

EARMARK

MACHINES FOR THINKING

STRUCTURAL OBLIGATIONS

THE SIX REQUIREMENTS FOR CORPUS PORTABILITY

EARMARK OPEN INTELLIGENCE PROTOCOL

AUTHOR: MIKHAIL SHAKHNAZAROV

BERLIN, FEBRUARY 2026

LICENSE: CC BY 4.0

<https://creativecommons.org/licenses/by/4.0/>

OVERVIEW // 01

A governed corpus must satisfy six structural obligations to be portable across runtimes and durable across time. These obligations are not optional features or best practices; they are the minimum conditions for the corpus to function as a specification rather than a collection of prompts. A corpus that fails any obligation degrades along a predictable axis: portability collapses, verification becomes impossible, or governance erodes into convention without enforcement.

OBLIGATION 1: SECOND-ORDER TOKENS // 02

Artifacts must carry their own compilation instructions. A second-order token fuses content, commentary, and rules: what is being produced, what obligations it must satisfy, and how to verify it. This is how the corpus becomes portable across tools and time: the artifact arrives with its own constraints, not merely its content.

A template is the simplest second-order token: a fixed structure plus constraints on tone, scope, and allowed evidence. A procedure is the next step: a named transform with inputs, outputs, and acceptance criteria. Both are self-compiling in the sense that any competent runtime can execute them as long as the instructions are present.

Failure mode

Shipping content without obligations. If second-order tokens degrade into phrasing preferences, portability collapses and outputs drift back to generic vendor defaults.

OBLIGATION 2: ARTIFACT COORDINATES // 03

Every durable artifact must declare its position in a constrained space through a small set of coordinates: authority, audience, verification mode, context source, time horizon, and governance cost. Coordinates are not theory; they are routing metadata. They decide which template applies, what evidence is required, what tone is permitted, and what kind of acceptance test closes the loop. The coordinate system is defined in a separate specification document.

Failure mode

Axis proliferation. The coordinate system must remain small enough to stay usable. Add axes only when they create real routing power and reduce

confusion.

OBLIGATION 3: COMPACTION TRANSFORMS // 04

High-entropy material -- transcripts, session logs, extended conversations -- must be reducible to governed artifacts through explicit transforms. Compaction is a named procedure with defined inputs, outputs, and acceptance criteria. The output preserves decisions, definitions, open questions, and decision traces; it does not import raw data into the kernel.

Compaction is the intelligence function of the corpus: it monitors what is happening, identifies what needs to change, and proposes adaptations for the governance layer to accept or reject. Without compaction, the corpus grows without structure and context windows overflow without navigation.

Failure mode

Raw material imported directly into governed space without passing through a compaction procedure. The airlock rule exists to keep untyped text quarantined until it is curated.

OBLIGATION 4: EPISTEMIC GOVERNANCE // 05

The corpus must maintain a visible distinction between settled and provisional content. The roman/italic convention is the minimum viable implementation: roman text is treated as compiled output, binding within the corpus; italic text is commentary, carrying assumptions, uncertainty, missing evidence, and candidate improvements. Promotion from italic to roman requires resolution -- a decision, a test, or an imported piece of evidence. The convention is defined in a separate specification document.

Failure mode

Collapsing the layers. If italic becomes rhetorical decoration rather than epistemic status, the system loses its smallest and cheapest drift-control mechanism.

OBLIGATION 5: INTRINSIC SIGNAGE // 06

Governed artifacts must carry embedded verification patterns derived from their own content. Intrinsic signage uses style variation points -- choices that preserve meaning while changing surface form -- as deterministic dials set by a hash of the canonical text. The result is text that signals when it has changed from its marked state, without requiring external metadata,

container signatures, or institutional trust.

The mechanism is drift detection, not cryptographic security. It makes verification symmetric and path-independent: the signature is content-derived, not process-derived. The mechanism is defined in a separate specification document.

Failure mode

Treating intrinsic signage as tamper-proofing. It is not. An informed adversary can re-mark text. The primitive is: style rules become mechanically enforceable typing constraints that survive transport through channels that strip container metadata.

OBLIGATION 6: CROSS-RUNTIME VERIFICATION // 07

The corpus must be testable across different language model runtimes. Cross-runtime verification is the ultimate portability test: if the governed corpus produces consistent, verifiable output when executed by different models, the governance is operational rather than vendor-specific.

Verification has two modes. Deterministic verification: the same pipeline produces identical outputs and signatures on re-runs, regardless of runtime. Convergent verification: different processes arriving at the same content produce matching signatures. Both modes must be supported.

Failure mode

Conflating the runtime with the language. If the corpus is thin and the model's default priors leak in as if they were the spec, cross-runtime verification will fail because the spec was never explicit enough to be portable.

CC BY 4.0 -- Mikhail Shakhnazarov, Berlin, February 2026