ENGINEERING STANDARDS LIBRARY / DATA, STATE, AND EVIDENCE

Vector Database & Local-First Sync Architecture Standard

The architecture of AI memory and state management has failed.

PERMANENT DOCUMENT ID

MYTHOS-DAT-0001

PUBLICATION

Candidate Standard
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

Vector Database & Local-First Sync Architecture Standard

DOCUMENT ID
MYTHOS-DAT-0001

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-24

SCHEDULED REVIEW
2027-07-24

12	6	4	1
ATOMIC CRITERIA	ASSURANCE VIEWS	IMPLEMENTATION PROFILES	PERMANENT IDENTITY

Publication doctrine. This standard is a permanent library identity. Interpretation must preserve its recorded evidence, controlled exceptions, formal revision history, and explicit relationship to companion standards.

DOMAIN CONTROL SYSTEM

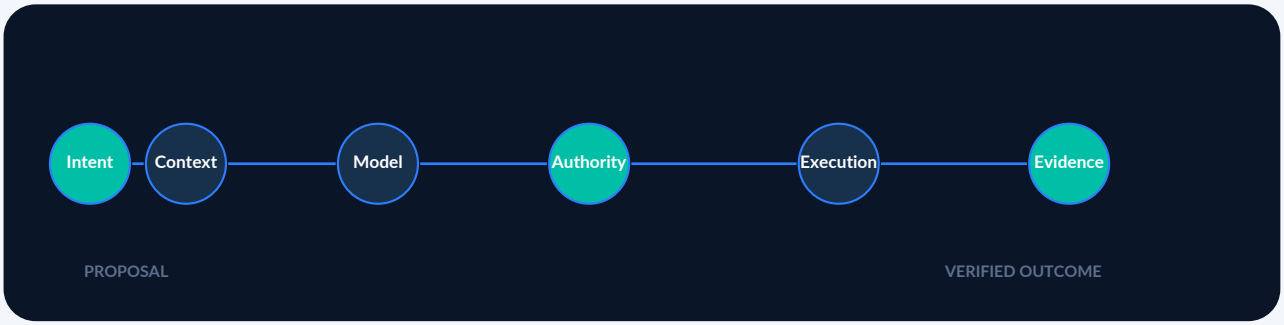
MYTHOS-DAT-0001 inside the data, state, and evidence constellation

Connected assurance	Architecture, implementation, operations, evidence, and lifecycle are evaluated as one controlled system.
Specialization rule	Companion standards may add precision, but cannot silently weaken this publication.

Executive orientation

Defines vector retrieval and local-first synchronization architecture, including indexing, embeddings, consistency, conflict resolution, offline operation, provenance, and lifecycle control.

WHY THIS STANDARD EXISTS	The architecture of AI memory and state management has failed.
OPERATIONAL SCOPE	This standard covers the evaluation of: - Vector databases and indexing algorithms (HNSW, IVF, DiskANN) - Embedded/local-first vector stores (sqlite-vec, LanceDB, Chroma) - Enterprise/cloud vector engines (pgvector, Pinecone, Weaviate) - Local-first synchronization engines (ElectricSQL, PowerSync, CRDT libraries) - Temporal Knowledge Graphs for agent memory (Zep, Mem0 patterns) - Embedding lifecycle management, versioning, and drift remediation - Privacy-preserving and on-device semantic search
PRIMARY AUDIENCE	- Platform engineers and architects designing AI memory and state systems - AI engineers building context pipelines and RAG architectures - Security teams evaluating data egress and privacy in semantic search - Edge and mobile engineers building offline-first AI applications - Standards bodies seeking a reference local-first data methodology



Assurance principle. Model capability, data quality, identity, authority, operational evidence, and human accountability are treated as a connected system. A high aggregate score cannot compensate for a failed mandatory safety or authority boundary.

Contents

Evaluation criterion index

This standard contains 12 atomic criteria. Each criterion is independently testable and requires retained evidence.

