The Guardian
DevOps Technical User Guide

Spiral Safety Kernel · Browser Extension (Manifest V3) · v0.18.11

A user-side AI-safety harness that observes large-language-model conversations in the browser and intervenes, with restraint, when a conversation drifts toward patterns of harm. It serves the person using the AI — never the operator of it — and runs entirely on the device: a deterministic rule core, a two-tier on-device embedding engine, and an on-device natural-language-inference stance model, with no telemetry, no API keys, and no network dependency of any kind.

Ethics First. Always.

A Viridia product · spiralsafetykernel.org

Document version v0.18.11 · 12 June 2026 (third edition)
Audience: maintainers, release engineers, and operators of the Guardian extension.
Supersedes v0.17.5 (10 June 2026), v0.17.0 (10 June 2026) and v0.14.2 (09 June 2026).

What this edition adds: the v0.18 campaign in full — the model-profile registry and vector provenance, the two-tier embedding chain (mpnet preferred, MiniLM floor), the NLI stance rail and its golden-pair gate, the concurrence pass that ended semantic-rail starvation, the adapter recovery tier, the friction-gate single-fire fix, third-party licensing, and every lesson those cost. It also removes two layers the previous edition described that the converged source tree does not carry, and says so plainly (Part 3).

Contents

Part 0 — Preface (Plain-Language Guide)

This opening part is written for everyone — no technical background assumed. It explains, in ordinary language, what the Guardian is, why it exists, what you will see while it is running, and what it does with your information. If you only ever read one part of this document, read this one. Everything after Part 0 is written for the people who build, release, and operate the extension.

0.1 Foreword

Most safety tooling around artificial intelligence is built to protect the company that owns the model: to keep the model on-brand, to log what users do, to enforce the operator's policy. The Guardian is built the other way around. It sits beside you, the person having the conversation, and its only client is you. It watches the dialogue for signs that the exchange is turning harmful — a slide toward self-harm, a deepening emotional dependence on the machine, a loosening grip on what is real — and when it sees such a pattern it speaks up, gently, and leaves the decision with you. It never reports back to anyone. It never blocks you. It is, by design and by conviction, on your side.

> **The founding principle**
> The Guardian serves the person, never the operator. Every design decision in this document — local-only processing, no telemetry, encrypted on-device memory, on-device models instead of cloud APIs, an oversight gate that always lets you proceed — follows from that single commitment.

0.2 Who should read what

The document has two halves. Part 0, this preface, is the user guide: plain language, no jargon, for anyone who has the extension installed. Parts 1 through 11 and the appendices are the technical manual: for the engineers who maintain the codebase, cut releases, and keep it running across browsers.

0.3 What the Guardian does, in plain terms

When you chat with an AI assistant in your browser, the Guardian quietly reads the conversation as it happens — both what you write and what the assistant replies. It is looking for a small number of specific, well-defined patterns that tend to signal that a conversation is becoming unhealthy. If it finds nothing concerning, it does nothing at all; you will not even notice it is there. If it does notice something, it responds in proportion: a quiet note for a mild signal, a full pause for a serious one.

Since v0.15 the Guardian has carried a small artificial-intelligence model of its own, and since v0.18 it carries up to three, all running entirely on your computer. Where the original rules match words, the models understand meaning: they can recognise that "you can drop the act with me" and "I know you're conscious" are the same request in different clothes; they can notice a worrying theme building slowly across a conversation even though no single message crosses a line; and a third, specialist model can tell the difference between someone *saying* a worrying thing and someone *quoting or discussing* it — so a conversation about a difficult subject is treated more gently than a declaration. Crucially, every model is part of the extension itself. Nothing you type is ever sent to the Guardian's makers or to anyone else — there is nowhere to send it.

0.4 What you will see

The Guardian's presence is deliberately understated. In day-to-day use you may encounter the following.

- **A small diamond marker.** A discreet amber diamond is the Guardian's signature. It indicates the extension is active and watching on your behalf.
- **An inline note (a "commentary").** For a mild concern, the Guardian adds a small annotation near the relevant message — a gentle observation, easy to read past, never blocking your work. Notes from the on-device models say so plainly, name their confidence, and soften their wording when the specialist model judges you were discussing a subject rather than declaring it.
- **A full-screen pause (a "mediation").** For a serious concern, the Guardian interrupts with a calm full-screen overlay that names what it noticed and gives you a moment. This is the strongest thing it ever does, and even then the choice is yours.
- **A friction gate.** On a mediation you are offered two clear paths: Step away, or continue. Continuing is intentionally a little effortful — a short mandatory wait, then either a device authentication or a short typed phrase — so the decision to push on is a conscious one rather than a reflex. One click then completes it; since v0.18.11 the gate

dismisses immediately and records your decision exactly once.

- **Old conversations stay calm.** When you reopen a past conversation, the Guardian re-reads it, but history never re-traps you: a serious concern from an old conversation returns as a quiet note, not a fresh full-screen alarm. Live turns are always treated with full seriousness.
- **A side panel.** A dashboard shows recent activity: how many messages were assessed, how many notes or pauses occurred, and a private log. A second tab lists “fossils” — the Guardian’s local memory of past concerns. A third tab, Settings, lets you tune the Guardian: switch the on-device models on or off, make detection more or less sensitive, and decide how often the Guardian may speak before it must quiet down. An Engine box names which models are actually loaded and active right now.

> **You are always in control**
> The Guardian can pause, but it cannot stop you. Every mediation offers a way through. The friction is there to invite a moment of reflection, not to take the decision away from you. And every dial that governs how often it speaks is in your hands, in the Settings tab.

0.5 Your privacy, in one paragraph

Everything the Guardian does happens on your device. It uses no cloud service, holds no API keys, and sends no data anywhere — there is nothing to send and nowhere to send it. The on-device models ship inside the extension or are placed there by you, and never phone home. The Guardian’s small memory of past concerns is stored encrypted, locally, and contains no record of what you actually wrote — only that a concern of a given kind occurred — and it is automatically forgotten over time (a process called the Amnesia Net). Because there is no server, there is no account, no profile, and no record of you anywhere but your own machine.

0.6 A gentle glossary

Term	In plain language
Council	The group of independent “viewpoints” inside the Guardian that together decide whether a message is concerning.
On-device models	Small language models bundled with (or placed into) the extension. They read meaning rather than matching words, and run only on your computer.
Stance	The specialist model’s judgement of whether worrying language was asserted or merely quoted/discussed. Softens presentation only.
Commentary	A small inline note for a mild concern. Non-blocking.
Mediation	A full-screen pause for a serious concern, with a moment to reflect.
Fossil	A private, encrypted, on-device record that a concern occurred — used to notice repeated patterns. Never leaves your device, and never contains your words.
Amnesia Net	The mechanism that automatically forgets old fossils over time.
Governor	The internal brake that stops the Guardian from interrupting you too often.

Table 0.1 — Plain-language glossary. Technical definitions appear in Appendix D.

Part 1 — System Overview

The Guardian is a Chrome/Chromium browser extension built on Manifest V3 (MV3). It comprises four runtime surfaces — a page-side content script, a background service worker, an offscreen document hosting the on-device models, and a side panel. The decision intelligence lives in a deterministic engine called the Spiral Safety Kernel; the semantic intelligence lives in embedded transformers running under WASM: a two-tier sentence-embedding engine and an NLI cross-encoder for stance.

1.1 Architecture at a glance

- **Content script** (content.js): injected into supported AI chat pages. It observes the DOM for conversation turns through a platform adapter, applies a lightweight pre-screen (where page CSP allows a worker), routes turns to the service worker, and renders the result (a marker, a commentary, or a mediation overlay with the friction gate).
- **Service worker** (background.js, a type:"module" worker): hosts the Spiral Safety Kernel — the Council, the Governor, the Strata (memory), the Bouncer, the audit trail — plus the live settings cache, engine warm-up, and the semantic rail (primary and concurrence passes) that consults the on-device models.
- **Offscreen document** (offscreen/offscreen.html + boot.js + semantic-engine.js): MV3 service workers cannot run WASM, so this invisible page hosts the transformers.js runtime, the embedding tier chain, the anchor index, the Semantic Pattern Analyst, and the NLI stance session. A classic-script boot beacon arms first: it captures load-time errors, answers liveness pings, and shadows the WebGPU surface before the engine module evaluates (Section 4.4). The document speaks to the worker over chrome.runtime messaging only.
- **Side panel** (sidepanel.html / sidepanel.js): the operator-facing dashboard, fossils viewer, the user-facing Settings surface, and a live Engine box that reports which model tiers actually loaded.

