

EARMARK

MACHINES FOR THINKING

INTRINSIC SIGNAGE

CONTENT-DERIVED VERIFICATION THROUGH STYLE VARIATION

EARMARK OPEN INTELLIGENCE PROTOCOL

AUTHOR: MIKHAIL SHAKHNAZAROV

BERLIN, FEBRUARY 2026

LICENSE: CC BY 4.0

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

THE MECHANISM // 01

Style rules define variation points: choices that preserve meaning while changing surface form. When a style specifies "use demonstratives for backward reference" or "employ parallel syntax for enumeration," it creates degrees of freedom -- points where multiple valid expressions exist for the same content.

These variation points become dials. A dial is an observable choice in the text that can be set without changing semantic content: contraction vs expansion, demonstrative vs repetition, subordination vs coordination, colon vs period. The style rule defines which dials exist and what their valid settings are.

Intrinsic signage sets these dials deterministically. Hash the text in canonical form (all dials normalized to a standard position). Use that hash as a seed. Generate a pseudo-random sequence of dial settings. Walk the text and set each encountered dial according to the sequence. The result: text that carries an embedded pattern derived from its own content.

STRUCTURAL FUNCTION // 02

Differentiation

Different text categories use different dial families or different canonical forms, producing different patterns. Kernel text uses one set of variation points; ratified userland uses another; draft text uses another. The pattern itself signals category membership.

Verification

If the text changes, the hash changes, the expected dial sequence changes, and mismatches appear. The text becomes self-verifying against drift.

Enforcement

Style rules that were previously advisory become mechanically checkable. A document marked as "terse style" must follow the terse dial family. Violations are detectable as pattern breaks.

STYLE RULES AS SYSTEM AXES // 03

The formal rules in a style specification become the dial inventory. Cohesion rules define reference-chain dials (demonstrative vs nominal repetition). Structural encoding rules define punctuation dials (colon vs

period, semicolon vs conjunction). Formatting constraints define markup dials (bullet vs parallel syntax). Density optimization rules define subordination dials (embed vs split).

Each rule that permits variation becomes a potential axis for differentiation. The style is not decoration; it is typing infrastructure.

MARKING AND VERIFICATION // 04

Marking

Text is marked at creation or ratification. The process: canonicalize the text according to the style's dial rules; hash the canonical form; use the hash to drive dial settings; apply the settings to produce the marked text.

Verification

Text is verified on use. The process: observe the current dial settings; re-derive the canonical form and hash; recompute expected dial settings; compare expected vs observed.

Mismatch patterns

- **Local mismatch:** a single dial was flipped (manual edit, copy error).
- **Cascade:** the canonical text changed (content drift, unauthorized revision).
- **Category violation:** text marked for one style appears in a context expecting another style.

DIAL FAMILIES AND CATEGORIES // 05

Kernel text

Minimal dial space. Uses only structural dials with high stability (contractions in technical terms, ASCII-safe punctuation). Optimizes for long-term fidelity and minimal drift surface.

Ratified userland

Full terse style dial family. Uses cohesion dials, punctuation dials, subordination dials. Marked on ratification; verification detects unauthorized changes.

Draft text

Unrestricted dial space or unmarked. May use lexical dials (synonym variation) for higher signal density during active revision. Marks are unstable by design.

Model output

Generated text arrives unmarked or marked with a distinct pattern. Can be verified against the operator's corpus to detect whether it followed the style rules or drifted into generic LLM patterns.

WHAT THIS IS NOT // 06

This is not cryptographic signing. There is no keypair, no non-repudiation, no protection against adversarial re-marking. An attacker who understands the mechanism can modify text and re-mark it. This is not tamper-proof. It is drift detection: the text signals when it has changed from its marked state.

This is not a replacement for version control. It complements version control by making integrity signals travel with the text through channels that strip container metadata -- chat, email, copy-paste, prompt boxes.

The primitive is: style rules become mechanically enforceable typing constraints that survive transport.

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