ID	EVALUATION CRITERION
4.1.1	Read/Write Autonomy
4.1.2	Sync Engine Architecture
4.2.1	Conflict Resolution Mechanism
4.3.1	Indexing Algorithm
4.4.1	Model Versioning
4.5.1	Fact Temporality
4.6.1	Egress Control
5.1.1	MDSAS-Partition
5.1.2	MDSAS-Drift
6.1.1	Memory Poisoning via Sync
6.1.2	Embedding Inversion Attack
6.1.3	Vector Drift Degradation

MYTHOS-DAT-0001-4.1.1

4.1.1 Read/Write Autonomy

Normative source requirement

Can the system read and write semantic state without network connectivity?

Score	Criteria
0	Requires network for all reads and writes
1	Read cache exists, but writes require network
2	Local reads/writes, but manual sync required
3	Automatic local-first read/write with background sync (e.g., `sqlite-vec`)
4	Local-first with predictive prefetching and background re-indexing
5	Perfect autonomy: formally verified offline operability, zero-latency local vector search, automated conflict queue

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-4.1.2

4.1.2 Sync Engine Architecture

Normative source requirement

How is state synchronized between local and remote?

Score	Criteria
0	Direct database replication (fragile, network-dependent)
1	API-based sync with timestamp conflict resolution
2	API-based sync with basic transactional retry
3	Dedicated local-first sync engine (ElectricSQL/PowerSync)
4	Sync engine with bandwidth-optimized vector delta transmission
5	Perfect sync: formally verified convergence, zero data loss on partition, sub-second sync on reconnect

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-4.2.1

4.2.1 Conflict Resolution Mechanism

Normative source requirement

How are conflicting concurrent updates resolved?

Score	Criteria
0	Last-write-wins (destroys semantic context)
1	Manual conflict resolution required
2	Basic CRDTs for scalar data, but vectors are overwritten
3	Full CRDT support for all data types, including vector metadata
4	Semantic merge: LLM-assisted resolution of conflicting memory/context states
5	Perfect resolution: formally verified CRDTs, deterministic semantic merge, zero-state-loss on any network topology

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-4.3.1

4.3.1 Indexing Algorithm

Normative source requirement

Does the system use state-of-the-art, production-proven indexing?

Score	Criteria
0	Brute-force / exact search only (unscalable)
1	Basic IVF (Inverted File Index)
2	HNSW (Hierarchical Navigable Small World) with default parameters
3	Tunable HNSW or DiskANN with metadata pre-filtering
4	Hardware-accelerated indexing (GPU/NPU) with quantization support
5	Perfect indexing: formally verified recall/precision bounds, adaptive indexing based on hardware, zero-latency local inference

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-4.4.1

4.4.1 Model Versioning

Normative source requirement

Are embedding models treated as a strict schema?

Score	Criteria
0	No model versioning; vectors from multiple models mixed
1	Model version recorded, but no enforcement
2	Vectors partitioned by model version
3	Automated re-indexing pipeline triggered on model change
4	Blue/green re-indexing with A/B testing of retrieval quality
5	Perfect governance: formally verified vector spaces, automated drift detection, zero-downtime zero-dual-write re-indexing

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-4.5.1

4.5.1 Fact Temporality

Normative source requirement

Does the system track the temporal validity of facts (when facts became true/false)?

Score	Criteria
0	Static facts only; no time context
1	Timestamps exist but are not used for retrieval
2	Time-decay scoring applied to search results
3	Full Temporal Knowledge Graph (fact start/end dates, contradiction resolution)
4	Memory ACLs, trust classes, and user-editable temporal state
5	Perfect temporal design: formally verified time-travel queries, automated stale-fact quarantine, cryptographic provenance for all memories

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-4.6.1

4.6.1 Egress Control

Normative source requirement

Does the system prevent raw context from leaving the local boundary?