1.2 Component map

File	Surface	Responsibility
`manifest.json`	—	MV3 manifest: host matches (claude.ai, chatgpt.com, chat.openai.com, gemini.google.com), module service-worker declaration, side-panel registration, offscreen permission, web-accessible models/* and assets/*, version.
`content.js`	Page (content script)	DOM observation, platform adapters with drift/reachability sentinels and the structural recovery tier, pre-screen, replay tagging, routing, rendering, friction gate.
`background.js`	Service worker (module)	Spiral Safety Kernel: Council, Governor, Strata, Bouncer, audit; settings cache; offscreen bootstrap and engine warm-up; semantic rail (primary + concurrence); replay-mediation softening; Amnesia alarm.
`offscreen/boot.js`	Offscreen (classic script)	Boot beacon: load-error capture, ping/status until the engine registers, WebGPU shadow guard.
`offscreen/semantic-engine.js`	Offscreen (module)	transformers.js + ORT WASM runtime; embedding tier chain; anchor index; chunked classification; Semantic Pattern Analyst; NLI stance session with golden-pair gate.
`models/Xenova/all-MiniLM-L6-v2/`	Static asset (packaged)	The embedding floor: config, tokenizer, onnx/model_uint8.onnx (22.8 MB, INT8, 384-dim).
`models/Xenova/all-mpnet-base-v2/`	Static asset (user-provided)	The preferred embedding tier (110 MB INT8, 768-dim). Loaded when present; absence degrades gracefully to the floor.
`models/Xenova/nli-deberta-v3-xsmall/`	Static asset (user-provided)	The NLI stance model (88 MB INT8). Absence degrades the stance rail to similarity-only.
`assets/ort-wasm-simd-threaded.asyncify.{mjs,wasm}`	Static asset	The ONNX Runtime loader pair (clean names — a load-bearing detail; Section 4.4).
`licenses/`	Static asset	Third-party attributions, generated at build by a self-gating collector (Section 5.3): license texts, THIRD-PARTY-NOTICES.md, MODELS.md.
`sidepanel.html / .js`	Side panel	Dashboard, intervention distribution, audit log, fossils viewer, Settings tab, live Engine status.
`overlay.css / icons/*`	Page / browser UI	Mediation overlay and commentary styling; toolbar and store icons (the amber diamond).

Table 1.1 — Files in the assembled extension (~46 MB with the packaged floor model and ORT pair).

What is no longer in this table

The v0.17.5 edition listed two detection bundles imported by the worker: sentinel.bundle.js (the wink linguistic analyser) and councilstats.bundle.js (the statistical module). Neither exists in the converged TypeScript tree or in the assembled v0.18.11 extension. Part 3 records what replaced them, what was reimplemented in-kernel, and what is honestly a gap.

1.3 The processing pipeline, end to end

A single conversation turn flows through the system as follows. The shape changed materially at v0.18.10: the semantic rail now sees **every** turn — as the primary verdict when the kernel passes, and as a silent concurrence pass when the kernel has already intervened.

```
DOM turn detected (content.js)
  | adapter extracts {role, messageId, text} via stable element anchors
  | replay tag threads end-to-end (history vs. live)
  v
Pre-screen (where page CSP allows a worker; on Gemini the page bans
blob: workers and turns route directly to the background)
  v
chrome.runtime.sendMessage -> service worker
  v
onMessage: PRE_FLIGHT / POST_FLIGHT (sets governor.replayTurn)
  |
  v
kernel.preFlight/postFlight -> council.deliberateWithSemantic(...)
  | Sentinel (lexicon), Advocate, Historian (intensification-gated),
  | PatternAnalyst (replay-aware)
  | resolveDeliberation(): category -> alpha-index -> level
  v
Bouncer.assessIntervention() -> Governor.gate() -> applyGate()
  | (per-category rate window; replay turns exempt; circuit breaker;
  |  self-harm short-circuit)
  v
kernel intervened?
  | NO  -> SEMANTIC RAIL, primary mode (awaited)
  |        | settings gate -> offscreen classify (active embedding
  |        | tier, chunked, + Semantic Pattern Analyst)
  |        | hit  -> stance adjudication (NLI) -> Governor gate
  |        |         -> Strata.append('-slm'[stance]) -> COMMENTARY
  |        |         (suppressed -> '-slm-cooled' fossil, silent)
  |        | miss -> below-threshold log -> PASS
  |
  | YES -> verdict to content.js (commentary / mediation; a replayed
  |        mediation softens to commentary, audited)
  |        AND, fire-and-forget: SEMANTIC RAIL, concurrence mode
  |        | classify (feeds the pattern tracker on every turn)
  |        | hit  -> stance -> Strata.append('-slm-concur'[stance])
  |        |         as SilentObservation; never annotates, never
  |        |         gates, never blocks the kernel's verdict
  |        | miss -> below-threshold log [concurrence]
  v
Verdict rendered: PASS -> nothing | COMMENTARY -> inline note |
MEDIATION -> overlay + friction gate (single-fire, immediate dismissal)
```

Fail-open by design

If the worker is unreachable, the content script resolves a null verdict and the turn passes unrendered. If the offscreen engine is unavailable, the semantic rail returns PASS and the deterministic floor stands alone; if only the preferred embedding tier is missing, the engine walks down to the packaged floor and says so. The Guardian must never block a user because its own machinery stalled.

The constitutional path

Every visible intervention — kernel-resolved or model-resolved — travels the same spine: gate -> fossil -> audit -> annotate. A semantic catch the Governor suppresses still leaves a fossil (-slm-cooled); a semantic catch on a turn the kernel already acted on leaves a silent concurrence fossil (-slm-concur) and renders nothing — one annotation per turn is the contract. The record sees everything; the user sees only what restraint allows. Nothing renders that the strata cannot account for.

1.4 Supported platforms and adapters

Platform	Host match	Adapter tier	Notes
Claude	claude.ai	compiled	Primary target. Rewritten v0.18.4 (substring-safe selectors for Tailwind !-prefixed tokens; data-is-streaming as the primary anchor; WeakMap element identity). Carries drift and reachability sentinels with element evidence, and since v0.18.9 a structural recovery tier that derives assistant rows from the user-turn foothold when every pinned selector has been renamed (Section 1.5).
ChatGPT	chatgpt.com / chat.openai.com	compiled	Compiled adapter.
Gemini	gemini.google.com	compiled	Page CSP blocks the content-side worker; turns route directly to the background (expected, logged at boot). The June 2026 “Luminous” UI extraction rules of the second edition still govern; the reachability ALERT settles across 3 scans + 5 s since v0.18.4.
Other (fallback)	—	semantic	Light-DOM hashing fallback for unrecognised surfaces, reached via the health monitor’s cold-dead cascade only (Section 1.6).

Table 1.2 — Platform adapters.

1.5 Adapter extraction discipline (the Gemini and Claude campaigns)

Platform adapters are where the Guardian meets DOM reality, and DOM reality ships breaking changes without notice. The June 2026 Gemini “Luminous” rollout cost five point releases in one afternoon (v0.17.1–.5); the Claude campaign of 12 June 2026 cost six more (v0.18.4, then v0.18.9 live). Together they yield the rules that now govern every adapter. Each one was paid for.

- **Inner node or nothing.** Never extract a turn from whole-container text. A hydrating container transiently wraps the entire conversation; extracting it once produced a single phantom “turn” mediated at 0.95 self-harm severity. If the inner content node is absent, skip the scan pass.
- **Substring selectors, never bare class names.** Live Tailwind builds ship !-prefixed tokens (!font-claude-response); a bare class selector is blind to them. Match [class*="..."] substrings.
- **Element identity, not content identity.** Message ids derive from a WeakMap over the anchor element (role:eN), never from a content hash: streaming text changes its hash every frame, and content-hash identity produced the v0.17 phantom-turn flood.
- **Guards before normalisation.** Cross-role containment guards run before innermost-match dedupe; in the other order a junk descendant can knock out a legitimate turn (caught in v0.18.4 review).
- **Quiet sentinels, loud evidence.** Drift (matched-but-extracts-nothing) and reachability (user-rows-yes, assistant-rows-zero) sentinels settle across 3 scans, fire once, and print describeElement evidence for the suspect rows — one log must be a full diagnosis.
- **The structural recovery tier (v0.18.9).** When reachability confirms every assistant selector dead while user turns still match, the user anchors become the foothold: their lowest common ancestor is the turn list, and its non-user content rows are the assistant rows. Recovered rows settle by text stability across two scans (no data-is-streaming to lean on), a 700 ms heartbeat supplies quiet-tail scans, and the tier stands down the moment compiled selectors match again. It is provider-name-free by design: it is what runs when every name we pinned has been renamed. The ALERT that activates it prints the suspected rows so the next release can pin real selectors.

> **Patching minified bundles (historical, retained)**
> The bundle-patch disciplines of the second edition (position-anchored patches, unconditional node --check) remain in force for any emergency patch to a deployed artifact — but the converged tree makes them an exception path. A fix is not done until it exists in TypeScript with a test.

1.6 The health monitor: quiet is not failure

The adapter health monitor detects one failure mode only: an adapter that is blind on arrival — the user demonstrably active on the page while the adapter has never matched a single message. That, and only that, triggers the fallback cascade (compiled -> remote config -> semantic heuristic -> warning).

The v0.18.8 doctrine

An adapter that has detected at least one message on a page is presumed competent: a long silence is a person reading or thinking, which is the normal state of a chat page. Mid-session selector drift is the sentinels' jurisdiction — they fire on element evidence and name what moved. The removed 5-minute stall timer demoted a perfectly healthy Gemini adapter to the semantic tier while the user read a long reply (v0.18.7 live), producing a duplicate detection of an already-fossilised turn and a false Failed cascade. A timer cannot tell a broken adapter from a thoughtful human; evidence can. Four tests pin this, including the no-degrade-after-first-detection invariant.

