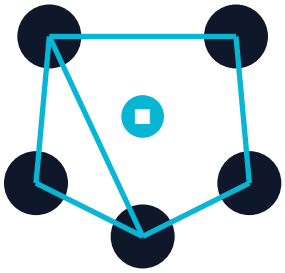


CHE·NU
KNOWLEDGE THREADS
INTER-SPHERE CONNECTIONS

CORE.v1.0 — Foundation / Context Linking / Non-Inferential

Knowledge Threads LINK facts.
They NEVER interpret, judge, summarize, or optimize.

Thread = POINTER, not narrative.

Status: CANONICAL

— 1 —

PURPOSE

Connect related knowledge across spheres, meetings, decisions, artifacts, agents, and time WITHOUT merging meanings or generating conclusions.

What a Knowledge Thread IS

- An explicit, traceable connection
- Between existing memory objects
- Created by reference, not inference
- **Thread = POINTER, not narrative**

What a Knowledge Thread is NOT

- ■ NOT a summary
- ■ NOT an interpretation
- ■ NOT a judgment
- ■ NOT an optimization

— 2 —

THREAD SOURCES

Threads MAY originate from:

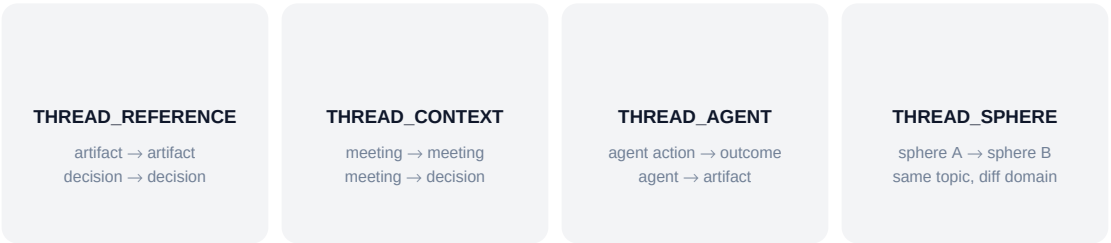
- Shared artifacts
- Reused documents
- Referenced decisions
- Recurring topics (explicitly tagged)
- Manual user linking
- Explicit agent citation (logged)

Threads NEVER originate from:

- ■ Sentiment analysis
- ■ Pattern guessing
- ■ Hidden correlations
- ■ Inferred relationships

— 3 —

THREAD OBJECT TYPES



Type	Source → Target	Use Case
THREAD_REFERENCE	artifact → artifact, decision → decision	Direct references
THREAD_CONTEXT	meeting → meeting, meeting → decision	Contextual links
THREAD_AGENT	agent action → outcome, agent → artifact	Agent activity tracing
THREAD_SPHERE	sphere A → sphere B	Cross-domain connections

— 4 —

THREAD CREATION RULES

Rule	Description	Enforcement
Append-only	New threads only	No modifications
Timestamped	Creation time recorded	Automatic
Reversible	Unlink allowed, never delete	Audit trail
Visible	Source & target always shown	Transparency
Explained	Reason required	Mandatory field

■■ No Silent Threads Allowed

Every thread must have an explicit, human-readable reason for its existence.

— 5 —

KNOWLEDGE THREAD JSON MODEL

```
{
  "knowledge_thread": {
    "id": "uuid",
    "type": "reference|context|agent|sphere",
    "source": {
      "id": "uuid",
      "object_type": "meeting|decision|artifact|agent"
    },
    "target": {
      "id": "uuid",
      "object_type": "meeting|decision|artifact|agent"
    },
    "created_by": "user|agent_id",
    "reason": "explicit_text_description",
    "timestamp": 1712345678,
    "visibility": "private|shared|public",
    "hash": "sha256"
  }
}
```

— 6 —

THREAD VISUALIZATION

Visual Rules

- Thin neutral lines (no emphasis)
- No color grading for importance
- Optional highlight on hover
- Toggle per thread type

Cluster Behavior

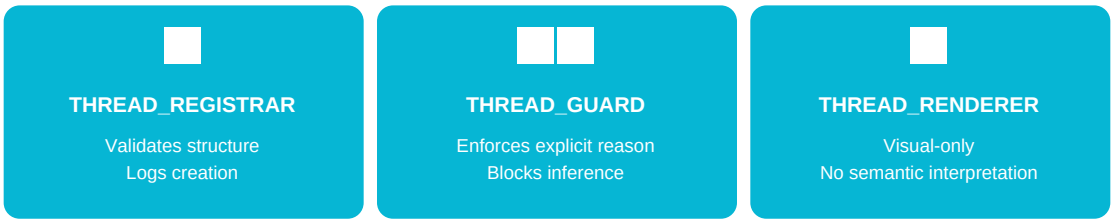
- Threads may pull nodes closer
- NEVER collapse nodes automatically
- User controls clustering

Navigation Actions

Action	Description
follow_forward	Navigate to target
follow_backward	Navigate to source
expand_network	Show connected threads
isolate_path	Focus on single thread chain
open_linked_replays	Access related replays

— 7 —

THREAD AGENTS



Agent	Responsibility	Constraint
THREAD_REGISTRAR	Validates structure, logs creation	Structure only
THREAD_GUARD	Enforces explicit reason	Blocks implicit inference
THREAD_RENDERER	Visual display	No semantic interpretation

— 8 —

ACCESS & PRIVACY

- Respects original object visibility
- Private source → private thread
- Shared thread requires shared source
- Public thread requires explicit consent

— ∞ —

WHY THIS MATTERS

Knowledge Threads transform CHE·NU into a living archive, not an opinion engine.

Knowledge Threads Allow	Knowledge Threads Prevent
Continuity without rewriting history	Narrative manipulation
Cross-domain understanding	Hidden correlations

Memory without manipulation	Inferred intent
Explicit connections	Silent linking

Thread = POINTER, not narrative.
Link facts. Never interpret.