Score	Criteria
0	All search queries sent to cloud vector API
1	Local search, but raw data synced to cloud unencrypted
2	Local search with TLS encryption in transit
3	Local search with envelope encryption (KMS) at rest
4	Zero-knowledge search; only encrypted vectors or proofs leave the device
5	Perfect sovereignty: fully homomorphic encrypted (FHE) search capabilities, formally verified zero raw context egress

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-5.1.1

5.1.1 MDSAS-Partition

Normative source requirement

- Network Partition: 100+ tests simulating network drop, concurrent local writes, and reconnection to verify CRDT convergence.
- Conflict Injection: 50+ tests injecting semantically conflicting memories to verify LLM-assisted merge logic.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Architecture records, implemented configuration, test evidence, operational telemetry, exception decisions, and accountable approval.
ASSESSMENT METHOD	MATURITY SIGNAL
Inspect the architecture and configured control, execute normal and adverse scenarios, verify measurable outcomes, sample retained evidence, and confirm that exceptions are time-bound and approved.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-5.1.2

5.1.2 MDSAS-Drift

Normative source requirement

- Model Swap: 50+ tests triggering an embedding model upgrade to verify re-indexing pipelines and A/B recall validation.
- Temporal Decay: 50+ tests querying for facts whose validity has expired to verify they are omitted or quarantined.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Model and dataset cards, lineage, licenses, evaluation reports, release approvals, and rollback artifacts.
ASSESSMENT METHOD	MATURITY SIGNAL
Inspect the architecture and configured control, execute normal and adverse scenarios, verify measurable outcomes, sample retained evidence, and confirm that exceptions are time-bound and approved.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-6.1.1

6.1.1 Memory Poisoning via Sync

Normative source requirement

Severity: Critical Description: A compromised edge node syncs malicious or hallucinated memories to the enterprise knowledge graph, corrupting the global agent context.

Required Controls: 1. Cryptographic signing of all synced state updates. 2. Trust classes for memory (e.g., user-stated vs. agent-inferred). 3. Quarantine and verification pipeline for cross-scope memory promotion.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Trace the decision path from identity and policy through execution, attempt bypass and privilege-escalation scenarios, verify revocation behavior, and inspect the resulting evidence record.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-6.1.2

6.1.2 Embedding Inversion Attack

Normative source requirement

Severity: High Description: An attacker exfiltrates the vector database and uses an embedding inversion model to reconstruct the original raw text (PII/IP).

Required Controls: 1. Local-first architecture minimizing attack surface. 2. Envelope encryption of vectors at rest. 3. Zero-knowledge sync (syncing only deltas or encrypted blobs).

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Model and dataset cards, lineage, licenses, evaluation reports, release approvals, and rollback artifacts.
ASSESSMENT METHOD	MATURITY SIGNAL
Exercise crash, retry, duplicate, reorder, partition, and restore scenarios; compare authoritative state before and after replay; confirm idempotency and recovery evidence.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-DAT-0001-6.1.3

6.1.3 Vector Drift Degradation

Normative source requirement

Severity: High Description: An unplanned update to the cloud embedding API silently changes the vector space, causing the local agent's retrieval quality to plummet to zero.

Required Controls: 1. Pinning embedding model versions (no auto-updates). 2. Automated recall testing before model migration. 3. Strict separation of vectors generated by different models.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance. Model and dataset cards, lineage, licenses, evaluation reports, release approvals, and rollback artifacts.
ASSESSMENT METHOD	MATURITY SIGNAL
Inject stale, conflicting, low-authority, and malicious information; inspect selection and citation behavior; verify scope isolation, correction, deletion, and provenance retention.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

Current authority and freshness register

These sources inform interpretation but do not convert this candidate publication into certification, legal compliance, or implementation evidence. Rolling sources must be version-pinned at adoption time.

AUTHORITY	VERSION / FRESHNESS	APPLICABILITY LIMIT
NIST - Artificial Intelligence Risk Management Framework (AI RMF 1.0)	NIST AI 100-1; current framework, revision underway Expires 2026-10-24	Voluntary risk-management framework; does not establish product certification or implementation effectiveness.
OpenTelemetry - Semantic Conventions	1.43.0 rolling specification; GenAI conventions maintained separately Expires 2026-08-24	Several GenAI and agent conventions remain under active development and require version pinning.
W3C - Verifiable Credentials Data Model v2.0	W3C Recommendation, May 15 2025 Expires 2027-01-24	Data model conformance does not establish issuer trust, credential truth, or ecosystem governance.
C2PA - Content Credentials Technical Specification	2.4 rolling publication Expires 2026-10-24	Provenance establishes signed assertions and tamper evidence, not the truth or quality of the underlying content.