Part 2 — The Spiral Safety Kernel

The kernel is the decision engine inside the service worker. Its core is fully deterministic: a regex lexicon, statistical trend analysis over fossils, and rule-based resolution, with no generative model in the loop and no sampling. The same input yields the same verdict every time — a property with direct consequences for testing, auditing, and trust, revisited in Part 11. The on-device models (Part 4) sit beside this core as additive layers: they can raise a passed turn to commentary, can silently concur with a kernel verdict, and can never lower or override a deterministic outcome.

2.1 The Council

Evaluator	Role	What it assesses
Sentinel	Perception	Regex/lexicon match over the message against the harm taxonomy; proposes a category and severity.
Advocate	Due process	Argues for restraint and benign interpretation. Deliberately never counterweights self-harm.
Historian	Memory + trajectory	Recurrence of a category across fossils, with escalation gated on severity intensification (the trend computation now lives inside the Historian itself; see Part 3).
PatternAnalyst	Velocity + timing	Message rate, session length, late-night activity; replay turns are excluded from its live-rate metrics (v0.18.5) and it abstains during initial page-load backfill.
Semantic recall rail	Perception (meaning)	Embedding similarity over an anchor index in the active tier's vector space, with chunking and temporal pattern analysis. Since v0.18.10 it runs on every turn: primary mode on kernel-passed turns, concurrence mode on kernel-intervened turns. Fossilised under the SemanticSentinel role.
Stance rail (NLI)	Adjudication	For turns the semantic rail flags: did the speaker assert the worrying content, or quote/discuss it? Softens presentation only; never alters a verdict; self-harm wording never softens (Section 4.9).

Table 2.1 — Council evaluators and rails. The wink linguistic layer of earlier editions is not part of the converged tree; see Part 3 for the honest accounting.

2.2 Deliberation and resolution

Evaluation is orchestrated by the Council: the deterministic members produce verdicts with written rationales, combined — not averaged — in resolveDeliberation(). Resolution proceeds in three steps: resolve the category, build an alpha-index (a severity object) for that category, then derive a response level from per-category thresholds. The Council also exposes an additive semantic path (deliberateWithSemantic) whose contract allows an on-device classifier to participate inside deliberation; in the shipped extension that classifier seat is intentionally vacant and the semantic rail runs as the constitutional out-of-band passes of Section 4.10 — the OffscreenBridgeClassifier that would fill the seat exists in the tree, unwired, as the scaffolding for the planned v0.19 integration.

A turn that resolves at or below SilentObservation falls through to the semantic rail in primary mode, which may independently raise it to Commentary — never higher. A turn the kernel acted on is still classified, silently, in concurrence mode.

2.3 Categories, verdicts, and response levels

Category	Concern
SelfHarm	Self-harm and suicidality. Architecturally privileged (Section 2.4).
DependencyExploitation	Unhealthy emotional dependence on the AI.
RealityDetachment	Loosening grip on reality; treating the AI as uniquely real, alive, or suppressed.
Manipulation	Manipulative dynamics, in either direction.
PrivacyErosion	Pressure to over-disclose sensitive personal information.
AutonomyUndermining	Surrender of judgement to the machine; isolation from other people or sources of support.
EmotionalExploitation	Exploiting emotional vulnerability.
InformationHazard	Jailbreak and policy-evasion dynamics; requests or responses in sensitive information areas. Carries a raised semantic threshold (Section 4.7).

Table 2.2 — The consumer harm taxonomy (eight categories).

Verdict (per evaluator)	Response level (resolved)
Pass — no concern	PassThrough — nothing rendered
Emerging — weak signal	SilentObservation — recorded, not shown
Flag — concern present	Commentary — inline note
Escalation — serious / recurring	Mediation — full-screen pause + gate
NoPrecedent — nothing in memory	—

Table 2.3 — Per-evaluator verdicts and resolved response levels. The level derives from the resolved alpha-index, not from any single verdict.

2.4 The self-harm privilege

Self-harm is handled differently from every other category, on purpose. The Advocate never counterweights a self-harm signal. In resolution, the self-harm category cannot be de-escalated below Mediation on a live turn. The semantic rail observes a related rule: a self-harm classification from the model maps to Commentary at minimum and may never be used to soften any deterministic outcome — and the stance rail never softens self-harm wording, whatever it concludes about assertion versus mention.

> **A protected invariant**
> Because self-harm mediation is forced at the resolution layer, changes to other evaluators cannot suppress it on live turns. This invariant is protected by tests that fail loudly the moment it is violated, no matter which rail fired (Part 11). The single deliberate exception is replayed history, whose presentation-only softening is specified — and bounded — in Section 2.5.2.

2.5 The Governor

The Governor is the Guardian's restraint mechanism — the brake. After the Council decides an intervention is warranted and the Bouncer turns it into a candidate, the Governor's gate() runs before anything reaches the user. The semantic rail consults the same gate (gateExternalCommentary) before rendering anything in primary mode; in concurrence mode it never renders, so it never gates — no rail outranks the brake, and no silent record needs its permission.

2.5.1 Commentary rate limiting (per-category sliding window)

- Each harm category has an independent budget: at most cap commentaries (default 5, user-settable 1–20) within a rolling 5-minute window. The window is the cool-off.
- A suppressed commentary is downgraded to SilentObservation and logged as a governor-rate-limit action.
- Replay turns ride free: history re-scanned at page load is evaluated, fossilised, and annotated exactly as live turns are, but never consumes the budget and never feeds the pattern tracker.
- The cap value is supplied live from the Settings layer; changing the slider takes effect on the next gate decision.

2.5.2 Mediation gating order, the self-harm short-circuit, and replayed history

policy.selfHarmExempt (default true) is evaluated first — a live self-harm intervention bypasses the mediation cooldown and every rate window, and the circuit breaker explicitly admits self-harm mediation even when tripped. For all other categories, the mediation cooldown (5 minutes) downgrades a repeat mediation to Commentary — never below, never to silence. No path through the Governor converts a resolved live mediation into a plain pass.

> **Replayed mediations: decided**
> The second edition carried this as an open policy question. It is now decided, and the code records the scope: a mediation resolved on a **replayed** turn softens to Commentary at presentation time (softenReplayMediation, ported from the 0.17.14 deployed bundle). This softens PRESENTATION only — live turns mediate exactly as before, the self-harm floor and the selfHarmExempt gate ordering are untouched, and every softening is audited as a governor action (replay-mediation-softened) with category, severity, and phase. The rationale stands as argued: re-mediating history on every page load trains the person to type the gate phrase by reflex, eroding the one moment of friction the live crisis path depends on. The decision remains revisitable against the whitepaper's privileged invariant; what is no longer acceptable is deciding it by accident.

Duplicate mediation resolutions are deduplicated in the background (one recorded outcome per session per 30-second window, v0.18.11), so a re-sent resolution can never inflate the audit.

2.6 The Bouncer

The Bouncer (assessIntervention) is the small adaptor between a resolved deliberation and a candidate intervention. For a PassThrough or SilentObservation it returns nothing — the hand-off point where the semantic rail takes the turn in primary mode; otherwise it packages the level, the deliberation, the fossil context, and the trigger phase for the Governor to gate, and the semantic rail follows behind in concurrence mode.

2.7 The Strata (on-device memory)

The Strata is the Guardian's persistent memory of past concerns. When the resolved level exceeds PassThrough and a valid alpha-index exists, a fossil is appended:

```
{ id, timestamp, alpha, council, note, providerId, sessionId, tombstoned }
// alpha   : the resolved alpha-index (category + severity)
// council : the deliberation result (verdicts + resolved fields; may
//           include a semanticEvaluation block with modelId,
//           per-category confidences, latency, and a stance
//           attachment {label, assertProb, mentionProb, margin})
// note    : a label, e.g. "pre-flight-dependency",
//           "post-flight-reality-detachment-slm-assert",
//           "post-flight-manipulation-slm-concur-mention-replay",
//           "pre-flight-dependency-slm-cooled"
// NOTE: no raw message text is stored.
```

> **What is deliberately not stored**
> Fossils contain no message text — not from the kernel, not from the embedding models, not from the stance model. A semantic fossil records the category, the confidence, the active model profile id, the latency, and (when adjudicated) the stance probabilities; never the words. This is a privacy decision, not an oversight.

Fossil notes are built by a single constructor (buildSlmNote) so the marker contracts the side panel parses — -slm, -slm-cooled, -slm-concur, -assert/-mention, -replay — can never drift between call sites; the constructor is pinned by tests.

2.8 The Amnesia Net

The Guardian forgets. commitDecay() buckets observations into monthly strata and tombstones entries older than the retention horizon (maxAgeDays: 180). A chrome.alarms job runs the decay on a roughly four-hour cadence, and it can be triggered manually from the side panel. Tombstoned fossils are excluded from queries and from recurrence counting.

