- memory
- storage
- network protocols
- distributed runtime systems

6.2 Structural State Synchronization

Structural Messaging focuses on synchronization of structural state across execution environments, including:

- evidence consistency
- invariant preservation
- snapshot reproducibility
- validator agreement

6.3 Quantum Entanglement Analogy

Structural Messaging may be conceptually compared to quantum entanglement as a theoretical analogy.

In quantum entanglement, systems maintain correlated states without classical payload transfer.

Similarly, Structural Messaging focuses on correlated structural state across runtime boundaries.

This specification does not assume quantum implementation.

The analogy illustrates that Structural Intelligence messaging concerns ****state correlation rather than data movement****.

6.4 Messaging Reach and Medium Penetration

Because Structural Messaging is defined independently of physical transport mechanisms, its effective reach depends on the available state-synchronization media.

As computation and communication substrates evolve — from memory buses to distributed systems and potentially quantum communication channels — the propagation distance and medium penetration capability of Structural Messaging may expand accordingly.

Propagation Principle

Structural Messaging is not bound to a transport medium; it is bound to structural state synchronization.

7. Relationship to DBM-SI Runtime

Conceptually:

DBM-SI Runtime = execution engine

Structural Messaging = payload-structure communication model

Runtime components interacting with Structural Messaging include:

- EvidenceChain
- EvidenceValidator
- ExecutionStatus
- Snapshot system
- ConvergenceChecker
- Orchestrator

8. Auditability and Reproducibility

Structural Messaging enables:

Execution Auditability:

- receipts linked to evidence chains
- validator-visible status propagation
- invariant verification

Snapshot Reproducibility:

- frozen runtime state
- deterministic evidence references
- snapshot history preservation

9. Version Note

- v0.1 — Initial specification
- v0.1.1 — Alignment update with ITEM #266 architecture narrative (no protocol changes)

End of StructuralMessaging-Spec v0.1.1