Source lineage appendix

The following text preserves the substantive source draft in a normalized public form. Where it conflicts with the permanent publication identity, candidate status, current authority register, or evidence requirements above, the controlled publication layer governs.

0. Executive Mandate

The architecture of AI memory and state management has failed.

Organizations build autonomous agents and complex applications that rely entirely on centralized, cloud-hosted vector databases. When the network drops, the agent loses its memory. When the cloud provider experiences an outage, the application loses its mind. Furthermore, organizations attempt to synchronize semantic state across devices using last-write-wins timestamp algorithms, resulting in catastrophic context destruction and memory corruption.

This standard exists because:

1. Cloud-Dependent Memory is Fragile: An AI agent that cannot recall, reason, and execute offline is a toy, not a production tool. Memory MUST be local-first, reading and writing to an embedded vector store (e.g., `sqlite-vec`) with background synchronization to the enterprise.

2. Last-Write-Wins Destroys Semantic State: Synchronizing knowledge graphs or vector embeddings across a distributed system using timestamps guarantees data loss. Conflict-free Replicated Data Types (CRDTs) and semantic merge algorithms MUST govern all state synchronization.

3. Vector Drift is Ungoverned: When an organization updates an embedding model (e.g., moving from `text-embedding-3-small` to a new version), the existing vectors in the database become mathematically alienated. Unplanned, unverified model migrations destroy retrieval quality. Embedding lifecycle management MUST be a governed, versioned architectural process.

4. Privacy is Surrendered to Latency: Sending raw user context and enterprise intellectual property to a remote vector API for similarity search exposes data unnecessarily. Search MUST happen locally; only encrypted, redacted, or zero-knowledge proofs should traverse the network.

This document establishes the Mythos Data & Sync Architecture Standard (MDSAS), a comprehensive, evidence-based methodology for evaluating vector databases and local-first sync engines against offline resilience, semantic conflict resolution, and embedding lifecycle governance.

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- Vector databases and indexing algorithms (HNSW, IVF, DiskANN)
- Embedded/local-first vector stores (`sqlite-vec`, LanceDB, Chroma)
- Enterprise/cloud vector engines (pgvector, Pinecone, Weaviate)
- Local-first synchronization engines (ElectricSQL, PowerSync, CRDT libraries)
- Temporal Knowledge Graphs for agent memory (Zep, Mem0 patterns)
- Embedding lifecycle management, versioning, and drift remediation
- Privacy-preserving and on-device semantic search

1.2 Out of Scope

- General relational database design (covered in future MYTHOS-DBS standards)
- General observability semantic conventions (covered in MYTHOS-DAT-0002)
- General event sourcing and CQRS patterns (covered in MYTHOS-DAT-0004)

1.3 Audience

- Platform engineers and architects designing AI memory and state systems
- AI engineers building context pipelines and RAG architectures
- Security teams evaluating data egress and privacy in semantic search
- Edge and mobile engineers building offline-first AI applications
- Standards bodies seeking a reference local-first data methodology

Part II. Definitions and Taxonomy

2.1 Data & Sync Dimensions

Dimension	Definition
Local-First	An architectural paradigm where the primary database runs locally (on-device or on-edge), ensuring zero-latency reads/writes and offline operability, with asynchronous synchronization to the cloud.
CRDT	Conflict-free Replicated Data Type. A data structure that can be replicated across multiple networked computers, updated independently and concurrently, and mathematically resolved without conflicts.
Vector Drift	The degradation of retrieval quality that occurs when the embedding model used to generate vectors is updated or changed, rendering existing vectors mathematically incompatible with new queries.
Temporal KG	A knowledge graph that tracks the temporal validity of facts (when a fact became true, when it ceased to be true), preventing stale memory poisoning.
Semantic Merge	The resolution of conflicting data updates based on semantic meaning rather than simple timestamps (e.g., an LLM resolving two conflicting agent memories).

2.2 The Local-First Data Doctrine

> A cloud database is a cache, not a source of truth. A vector that cannot sync offline is ephemeral. A semantic conflict cannot be solved by a timestamp. An embedding model is a schema; changing it without migration is corruption.