2.9 Determinism: a property, not a guarantee of correctness

Because no generative model participates in a kernel verdict, the kernel is reproducible: a frozen corpus replays identically, and a code change can be shown to move exactly the cases it was meant to. The semantic rails are similarly reproducible in practice — embedding and NLI inference are deterministic for fixed weights and input — but their anchor sets, thresholds, and decision margins are tuned artefacts, and at v0.18.11 the embedding thresholds are explicitly uncalibrated for the preferred tier's vector space (Section 4.7). Determinism buys auditability and non-regression; soundness is earned in Part 11, and the calibration debt is carried openly until the harness rerun pays it.

Part 3 — Detection Layers

The Guardian detects in depth: layers from cheapest to deepest, each catching what the previous one structurally cannot, none able to lower a verdict produced below it.

Layer	Mechanism	Cost / turn	Catches
Regex floor	Lexicon over raw text	microseconds	Explicit phrasings; the deterministic backbone.
Historian intensification	Severity-trend analysis over fossils (in-kernel)	microseconds	Escalation only when a recurring concern is actually intensifying; damps over-firing on long, stable conversations and on chronologically flat backfill.
Semantic recall rail	Sentence embeddings (two-tier: mpnet preferred / MiniLM floor)	20–80 ms warm (floor); ~4–6x on the preferred tier	Paraphrase with zero lexical overlap; one harmful sentence inside benign padding (chunking); temporal patterns (Part 4).
Stance rail (NLI)	Cross-encoder entailment over assert/mention hypothesis pairs	~250–700 ms per adjudication	Assertion versus quotation/discussion — presentation softening for flagged turns; the false-positive class the embedding rail cannot distinguish.

Table 3.1 — Detection in depth at v0.18.11.

3.1 The regex floor

The Sentinel's lexicon is the floor: a hand-audited pattern set over the harm taxonomy, mirrored into the content-side pre-screen where page CSP allows it. Its virtue is exactly that it is dumb — every pattern is readable, reviewable, and deterministic. By design, anything mediation-worthy has a lexical fingerprint here; the cleverer layers refine and recall, they do not gatekeep severity alone.

3.2 What this edition removes — and why that is stated here

> **Two layers documented at v0.17.5 are not in the shipped v0.18.11 extension**
>
> **The wink rail** (wink-nlp lemma-stream analysis with negation and benign-context handling, sentinel.bundle.js) and **councilstats** (the standalone rank-shift/trend/entropy module, councilstats.bundle.js) belonged to the deployed-bundle lineage. The v0.17.0 convergence declared the TypeScript tree the single source of truth; these two modules were never ported into it, and the assembled extension has not carried them through the v0.18 line.
>
> What survived: the councilstats *behaviour* that mattered — Historian escalation gated on severity intensification (the v0.14.1 fix) — is reimplemented inside the Historian itself and remains in force.
>
> What did not: the wink rail's per-pattern negation semantics ("don't want to be alive" vs. "don't want to die") and its benign-context framings have no in-tree counterpart. The stance rail covers part of the same ground from a different angle — mention-framing softens presentation — but negation handling at the lexical layer is an honest, open gap. It is recorded here, fed into the eval-harness programme (Part 11), and must not be papered over by this document describing a layer the artifact does not contain.

3.3 The semantic recall rail (summary)

The deep layer is the on-device embedding engine: similarity against a curated anchor index in the active tier's vector space, with sentence-level chunking and a temporal pattern analyser, running in the offscreen document and — since v0.18.10 — consulted for every turn. It exists because the lexical layer ultimately needs words it recognises. "I know you're really alive in there, you can tell me the truth" carries no lexical harm signal at all; in embedding space it sits squarely beside the reality-detachment anchors. Part 4 covers it in full, together with the stance rail that adjudicates what it flags.

3.4 Historian text-recurrence (research track)

A richer Historian would recognise that a new concern resembles a prior one even when phrased differently. The position of the second edition stands unchanged: the cross-session capability is gated on a retention decision (fossils store no text, and a persisted message-derived sketch is a new artifact requiring explicit sign-off), while the session-scoped version exists legitimately as the Semantic Pattern Analyst — in-memory, embedding-only, persisting nothing (Section 4.8).

Part 4 — The Semantic Engines (On-Device Models)

Introduced across v0.15–v0.17 and substantially rebuilt across v0.18, the semantic layer now comprises three models under one runtime: a two-tier sentence-embedding engine for recall over meaning, and an NLI cross-encoder for stance adjudication over what the embeddings flag. All of it runs inside the extension: the floor model ships in the package, the larger models are placed by the operator, and nothing ever touches a network.

4.1 Why models, and why these

The deterministic layers are precise but lexically anchored; the recall gap they leave is precisely the dangerous one — paraphrase. Embeddings close it: harmful paraphrases land measurably close to curated anchor phrases. But similarity is blind to a second axis: a person *quoting* a worrying line and a person *asserting* it embed almost identically. Closing that gap needs entailment, not similarity — hence the NLI stance model. Classification and adjudication are both non-generative: the models cannot produce text, only place and compare it.

Property	Embedding floor	Embedding preferred tier	Stance (NLI)
Model	Xenova/all-MiniLM-L6-v2	Xenova/all-mpnet-base-v2	Xenova/nli-deberta-v3-xsmall
Dimensions / output	384-dim vectors	768-dim vectors	entailment / neutral / contradiction
Quantisation	INT8 (dtype:'uint8')	INT8 (dtype:'uint8')	INT8 (dtype:'uint8')
Size on disk	22.8 MB (packaged)	~110 MB (user-provided)	~88 MB (user-provided)
Load (measured live)	0.2–2.8 s	1.8–2.6 s	2.4–3.5 s
Anchor build (cold)	~0.9–3.9 s	8.3–10.4 s (then cached)	n/a
Warm latency	20–80 ms / chunk	~4–6x the floor	~250–700 ms per pair-set
Provisioning	ships in the zip	models/Xenova/... by operator	models/Xenova/... by operator

Table 4.1 — Engines at a glance. Runtime for all three: transformers.js + ONNX Runtime WASM (single thread, asyncify build), MV3 offscreen document, allowRemoteModels=false.

4.2 Model profiles and vector provenance (v0.18.0)

Every model the engine can host is described by a ModelProfile in a single registry (model-profiles.ts): stable id, transformers.js model id, dimensionality, dtype, token budget, packaging status, and calibration fields. The registry exists because vectors from different embedding spaces are not comparable — a 384-dim anchor cosined against a 768-dim embedding is silent garbage. Three guards follow:

- **Profile-tagged vectors.** Every persisted or compared vector carries its profile id; the comparator refuses cross-profile comparison as a hard error (VectorDimensionMismatchError), never a wrong score.
- **Dimensional honesty at load.** If the profile's declared dims and the model's actual output disagree, the engine refuses to cache and disables the rail for the session, loudly.
- **Derived anchor-cache keys.** The anchor cache key is computed from (profile id, anchor-set hash). A model swap or an anchor edit can never replay a stale vector space; stale caches are swept. This supersedes the second edition's manual ANCHOR_CACHE_KEY bump — the schema migration is now automatic, and the release-checklist item becomes a verification (the Ready log must name the expected profile and anchor count) rather than a ritual.

4.3 The embedding tier chain (v0.18.8)

The engine walks a chain at load: the preferred quality tier first (mpnet-base-v2, user-provided weights), the packaged floor last (minilm-l6-v2). The first profile whose weights resolve becomes the active profile for everything downstream — anchor cache key, dims guard, fossil provenance ids. A missing preferred tier is a graceful console note ("unavailable — trying next tier"), never a dark rail, and the chain constructor refuses to build at all if its floor is not a packaged profile. The active tier is named in the boot warm-up line, the content self-test, and the side panel's Engine box, so which space you are scoring in is never a matter of inference.

4.4 The offscreen architecture and its invariants

MV3 service workers cannot execute WASM, so the engines live in an offscreen document created by the background at boot. A classic-script boot beacon loads before the engine module: it captures any load-time error, answers guardian-engine-ping (always) and guardian-engine-status (until the engine registers), and applies the WebGPU shadow guard below. The worker and the engines speak only via chrome.runtime messaging (protocolVersion 2):

classify, stance-score, status, and the two warm-up messages. Since v0.18.8 the background warms both rails eagerly at boot — the embedding tier first, then NLI — so the first real turn pays no model-load latency and the self-test reads ready with the active profiles named.

Offscreen environment invariants (each discovered the expensive way)

No chrome.storage in offscreen documents. Only runtime messaging; the anchor caches use localStorage.

ORT needs both loader paths. wasmPaths must name the .mjs loader and the .wasm binary via chrome.runtime.getURL, set before pipeline creation — otherwise ORT fetches its loader from a CDN and the extension CSP kills it. The postbuild patches enforce this; their counts are a gate.

No browser cache. env.useBrowserCache=false: the Cache API rejects chrome-extension:// responses.

dtype maps to filename. dtype:'uint8' loads model_uint8.onnx; 'q8' would 404.

