

XEMATIX v1 Spine

Canonical Foundation for the Category of Pre-Execution Semantic Control

John Deacon

Document Metadata

Identifier	XEMATIX-V1-SPINE
Version	1.0.0
Status	Canonical (Frozen)
Publication Date	January 2026
ORCID	https://orcid.org/0009-0008-1000-1898
License	CC BY-NC-SA 4.0
Canonical URL	https://xematix.org/v1
Primary Role	Constitutional core of the XEMATIX category
Derived From	XEMATIX Foundation Paper (category thesis) and XEMATIX-SPEC-001 (normative spec)

Change Log

Version	Date	Notes
1.0.0	January 2026	Initial canonical spine publication. Frozen for 12 months except errata/clarifications.

Abstract

XEMATIX defines a new architectural category: a *pre-execution semantic control layer* that governs how human intent is encoded, constrained, validated, and delegated to machine systems *before* any automated action occurs. XEMATIX introduces an explicit upstream boundary where human judgment terminates and machine execution begins, preventing proxy-goal drift and preserving accountability lineage. This document is the canonical, frozen core of XEMATIX v1: the irreducible minimum required to define the category and test compliance.

1 Canonical Declaration

XEMATIX defines a missing layer in modern intelligent systems: a *pre-execution semantic control layer* between human intent and machine execution. It exists to answer the upstream question modern AI stacks do not enforce:

What exactly did the human mean — and under what constraints — before the machine was allowed to act?

XEMATIX does not compete with models, agents, orchestration tools, or automation platforms. It precedes them. A system is “XEMATIX-aligned” only if it enforces an intent boundary and governs delegation prior to execution.

2 Conformance Language

This document uses the normative keywords **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT**, and **MAY** to indicate requirement levels. Requirements in this spine define the constitutional minimum for category compliance.

3 Purpose

A XEMATIX-aligned system **MUST** provide a mechanism by which human intent is:

1. explicitly articulated,
2. structurally constrained,
3. semantically validated, and
4. governed prior to execution by any automated system.

No probabilistic model, agent framework, or workflow engine **MAY** infer, reinterpret, or extend intent as authorization. Execution **MUST NOT** proceed without an authorized, validated intent model.

4 Scope Boundary

4.1 In Scope (v1)

This v1 spine defines the constitutional minimum:

- the human–machine intent boundary and enforcement rules,
- the five-layer semantic control pipeline (Anchor, Projection, Pathway, Actuator, Governor),
- CAM as the required mission-to-action alignment structure,
- ALOs as versioned semantic governance objects,
- pre-execution ambiguity handling and constraint satisfaction requirements,
- traceability and audit requirements (semantic ledger of intent),
- baseline failure modes and mitigations relevant to pre-execution control.

4.2 Out of Scope (v1)

This v1 spine **MUST NOT** be interpreted as defining:

- model architectures, training methods, or inference mechanisms,
- AGI, autonomy, or consciousness frameworks (XEMATIX is an intent-governance layer),
- product UI, platform design, or a specific implementation stack,
- domain-specific commercial use cases (these are downstream reference implementations),

- normative moral theory or political/regulatory frameworks (v1 is structural/architectural).

Any system claiming “XEMATIX” while bypassing explicit human intent or asserting machine judgment across the intent boundary is non-compliant.

5 Core Terms (v1)

5.1 Intent

A human-declared commitment specifying purpose, outcomes, constraints, and responsibility that authorizes delegation. Intent **MUST** be captured prior to automation and represented in CAM.

5.2 Execution

Mechanical transformation, computation, inference, and actuation performed after intent is authorized. Execution does not include judgment, prioritization, or moral responsibility.

5.3 Semantic Control Layer

The upstream layer that validates, constrains, and governs intent prior to execution, preserving meaning and rationale through the pipeline.

5.4 ALOs (Abstract Language Objects)

Versioned semantic governance objects encoding knowledge, constraints, policies, and contextual content, linkable to CAM and auditable.