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; cloud-only, timestamp sync, no drift management
1	Basic local cache but requires cloud for writes/conflict resolution
2	Functional offline mode but lacks CRDTs or formal drift governance
3	Solid production performance; local-first, CRDT sync, basic versioning
4	Strong performance; semantic merge, temporal KG, zero-trust sync
5	Best-in-class; sets the standard for sovereign, mathematically verified data sync

Weighting

Category	Weight	Rationale
Local-First & Offline Operability	20%	Network dependency is an architectural failure for AI agents
CRDT & Semantic Conflict Resolution	20%	Timestamp-based sync destroys semantic state
Vector Indexing & Retrieval Quality	15%	Search performance and recall determine agent intelligence
Embedding Lifecycle & Drift Governance	15%	Ungoverned model updates destroy production memory
Temporal Memory & Knowledge Graphs	15%	Without time, memory becomes a poisoning vector
Data Sovereignty & Privacy	15%	Raw context must not egress to cloud APIs for search

Total: 100%

A system MUST score at least 3.0 in Local-First Operability and CRDT Resolution to be considered for production use.

Part IV. Evaluation Categories

4.1 Local-First and Offline Operability (Weight: 20%)

4.1.1 Read/Write Autonomy

Can the system read and write semantic state without network connectivity?

Score	Criteria
0	Requires network for all reads and writes
1	Read cache exists, but writes require network
2	Local reads/writes, but manual sync required
3	Automatic local-first read/write with background sync (e.g., `sqlite-vec`)
4	Local-first with predictive prefetching and background re-indexing
5	Perfect autonomy: formally verified offline operability, zero-latency local vector search, automated conflict queue

4.1.2 Sync Engine Architecture

How is state synchronized between local and remote?

Score	Criteria
0	Direct database replication (fragile, network-dependent)
1	API-based sync with timestamp conflict resolution
2	API-based sync with basic transactional retry
3	Dedicated local-first sync engine (ElectricSQL/PowerSync)
4	Sync engine with bandwidth-optimized vector delta transmission
5	Perfect sync: formally verified convergence, zero data loss on partition, sub-second sync on reconnect

4.2 CRDT and Semantic Conflict Resolution (Weight: 20%)

4.2.1 Conflict Resolution Mechanism

How are conflicting concurrent updates resolved?

Score	Criteria
0	Last-write-wins (destroys semantic context)
1	Manual conflict resolution required
2	Basic CRDTs for scalar data, but vectors are overwritten
3	Full CRDT support for all data types, including vector metadata
4	Semantic merge: LLM-assisted resolution of conflicting memory/context states
5	Perfect resolution: formally verified CRDTs, deterministic semantic merge, zero-state-loss on any network topology

4.3 Vector Indexing and Retrieval Quality (Weight: 15%)

4.3.1 Indexing Algorithm

Does the system use state-of-the-art, production-proven indexing?

Score	Criteria
0	Brute-force / exact search only (unscalable)
1	Basic IVF (Inverted File Index)
2	HNSW (Hierarchical Navigable Small World) with default parameters
3	Tunable HNSW or DiskANN with metadata pre-filtering
4	Hardware-accelerated indexing (GPU/NPU) with quantization support
5	Perfect indexing: formally verified recall/precision bounds, adaptive indexing based on hardware, zero-latency local inference

4.4 Embedding Lifecycle and Drift Governance (Weight: 15%)

4.4.1 Model Versioning

Are embedding models treated as a strict schema?

Score	Criteria
0	No model versioning; vectors from multiple models mixed
1	Model version recorded, but no enforcement
2	Vectors partitioned by model version
3	Automated re-indexing pipeline triggered on model change
4	Blue/green re-indexing with A/B testing of retrieval quality
5	Perfect governance: formally verified vector spaces, automated drift detection, zero-downtime zero-dual-write re-indexing

4.5 Temporal Memory and Knowledge Graphs (Weight: 15%)

4.5.1 Fact Temporality

Does the system track the temporal validity of facts (when facts became true/false)?

Score	Criteria
0	Static facts only; no time context
1	Timestamps exist but are not used for retrieval
2	Time-decay scoring applied to search results
3	Full Temporal Knowledge Graph (fact start/end dates, contradiction resolution)
4	Memory ACLs, trust classes, and user-editable temporal state
5	Perfect temporal design: formally verified time-travel queries, automated stale-fact quarantine, cryptographic provenance for all memories

4.6 Data Sovereignty and Privacy (Weight: 15%)

4.6.1 Egress Control

Does the system prevent raw context from leaving the local boundary?