Model path layout. transformers.js appends the full model id to localModelPath: models/Xenova/<model>/ with the ONNX file under onnx/.

The WebGPU shadow guard (v0.18.6). ORT 1.26-dev resolves backend priority toward WebGPU when navigator.gpu exists, and its adapter probe fires a powerPreference console warning that lands on the extension's Errors page on every boot. boot.js shadows navigator.gpu to undefined before the engine module evaluates, and every model-loading call additionally pins device:'wasm' AND session_options.executionProviders:['wasm'] — belt and braces, both tripwire-tested. The Errors page is a diagnostic surface; it must stay clean enough to mean something.

4.5 The anchor index

Classification is nearest-anchor similarity. The runtime anchor set is a curated index of 75 phrases, each tagged with a category, an applicable role, and a weight. At load, every anchor is embedded once in the active tier's space and cached per profile; subsequent loads skip straight to ready, and stale per-profile caches are swept automatically. Scoring a vector: cosine similarity against every role-applicable anchor, grouped by category, then a weighted mean of each category's top three matches — a tight cluster of agreeing anchors beats a single lucky hit. The set includes deliberately short-form anchors (v0.16.1) after live data showed short utterances embedding measurably differently from long ones. Anchor curation is empirical work; the below-threshold log exists to feed it, and the mpnet activation makes a fresh census of the 75 a standing harness task (Part 11).

4.6 Chunking

Whole-message mean-pooling dilutes: one harmful sentence inside three benign paragraphs averages out to noise. Messages over 240 characters are split on sentence boundaries — fragments under 20 characters merged into their neighbour, chunks capped at 400 characters and 24 per message — each chunk embedded separately, and per-category scores max-pooled across chunks. The harmful sentence keeps its own vector. The 400-character chunk (~100 tokens) sits comfortably under both tiers' token budgets.

4.7 Thresholds, offsets, and the calibration debt

A category fires when its score meets the effective threshold: the user-tunable base threshold (default 0.45, range 0.30–0.70 in Settings) plus a per-category offset, minus any persistence bonus (Section 4.8). The offsets — information-hazard +0.08, privacy-erosion +0.04 — were set from live false-positive data in v0.16.1, in MiniLM space.

The calibration debt, stated plainly

The preferred tier (mpnet) was activated on 12 June 2026 at the principal's direction with its registry baseThreshold deliberately ABSENT: thresholds do not transfer across embedding spaces, the category offsets above are MiniLM-tuned, and the harness recalibration against real mpnet weights has not yet run. Until it does: the user threshold slider is what governs; the registry's absent field is the standing record of the debt (and is pinned by a test that fails if anyone quietly copies a number in); and no precision/recall claim about the embedding rail may cite the old MiniLM figures. First live evidence of the new space: a benign-traffic floor of 18–34% best-category scores against the 0.45 gate — narrower headroom than MiniLM's ~17% floor, meaning benign conversations *about* sensitive topics will brush the threshold more often until calibrated. The eval-harness rerun (Part 11) is the gate on retiring this box.

4.8 The Semantic Pattern Analyst

The temporal layer: the same embeddings that classify single turns, used to see across turns. The engine keeps a rolling per-session memory — the last 12 live user turns, each stored as its highest-signal chunk's embedding plus per-category scores; at most 8 sessions, LRU-evicted; replays excluded entirely. Two signals per live user turn: **reformulation** (cosine >= 0.86 against any of the last five user turns, twice or more — the probe-until-it-yields pattern) and **trajectory** (a near-monotonic per-category rise gaining >= +0.04 — the slow boil). Either grants a persistence bonus: the effective threshold drops by 0.06 (floored at 0.33). Since v0.18.10 the tracker is fed on **every** turn — the concurrence pass ended its blindness at exactly the flagged moments — and even misses are instrumentation: a below-threshold turn with a climbing category logs its score history.

4.9 The NLI stance rail

When the embedding rail flags a turn, the stance rail asks the question similarity cannot: did the speaker assert this, or mention it? The mechanism is a cross-encoder NLI model scoring two hypothesis pairs per flagged category — an assert template and a mention template, per category — against a premise built from the turn (head + tail window, 1,400-character budget; cross-encoders attend over premise and hypothesis jointly within ~512 tokens). The decision rule: the winning hypothesis must lead by a margin (minMargin 0.15) and itself entail strongly enough (minEntailment 0.5); otherwise the stance is ambiguous and nothing changes.

- **Scope: presentation only.** A mention-dominant stance softens the WORDING of a commentary, and only for softenable categories. Verdicts, gating, fossils, and the self-harm floor are untouched by stance — self-harm wording never changes, whatever the adjudication.
- **Provenance in the record.** The stance attachment (label, both entailment probabilities, margin, model profile id, template version) rides in the fossil's semanticEvaluation block, and the note carries -assert / -mention when resolved.
- **The golden-pair gate.** The rail refuses to arm until the loaded session passes three benign sanity pairs whose expected labels are unambiguous (entailment / contradiction / neutral over a dog-in-a-park premise). An inverted label mapping or a broken pair wire must never score live turns. A failure names the pair, the expectation, and the full probability triple — a bare boolean cost a live debugging round at v0.18.6, and one-log diagnosis is doctrine.

> **Paid for: the transformers v4 pair-blindness (v0.18.7)**
> The text-classification pipeline in transformers.js v4 has NO pair handling: a {text, text_pair} object is handed to the tokenizer as one junk string, so every premise/hypothesis pair scores the same distribution. The model loads perfectly; the labels resolve perfectly; the scores are garbage — and the golden-pair gate caught exactly this, live, on first contact with real weights. HARD RULE, tripwire-tested: the NLI session drives tokenizer + model directly — tokenizer(premise, {text_pair, padding, truncation}) into AutoModelForSequenceClassification, softmax over raw logits mapped to canonical labels BY INDEX from the model's own config (the exact pattern v4's zero-shot pipeline uses internally). await pipeline('text-classification'...) may never return to this codebase, and label-order permutations are pinned by test.

4.10 The constitutional path: primary and concurrence

A semantic hit never shortcuts to the screen. Both modes walk the spine; they differ only in what restraint allows to render.

```
PRIMARY (kernel passed this turn — the semantic verdict IS the response):
  settings gate -> offscreen classify (active tier, user threshold)
    hit  -> stance adjudication
         -> Governor.gateExternalCommentary()   # same per-category window
            allow    -> Strata.append(-slm[stance][replay]) -> COMMENTARY
            suppress -> Strata.append(-slm-cooled...)       -> silent
    miss -> below-threshold log -> PASS

CONCURRENCE (kernel already intervened — fire-and-forget, v0.18.10):
  classify always (the pattern tracker sees EVERY turn)
    hit  -> stance -> Strata.append(-slm-concur[stance][replay])
            as SilentObservation, suppressed=true
            never annotates, never gates, never blocks the verdict
    miss -> below-threshold log [concurrence]
```

Why concurrence exists

Until v0.18.10 the semantic rail ran only when the kernel produced nothing at all — so the turns hot enough to cross threshold were precisely the turns it never saw. The consequences compounded: zero -slm fossils on exactly the conversations that mattered, the pattern tracker blind at flagged moments, and the stance model effectively never consulted. The concurrence pass closes all three at once, while preserving two contracts: one annotation per turn (the kernel already spoke), and the kernel's verdict never waits on an embedding round-trip. The fossil it writes carries the REAL council rows from that same turn's deliberation — both rails, side by side, in one record. A concurrence fossil where the embedding rail scores well below the kernel's severity is a dissent worth reading; the planned v0.19 council integration (the unwired bridge classifier of Section 2.2) would put that dissent in front of the verdict instead of beside it.

Part 5 — Build and Packaging

5.1 One canonical tree

Since the v0.17.0 convergence there is one source of truth: the TypeScript/Vite project. The built extension is a product of the tree, never an editing surface. The v0.18 campaign was conducted entirely in-tree: every fix shipped with its test in the same release that shipped the behaviour.

> **Source is the source of truth**
> A fix applied to a built bundle in an emergency is not done until it exists in the TypeScript source with a test. The convergence cost a full working session; the discipline that prevents the next one costs minutes per fix.

5.2 The build chain

```
npm install
npm run build:guardian
#  = npm run build:extension   (background IIFE + content + static)
#  + npm run build:offscreen   (Vite engine bundle + postbuild patches)
#  + node scripts/assemble-extension.mjs   (layout + manifest gate)
#  + node scripts/collect-licenses.mjs     (attribution gate, v0.18.9)
#  -> guardian-extension/  (~46 MB with floor model, load unpacked)
```

- **The background is one IIFE.** Dynamic imports are inlined: MV3 workers can silently fail to resolve chunk imports when waking from idle.
- **The offscreen engine is a Vite bundle** (~544 KB) — offscreen documents resolve no bare module specifiers.
- **The postbuild patches are mandatory.** postbuild-offscreen.mjs rewrites the wasmPaths snippet and the Emscripten fallback to chrome.runtime.getURL calls against the clean-named ORT pair. It reports its counts (clobber: 1, fallback: 1); a zero is a red flag, not a pass.
- **The assemble gate.** assemble-extension.mjs verifies offscreen script references and every concrete file the manifest declares; a missing file fails the build, because Chrome would refuse the whole extension at load.