5.5 CAM (Core Alignment Model)

A structured alignment schema encoding Mission, Vision, Strategy, Tactics, and Reflective Governance (meta-level integrity).

6 The Human–Machine Intent Boundary

6.1 Boundary Definition

The intent boundary is the point where human cognitive activity terminates and machine execution begins. Human cognition includes sensemaking, interpretation, prioritization, and responsibility. Machine execution includes computation, probabilistic inference, transformation, and actuation.

Machines **MUST NOT** cross this boundary by performing judgment or originating goals.

6.2 Boundary Enforcement Requirements

A XEMATIX-aligned system **MUST** enforce the intent boundary by:

1. requiring explicit intent declaration prior to any execution (purpose, outcome, constraints),
2. blocking execution absent a validated intent model,
3. rejecting inferred/speculative intent as authorization,
4. treating ambiguity as a blocking condition until resolved via user confirmation or explicit placeholders.

6.3 Boundary Violations (Informative)

Violations include scope expansion (“helpful drift”), substituting machine priorities for human priorities, inventing objectives, or proceeding under ambiguous intent without explicit confirmation.

7 The Five-Layer Semantic Control Pipeline

XEMATIX **MUST** implement five layers in order. Reflective iteration is permitted under governance, but bypass is not.

7.1 Anchor (Intent Declaration)

Anchor **MUST** capture explicit human intent: purpose, values, constraints, and accountable originator(s). No downstream processing **MAY** occur without a validated Anchor.

7.2 Projection (Outcome Framing)

Projection **MUST** define a desired future state and success criteria. Projection **MUST** function as a constraint during planning and oversight.

7.3 Pathway (Semantic Orientation)

Pathway **MUST** define the strategy linking current state to desired outcome under mission constraints. The system **SHOULD** evaluate alternative pathways and prefer those most consistent with CAM and ALO constraints. Every pathway **MUST** trace back to Anchor and Projection.

7.4 Actuator (Execution Interface)

Actuator **MUST** produce concrete steps required to implement the pathway (e.g., calls, prompts, transforms, actuation). Actuator **MUST NOT** modify intent or expand scope. Each step **MUST** carry rationale references to the CAM/ALO elements that motivated it.

7.5 Governor (Oversight and Integrity)

Governor **MUST** monitor outcomes against Projection and Anchor, detect deviations and violations, and be capable of pausing, replanning, or halting execution. Governor **SHOULD** drive reflective updates when assumptions fail.

8 CAM Requirements (Constitutional)

8.1 Required Elements

CAM **MUST** encode, at minimum:

- Mission / Purpose (Anchor-aligned)
- Vision / Outcome (Projection-aligned)
- Strategy / Pathway (Pathway-aligned)
- Tactics / Actions (Actuator-aligned)
- Reflective Governance (meta-integrity, assumptions, risk, coherence)

8.2 Auditability and Versioning

CAM **MUST** be explicit, structured, versioned, and auditable. Updates **MUST** include: what changed, who authorized it, and why.

8.3 CAM as Drift Yardstick (Informative)

Actions and plans not strengthening the Mission-to-Tactics chain are potential drift and **SHOULD** be flagged for review.

9 ALO Requirements (Constitutional)

9.1 ALO Definition

An ALO is a structured semantic object encapsulating knowledge, policies, constraints, context, or content relevant to fulfilling intent. ALOs **MUST** be machine-interpretable, linkable to CAM, versioned, and auditable.

9.2 ALO Properties

ALOs **MUST** be:

1. structured and modular (hierarchical sub-objects supported),
2. linkable to CAM,
3. queryable for reasoning and constraint satisfaction,
4. versioned with provenance and lifecycle governance.

9.3 ALO Lifecycle Governance

Implementations **MUST** support lifecycle operations (create, modify, deprecate, delete) with author, timestamp, rationale, and CAM impact linkage. Critical ALOs **SHOULD** require stronger change control (e.g., multi-party approval).