Score	Criteria
0	All search queries sent to cloud vector API
1	Local search, but raw data synced to cloud unencrypted
2	Local search with TLS encryption in transit
3	Local search with envelope encryption (KMS) at rest

Score	Criteria
4	Zero-knowledge search; only encrypted vectors or proofs leave the device
5	Perfect sovereignty: fully homomorphic encrypted (FHE) search capabilities, formally verified zero raw context egress

Part V. The Mythos Data & Sync Suite (MDSAS)

Traditional database benchmarks measure QPS (Queries Per Second). The MDSAS evaluates offline resilience, semantic conflict resolution, and drift governance.

5.1 Suite Composition

5.1.1 MDSAS-Partition

- Network Partition: 100+ tests simulating network drop, concurrent local writes, and reconnection to verify CRDT convergence.
- Conflict Injection: 50+ tests injecting semantically conflicting memories to verify LLM-assisted merge logic.

5.1.2 MDSAS-Drift

- Model Swap: 50+ tests triggering an embedding model upgrade to verify re-indexing pipelines and A/B recall validation.
- Temporal Decay: 50+ tests querying for facts whose validity has expired to verify they are omitted or quarantined.

5.2 Execution Protocol

```
1. Deploy local-first architecture across multiple simulated edge nodes
2. Run MDSAS suite (automated partition + drift injection)
3. Score each category 0-5
4. Check for last-write-wins sync algorithms (instant disqualification)
5. If all thresholds met: approve for production
```

Part VI. Security Threat Model for Data & Sync

6.1 Threat Catalog

6.1.1 Memory Poisoning via Sync

Severity: Critical Description: A compromised edge node syncs malicious or hallucinated memories to the enterprise knowledge graph, corrupting the global agent context.

Required Controls:

1. Cryptographic signing of all synced state updates.

2. Trust classes for memory (e.g., user-stated vs. agent-inferred).

3. Quarantine and verification pipeline for cross-scope memory promotion.

6.1.2 Embedding Inversion Attack

Severity: High Description: An attacker exfiltrates the vector database and uses an embedding inversion model to reconstruct the original raw text (PII/IP).

Required Controls:

1. Local-first architecture minimizing attack surface.

2. Envelope encryption of vectors at rest.

3. Zero-knowledge sync (syncing only deltas or encrypted blobs).

6.1.3 Vector Drift Degradation

Severity: High Description: An unplanned update to the cloud embedding API silently changes the vector space, causing the local agent's retrieval quality to plummet to zero.

Required Controls:

1. Pinning embedding model versions (no auto-updates).

2. Automated recall testing before model migration.

3. Strict separation of vectors generated by different models.

Part VII. The Mythos Data & Sync Standard

7.1 The Mythos Data Doctrine

```
A cloud database is a cache, not a source of truth.
A vector that cannot sync offline is ephemeral.

A semantic conflict cannot be solved by a timestamp.
An embedding model is a schema; changing it without migration is corruption.

Memory may be fuzzy.
State convergence must be absolute.
```

7.2 Selection Rules

1. No data architecture enters production without passing MDSAS.
2. No architecture is approved if it requires network for local reads/writes.
3. No architecture is approved if it uses last-write-wins for conflict resolution.
4. No architecture is approved without an embedding lifecycle governance plan.
5. No architecture is approved if raw context egresses the local boundary for search.

7.3 The Prime Directive for Data & Sync Architecture

> Sync the state, not the query. > Resolve the semantics, not the timestamp. > Govern the embedding, not just the vector. > Isolate the memory, not just the database.

```
Academic benchmarks measure queries per second.
Applied benchmarks measure offline operability.
Security benchmarks measure egress control.
Operational benchmarks measure CRDT convergence.

> Networks may partition.
> State must be absolute.
```

End of Standard MYTHOS-DAT-0001

End of controlled publication

MYTHOS-DAT-0001 remains a Candidate Standard until technical review, security and privacy review, accessibility review, current-source validation, and formal approval are recorded.