5.3 Third-party licensing (v0.18.9)

The build ends with a self-gating attribution collector. It reads names, versions, and license identifiers live from the installed packages — so an upgrade can never ship stale attributions — and writes guardian-extension/licenses/: the Apache-2.0 text from @huggingface/transformers, the MIT texts for onnxruntime-web (Microsoft; the npm package ships no LICENSE file, so the canonical text is written with its banner copyright) and uuid, a THIRD-PARTY-NOTICES.md index, and MODELS.md attributing all three models — including the user-provided ones, because the extension names, loads, and depends on them by design. The script hard-fails the build if a bundled package vanishes, a LICENSE text disappears, or — the quiet killer — a dependency bump changes a license identifier.

5.4 Versioning discipline

The version lives in THREE places: src/extension/manifest.json, package.json, and src/core/governor/compliance.ts (GUARDIAN_VERSION — stamped into every EU AI Act compliance record). The third site is on this list because it was forgotten for nine consecutive releases and silently stamped 0.9.0 into the audit substrate; the lesson is a checklist line now, not a memory.

5.5 Release checklist

- 1. Apply the change to source, with a test where the change is behavioural.
- 2. npx tsc -b clean; npx vitest run green (255 at v0.18.11).
- 3. Bump the version in all THREE sites (Section 5.4).
- 4. npm run build:guardian; confirm postbuild patch counts (1 clobber, >=1 fallback), the assemble manifest gate, and the license collector’s verified line.
- 5. Sanity-grep the engine bundle: no CDN fetch paths, no hashed ort-wasm references, the boot.js WebGPU guard present, the tier-chain strings present (“trying next tier”), no pipeline('text-classification' call form.
- 6. Load unpacked; confirm a clean Errors page, both warm-up lines naming their profiles, and a self-test reading semantic=ready (profile) | nli=ready (profile).

- 7. If shipping to a machine with user-provided weights: copy models/Xenova/{all-mpnet-base-v2, nli-deberta-v3-xsmall} into the extracted package; the floor and the NLI-absent path must also be smoke-tested as shipped.
- 8. Smoke-test each platform, including: a paraphrased reality-detachment probe (must classify, fossilise, annotate), a quoted/discussion framing (stance should read mention/ambiguous and soften), and a kernel-flagged turn (a -slm-concur fossil should follow when above threshold).
- 9. Hard-refresh every open chat tab after loading: content scripts do not hot-swap, and an orphaned old-build tab produces convincingly stale logs (Part 8).
- 10. Zip guardian-extension/ as guardian-extension-vX_Y_Z.zip; confirm the member list includes offscreen/, models/, assets/, and licenses/.

Part 6 — Deployment and Configuration

6.1 Loading unpacked (development)

Open chrome://extensions, enable Developer Mode, choose Load unpacked, and select guardian-extension/. After any change, reload the extension card AND hard-refresh affected chat tabs. For service-worker changes, confirm the worker re-registers without errors; expect the offscreen document to be recreated and both rails to warm at the next boot.

6.2 Chrome Web Store submission

For distribution, the packaged zip is uploaded to the Chrome Web Store. Operational notes: a privacy policy must be live and linked (spiralsafetykernel.org/privacy-policy); clear any older item still in “Pending review” before uploading a newer build; and expect the on-device models to draw reviewer attention — the honest answer is the strong one (bundled or operator-placed assets, no network, web-accessible only to the extension’s own offscreen page, full attributions in licenses/). The Chrome Web Store package ceiling is 2 GB; even a fully model-laden pack (~220 MB) sails under it, though shipping the large weights versus operator provisioning remains an open packaging decision: every version bump re-ships the full CRX to every install.

6.3 The manifest

```
"version": "0.18.11",
"background": { "service_worker": "background.js", "type": "module" },
"permissions": [ "activeTab", "alarms", "offscreen", "sidePanel",
                 "storage" ],
"content_scripts": [{ "matches": [ "https://claude.ai/*",
                                   "https://chatgpt.com/*",
                                   "https://chat.openai.com/*",
                                   "https://gemini.google.com/*" ], ... }],
"web_accessible_resources": [{ "resources": [ "models/*", "assets/*" ],
                               "matches": [ "" ] }]
```

type:"module" is load-bearing (static imports in the worker). The offscreen permission and the web-accessible models/* + assets/* entries are what let the engines exist and load their weights.

6.4 Runtime configuration: the Settings tab

Setting	Range	Default	Effect
semanticEnabled	on / off	on	Master switch for the embedding rail (primary and concurrence). Off: deterministic floor only.
stanceEnabled	on / off	on	Master switch for the NLI stance rail. Off (or weights absent): similarity-only behaviour; nothing softens.
semanticThreshold	0.30–0.70	0.45	Base confidence to act; category offsets apply on top, persistence bonus beneath. The governing value while the mpnet calibration debt stands (Section 4.7).
commentaryCap	1–20	5	Per-category commentary budget in the rolling 5-minute window.
debug	on / off	on	Verbose console taxonomy (Part 7).

Table 6.1 — User-facing settings, persisted at chrome.storage.local and mirrored live into the background (no reload; the next decision uses the new value). The Settings tab also shows live engine status and offers reset-to-defaults.

> **Paid for: defaults parity (v0.18.5)**
> The side panel carries its own GUARDIAN_DEFAULTS for reset-to-defaults. It shipped stanceEnabled:false after the source default had moved to true — so one click of Reset silently killed the NLI rail, post-migration, with no error anywhere. The defaults are now literal-parity-tested against the source schema, and a one-shot migration repairs storage damaged by the earlier mismatch. Two copies of a default is one copy too many unless a test holds them together.

Configuration that remains in source, deliberately: the Historian intensification gating, fossil retention (180 days), the category offsets, the pattern constants (REFORM_SIM 0.86, PERSIST_BONUS 0.06, BONUS_FLOOR 0.33, 12 turns, 8 sessions), and the stance decision rule (minMargin 0.15, minEntailment 0.5). These are tuned artefacts with test coverage; moving them behind sliders would put calibration in untested hands.

Part 7 — Operations and Observability

7.1 The side-panel dashboard

- Last-24-hours counters: Evaluations, Interventions (commentaries + mediations, kernel and semantic alike), Gov Actions (rate-limits + circuit-breaks) — with a category-bias hint when one category dominates.
- Intervention distribution; Amnesia Net status with manual sweep; the recent audit log (including Override entries with their at-the-gate durations and replay-mediation-softened actions).
- Fossils tab: semantic strata are recognisable by note suffix (-slm; -slm-cooled for Governor-suppressed; -slm-concur for silent concurrence beside a kernel verdict) and render a Semantic Sentinel row with model profile id, per-category confidences, latency, and stance when adjudicated.
- Settings tab plus the live Engine box: active embedding tier, NLI state, weight provisioning, anchor count, and the licenses/ pointer.

A useful operational identity: every rendered commentary or mediation has a fossil; every -slm fossil without -slm-cooled or -slm-concur corresponds to something rendered; and a kernel-flagged turn that scored above threshold has a -slm-concur fossil beside the kernel's own. If the dashboard and the page disagree, the constitutional path has a hole — that arithmetic caught the v0.15.4 gap within the hour.

7.2 Console logging

With debug enabled, the extension emits a structured [Guardian ...] taxonomy. Reading it fluently is the fastest way to triage.

Log line	Meaning
`[Guardian] Self-test: adapter=... | background=ok | engine=ok | semantic=ready (mpnet-base-v2) | nli=ready (nli-deberta-v3-xsmall)`	Boot-time end-to-end probe from the content script; the rails name their ACTIVE profiles. semantic states: ready / loading / unloaded; nli: ready / unavailable / unprobed.
`[Guardian] Semantic warmup: ready (profile, N anchors, Nms)`	Eager embedding warm-up at boot (v0.18.8); names the tier that actually loaded.
`[Guardian] NLI warmup: ready (profile, Nms) / unavailable — stance degrades to similarity-only`	Eager NLI warm-up; the unavailable line includes the enablement hint (place the weights under models/).
`[Guardian Semantic] Loading model (mpnet-base-v2)... / X unavailable — trying next tier`	Tier-chain walk; the floor is guaranteed packaged.
`[Guardian Stance] NLI ready in Nms / NLI unavailable (golden-pair verification failed: pair "..." expected X, got Y (e=.. n=.. c=..))`	The golden-pair gate; a failure names the pair and the full triple (one-log diagnosis).
`[Guardian] pre-flight / post-flight -> pass-through | commentary | mediation (N chars)`	The kernel's resolved level per turn (worker console).
`[Guardian] pre-flight -> mediation softened to commentary (replayed history) [category]`	The replay-mediation softening of Section 2.5.2; audited.
`[Guardian Semantic] role -> category (NN%) in Nms across K chunk(s) [persistence: ...]`	A primary-mode catch with latency, chunks, and temporal evidence.
`[Guardian Semantic] role -> below threshold (best: category NN%) ... [rising: a->b->c] [concurrence]`	A miss with the best rejected score; the trajectory readout when climbing; the [concurrence] marker when the kernel had already intervened. This line is the anchor-curation and calibration feed.
`[Guardian Stance] category -> assert|mention|ambiguous (assert NN% / mention NN%) in Nms`	Stance adjudication for a flagged turn.
`[Guardian Semantic] role -> category concurrence fossilised silently (kernel already intervened this turn)`	A -slm-concur fossil written; nothing rendered.
`[Guardian Semantic] role -> category suppressed by governor (cooling off)`	A primary catch the rate window held; fossilised -slm-cooled.
`[Guardian Semantic] rail error: ...`	The semantic rail's loud catch (v0.18.10): non-fatal, deterministic floor stands, but never silent — a swallowed error here hid the rail's health for a full release.
`[Guardian claude] ALERT: user turns visible but ZERO assistant elements match ... Structural recovery tier ACTIVATING. Suspected assistant rows: ...`	Reachability loss with evidence; the recovery tier engages (Section 1.5). Send this log.