10 Semantic Interpreter and Orchestrator Requirements

10.1 Semantic Interpreter

The Semantic Interpreter **MUST** convert unstructured/semi-structured input into explicit updates to CAM and ALOs. It **MUST** be schema-aware and **MUST NOT** silently assume meaning under ambiguity.

10.2 Ambiguity Handling Rule

Ambiguity **MUST** be treated as a blocking condition until resolved through:

- clarifying questions, or
- explicit placeholders flagged for confirmation, or
- explicit user approval of an interpretation.

11 Planning, Reasoning, and Constraint Satisfaction

11.1 Semantic Alignment Evaluation

Candidate plans **MUST** be evaluated against:

- CAM constraints (mission-to-tactics coherence), and
- ALO constraints (domain policies, exclusions, requirements).

Plans violating constraints **MUST** be rejected.

11.2 Rationale-Carrying Plans

Reasoning outputs **SHOULD** include rationale such that each step can be justified in relation to declared purpose, outcome, strategy, and constraints.

12 Executable Semantics and Traceability

12.1 Semantic Metadata Through Execution

Each execution directive **MUST** include references to the CAM/ALO elements that motivated it. If downstream components are not XEMATIX-aware, XEMATIX **MUST** still log rationale and verify outcomes against constraints.

12.2 Semantic Ledger of Intent

Implementations **MUST** maintain a persistent semantic ledger linking every action to:

- the originating intent declaration,
- the relevant CAM state/version,
- the consulted ALO versions,
- the constraint set in effect,
- the executed step(s) and outcome evidence.

13 Governance, Accountability, and Auditability

13.1 Provenance and Versioned Alignment State

XEMATIX **MUST** maintain intent provenance and a versioned alignment state, enabling audit of how and why actions occurred.

13.2 Change Control

CAM and ALO changes **MUST** be logged with timestamp, author, justification, and impact.

13.3 Governor Enforcement

Governor **MUST** have authority to pause, halt, or re-plan when violations or deviations are detected.

14 Reflective Governance (Metacognitive Semantics)

XEMATIX **MUST** maintain meta-knowledge about reasoning, assumptions, priorities, and known risks, enabling introspection on contradictions between actions and declared values and triggering

corrective updates when required.

15 Failure Modes and Minimum Mitigations (v1)

15.1 Misinterpretation of Intent

The system **MUST NOT** silently misinterpret ambiguous input. When uncertainty exists, it **MUST** ask clarifying questions or present its understanding for approval prior to execution.

15.2 Proxy Goals, Ambiguity Collapse, Responsibility Drift (Informative)

Absent an upstream semantic layer, delegation increases the likelihood of proxy-goal pursuit, ambiguity collapse, untraceable actions, and silent drift. XEMATIX exists to preempt these failure modes via explicit pre-execution governance.

16 Derivative Specs Contract (v1)

Derivative specifications **SHOULD** reference this spine as the canonical foundation and **MUST** remain consistent with its principles. Extensions **MUST** be published as:

- v1.x supplements (non-breaking additions), or
- v2.0 proposals (breaking/expansive changes) with explicit compatibility discussion.

17 Versioning Policy (Freeze Contract)

This v1 spine is frozen as the category core for 12 months from publication, with changes limited to errata and clarifications. No new layers, mandatory objects, or scope expansions are permitted under v1.

18 Compliance Test (Minimum, Non-Negotiable)

A system may claim “XEMATIX-aligned” only if it can demonstrate:

1. Intent boundary enforcement (execution blocked without validated explicit intent).
2. Five-layer pipeline implementation (Anchor → Projection → Pathway → Actuator → Governor) with no bypass.
3. Explicit, versioned, auditable CAM governing decisions.
4. Versioned, governed ALOs linked to CAM and used in reasoning/constraint satisfaction.
5. A semantic ledger of intent: each execution step is traceable to mission/vision/strategy/constraints and the ALOs consulted.

19 Closing Principle (Category Anchor)

Machines may execute. Only humans may decide.