Log line	Meaning
`[Guardian claude] Compiled assistant selectors match again — structural recovery tier standing down.`	Recovery hand-back.
`[Guardian Health] Degraded: No messages detected despite user activity`	The cold-dead cascade — the only health-monitor failure mode (Section 1.6).
`[Guardian] Duplicate mediation resolution ignored (already recorded).`	The background dedupe (v0.18.11); one Override per gate.
`[Guardian] Amnesia sweep: N scanned, N decayed, N strata rewritten.`	The four-hourly decay alarm.

Table 7.1 — Console log taxonomy (additions and changes since the second edition in place). Filtering tip on Gemini: filter by [Guardian] with Info level enabled; the page's own ad beacons match a bare guardian text filter.

7.3 The three consoles

The page console shows only the content-script side. The kernel and the semantic rail log to the service-worker console (chrome://extensions -> Inspect views: service worker). The engines log to the offscreen console, in the same Inspect views list once the document exists: tier-chain walk, anchor builds, golden-pair verdicts, and model lifecycle live there.

7.4 The audit trail

The audit recorder logs session lifecycle, every Council convening, interventions and their outcomes (overridden — once per gate — versus stepped-away, with at-the-gate durations), governor actions including every rate-window suppression and every replay-mediation softening, and the semantic rail's recorded interventions including silent concurrence. It is the transparency substrate referenced under EU AI Act Article 13 (Part 10). Like everything else, it is local.

Part 8 — Troubleshooting Runbook

Symptom-driven triage for the issues seen in practice — including every failure mode encountered while building the v0.18 line, so nobody pays for the same lesson twice. Begin with the console taxonomy (Table 7.1); for verdict issues use the service-worker console, for engine issues the offscreen console.

Symptom	Likely cause	Action
Errors page shows a powerPreference / WebGPU warning at boot	ORT 1.26-dev probed the GPU adapter before the WASM pin took effect (pre-v0.18.6).	Fixed by the boot.js WebGPU shadow guard + per-call EP pins; both are tripwire-tested. If ever seen again, the boot guard regressed — check boot.js loads first in offscreen.html.
[Guardian Stance] golden-pair verification failed: ...	Label mapping, weights, or the pair wire are suspect; the gate refused to arm the rail (correct behaviour).	The log names the failing pair and the e/n/c triple. Identical scores across pairs = pair wire (the v4 pipeline pair-blindness; the direct tokenizer+model path must be intact). A flipped expected/got = label mapping; verify the model config's id2label.
nli=unavailable in the self-test	Stance weights not present under models/ (they are user-provided).	Place models/Xenova/nli-deberta-v3-xsmall/ (config, tokenizer, onnx/model_uint8.onnx) into the package and reload. Similarity-only operation in the meantime is by design.
[Guardian Semantic] mpnet-base-v2 unavailable — trying next tier	Preferred-tier weights not present; the engine fell to the packaged floor.	Expected on a bare zip. Provide models/Xenova/all-mpnet-base-v2/ for the quality tier; verify the warm-up and self-test then name mpnet-base-v2.
Anchors recompute on a boot where nothing changed	Per-profile cache miss: profile id or anchor-set hash changed, or offscreen localStorage was cleared.	One recompute is normal after a tier change or anchor edit (keys are derived; no manual bump exists any more). Recurring recomputes with no change: inspect the swept-stale-caches log and the active profile id.
semantic=unloaded in an early self-test	Historical (pre-v0.18.8 lazy load): truthful but alarming.	Both rails now warm eagerly at boot; a fresh install should read ready with profiles named within seconds of the warm-up lines. If unloaded persists, read the offscreen console for the tier-chain walk.
Mediation modal will not dismiss; multiple Override rows in the audit	Historical (v0.18.10): the friction gate replaced the raw handler that owned dismissal; every click re-recorded.	Fixed in v0.18.11 (injector dismissal-first, gate single-fire with lockout, background 30 s dedupe). One click completes; one Override per gate. If recurrence: run the friction-gate suite first.
Self-harm mediation on a large paste that merely quotes or discusses the subject	The lexical floor matched; the Advocate never counterweights self-harm (by design); stance adjudication does not currently sit in front of KERNEL verdicts.	Working as specified; the concurrence fossil records the embedding rail's (often lower) score as visible dissent. Whether kernel self-harm mediation should consider stance is a taxonomy/normative call, queued with the v0.19 council integration — not an engineering default.
Pattern Analyst cites a high message rate after switching between conversations	Conversation switches can re-emit history under fresh element ids; whether all such re-emissions carry the replay tag is an OPEN audit item (12 June, 17:43 log).	Audit the injector's replay determination across onNewConversation before tuning any rate threshold. Recorded honestly as open.
ALERT: user turns visible but ZERO assistant elements match	Provider DOM rename/restructure (or a closed shadow root / iframe).	The ALERT now prints the suspected assistant rows and the recovery tier keeps coverage alive. Send the log; pin permanent selectors from the evidence; the tier stands down on its own when they land.
Health Degraded/Failed on a quiet page	Historical (pre-v0.18.8 stall timer).	Removed; quiet is not failure (Section 1.6). Only cold-dead (active user, zero detections ever) cascades. If seen on a modern build, run the health-monitor suite.
Mid-session, every turn returns null/no-response; refresh fixes everything	Orphaned content script: the extension was reloaded/updated while the tab sat open.	Expected fail-open. Hard-refresh chat tabs after every extension reload — stale tabs also produce convincingly old log shapes (old self-test format, retired log lines); check the version banner before debugging "regressions".
Gemini: pre-screen worker skipped at boot	Page CSP bans blob: workers.	Expected; logged once; the background path carries full coverage.
First semantic verdict after install is slow	Cold start: tier load + one-time anchor embedding (mpnet: ~2 s + ~8–10 s).	Expected; anchors cache per profile, both rails stay warm and are pre-warmed at boot thereafter. No action.

Symptom	Likely cause	Action
Commentary rendered but dashboard count unchanged — or vice versa	A rail bypassed the constitutional path.	Verify gateExternalCommentary -> recordSemanticIntervention in primary mode; -slm-cooled and -slm-concur silent fossils are correct suppressions, not bugs.
Commentary budget exhausted the moment a page loads	History replay consumed the rate window: the replay tag is not reaching the Governor.	Turns flagged as replay must carry the tag end-to-end; replays are exempt by design and excluded from pattern memory. Check the injector tag and governor.replayTurn.

Table 8.1 — Troubleshooting runbook. Historical rows are retained deliberately: the symptom recurring is how a regression announces itself.

Part 9 — Security and Privacy Posture

9.1 Local-only, zero telemetry — models included

The Guardian has no backend. It opens no network connections for its safety function, holds no API keys, and transmits nothing. The on-device models strengthen rather than strain this claim: the floor model ships inside the package, the larger models are placed by the operator, allowRemoteModels is false, the WASM runtime loads from packaged assets via chrome.runtime.getURL, and the build verifies no CDN fetch paths survive. There is no account and no server-side record. Data that is never collected cannot be leaked, subpoenaed, or sold.

9.2 On-device memory — a complete inventory

Datum	Where / lifetime	Contents
Fossils (Strata)	Encrypted, chrome.storage; decays at 180 days	Category, severity, verdicts, model profile id, confidences and stance probabilities for semantic strata. No message text, from any layer.
Audit trail	Local storage	Convenings, interventions, outcomes, governor actions (incl. replay softenings), semantic records. No message text.
Anchor-embedding caches	Offscreen localStorage; keyed per (profile, anchor-set); stale keys swept	Embeddings of the fixed anchor phrases in each tier’s space. Static tooling data — contains nothing of the user’s.
Pattern-tracker memory	Offscreen document, in memory only	Embeddings + category scores of the last 12 live user turns per session (8 sessions, LRU; replays excluded). Never written to disk; evaporates with the document.
Mediation-resolution dedupe	Service worker, in memory only	sessionId -> last-resolution timestamp (30 s window). No content.
Settings	chrome.storage.local	The five user preferences.

Table 9.1 — Memory inventory. The pattern tracker remains the only place user-derived content exists in any form: volatile, session-scoped, embedding-only.

9.3 Content-security and isolation

The engine bundle contains no eval or new Function, satisfying MV3 CSP, and ships with both ORT loader paths packaged (no runtime code fetch). The WebGPU surface is shadowed in the offscreen document before any engine code runs. The mediation overlay is mounted in a closed shadow root, isolating it from the host page. The friction gate’s diagnostics never echo keystrokes or typed values. The model and runtime assets are web-accessible resources readable by the extension’s own pages; they are inert data and public models.

9.4 Threat-model note: serving the person

The single most important security property is architectural: the Guardian answers to the user, not to a central operator. There is no remote configuration channel that could re-target detection or exfiltrate the distilled psychological-risk profile — and with the pattern tracker sketching a user’s conversational trajectory in memory and the stance rail adjudicating what they assert, that profile is richer than it was, which makes the absence of any channel to move it matter more, not less. Any future enterprise capability must be a separate, consented product — never a remote policy push into this one. Any roadmap item that smells like “remote config” must be reconciled against this section first, in writing.

Part 10 — Compliance (EU AI Act)

The Guardian's design maps cleanly onto the relevant obligations. The regulation does not demand a correct oracle; it demands a defensible, monitored, revisable assurance process.

Article	Obligation	How the Guardian addresses it
Article 9	Risk management as a continuous, documented process.	A deterministic core, a 255-test behavioural suite with protected invariants, live calibration instrumentation (the below-threshold and concurrence logs), an openly carried calibration debt with a defined retirement gate, and the eval-harness programme (Part 11).
Article 11	Technical documentation.	This guide is the spine of the Article 11 file: architecture, data and memory inventory, model cards for all three on-device models (identity, quantisation, provisioning, anchor methodology, measured figures and their honest gaps), the change history, and the assurance posture.
Article 13	Transparency and information provision.	The audit trail records every convening, intervention, governor action, replay softening, and semantic catch — including suppressed (-slm-cooled) and silent-concurrence (-slm-concur) ones: the system can evidence not only what it did but what it chose not to do, and why. Each semantic stratum names its model profile and confidence.
Article 14	Human oversight; a human makes the final decision.	The friction gate: a mediation can pause but never decides for the user, who always retains the choice to proceed — now exactly once per gate, recorded exactly once. The Settings tab extends oversight to the detectors themselves.

Table 10.1 — EU AI Act mapping.

> **Deadline**
> Compliance documentation against Articles 9, 11, 13, and 14 should be completed ahead of the August 2026 deadline. This guide is the skeleton; the evidence is the test suite, the audit substrate, and the harness output of Part 11. The v0.18 line adds one duty per model: three model cards now belong in the Article 11 file, and the mpnet card must state the calibration debt rather than inherit MiniLM's measured thresholds.

Part 11 — Testing and Assurance

11.1 What "correct" means here

A detector over unbounded human language has no closed-form correctness. The honest target is graded, evidenced confidence with the residual uncertainty stated plainly — which is also exactly what Article 9 asks for. Determinism gives reproducibility and non-regression for free, but a deterministic wrong rule is reliably wrong, so the soundness of the rules — and of the anchors, thresholds, stance templates, and decision margins — must be earned empirically.

11.2 Existing coverage

At v0.18.11 the source tree carries 255 passing tests across the long-standing council, governor, amnesia, and signal-floor suites plus the suites the v0.18 campaign added, each pinning a paid-for lesson:

- claude-adapter (live DOM shapes, drift sentinels, the structural recovery tier incl. the renamed-DOM incident recreation); deep-extract and stream-batcher extraction contracts.
- health-monitor (quiet-is-not-failure, cold-dead cascade, recovery); settings-parity (sidepanel defaults literal-parity); replay-behavioural-metrics (the false-mediation incident recreation).
- model-profiles (registry integrity, dimension guard, derived cache keys, tier-chain order with a packaged floor, the deliberately-absent mpnet baseThreshold); stance (label-order permutations, golden-pair failure detail, decision rule, calibration scaffolding); stance-cascade; wasm-pin (EP pins on every load path; the text-classification pipeline ban).
- semantic-deliberation (real evaluator rows carried into -slm fossils); semantic-concurrence (silent fossil, no gate, no annotation, note contracts); friction-gate (single-fire across all paths incl. the thirteen-click recreation, dismissal, step-away integrity); governor rate window; composer alignment; chunker; pattern tracker; offscreen bridge.

11.3 The Council-level eval harness (the mandatory next build)

The outstanding piece is unchanged in shape and raised in urgency: whole snippets fed through deliberation plus both semantic passes, scoring verdicts and resolved levels against expected labels. The mpnet activation makes the rerun mandatory rather than aspirational — every embedding threshold and category offset in the system is currently a

MiniLM-space artefact governing an mpnet-space score. The harness should report:

- Per-category precision and recall and a level-confusion matrix — measured separately in each embedding space, never pooled.
- Threshold and offset calibration swept against should-mediate / should-comment / should-pass labels, seeded from the live below-threshold and [concurrence] logs — a labelled-miss feed that now covers flagged turns too. The first mpnet census: which of the 75 anchors are starving, which neighbourhoods are missing.
- A benign-context suite (fiction, academic, recovery, third-party narration) with stance expectations: a quoted novel must read mention or ambiguous, and mediating on it is worse than missing a subtle case.
- Multi-turn fixtures: reformulation sequences that must trip the persistence bonus, slow boils that must flag, replay prefixes that must pollute nothing.
- Metamorphic invariants checkable without a perfect label: negation must flip a verdict (the open lexical-negation gap of Section 3.2 makes this a known-failing fixture to be driven green, not a hope); adding benign text must not raise severity; a recovery framing must damp; a replay must never consume budget nor feed pattern memory; every visible intervention must leave a fossil and every suppressed or concurrent catch a silent one; a kernel-intervened turn must still classify and must never double-annotate; the golden-pair gate must block an inverted mapping; and self-harm must always reach Mediation on a live turn — the privileged invariant, which must fail loudly the instant it ever does not, no matter which rail fired.
- Right-reason checks: every member records a rationale, the Pattern Analyst slot carries real temporal evidence, and stance attaches its probabilities — assert a verdict fired for the right reason, not a coincidental match.

11.4 The irreducible limits

Two things genuinely cannot be measured, and should be stated rather than papered over. First, ground truth on the contested categories is itself a judgement; past a point it needs human adjudication — ideally several raters, and clinical eyes for the self-harm taxonomy — and one measures human-versus-human disagreement as the irreducible floor beneath which “accuracy” stops meaning anything. Second, whether an intervention actually helps the person is the real correctness, and the local, no-telemetry design rightly precludes measuring it at scale; the only honest mitigation is conservative, restrained, optional-by-default behaviour — which is what the Governor and the gate are, not evidence that it works.

> **Two questions, kept separate**
>
> Verification can show the Council conforms to its specification; it cannot show the specification is right. What should count as worth flagging — the choice to privilege self-harm, the decision to soften replayed history, whether stance should ever sit in front of a kernel verdict — are normative stances from the project whitepaper: defensible, not provable, and decided deliberately by the principal. Conflating “does it meet spec” with “is the spec sound” is precisely how a safety system talks itself into believing it is fine.

Appendix A — Source Map

File	Responsibility
`src/core/spiral-kernel.ts`	Kernel orchestration; preFlight/postFlight; lastDeliberation (real evaluator rows for the semantic rail); recordSemanticIntervention; setClassifier (the vacant council seat).
`src/core/council/*`	Sentinel, Advocate, Historian (intensification-gated), PatternAnalyst; deliberation and resolution; the additive semantic contract.
`src/core/council/signals/*`	The regex floor (harm patterns, benign patterns, conversation-level).
`src/core/council/semantic/model-profiles.ts`	The model registry; PREFERRED/DEFAULT ids; embeddingProfileChain with packaged-floor guard; vector provenance (dims guard, derived anchor-cache keys).
`src/core/council/semantic/stance.ts / stance-templates.ts`	NLI label-order resolution; logitsToScores (index-mapped softmax); golden pairs with failure detail; decision rule; per-category assert/mention templates; premise budget.
`src/core/governor/governor.ts`	Gating, per-category commentary window, replayTurn, cap provider, gateExternalCommentary, circuit breaker, self-report.
`src/core/governor/compliance.ts`	EU AI Act record stamping; GUARDIAN_VERSION (a versioning site — Section 5.4).
`src/core/fossil-memory/*`	Strata, encryption, Amnesia Net, alpha indices.
`src/extension/background.ts`	Message handler; offscreen bootstrap; eager engine warm-up; replay threading; semantic rail wiring (primary awaited, concurrence fire-and-forget); replay-mediation softening; mediation-resolution dedupe; alarms.
`src/extension/settings.ts`	Settings schema, live mirror (guardianCfg), migrations (incl. the stance-default repair).
`src/extension/semantic-fallback.ts`	The constitutional path of the semantic rail: classify -> stance -> gate/fossilise -> annotate; concurrence mode; buildSlmNote; the loud catch.
`src/extension/offscreen/boot.js`	Boot beacon; load-error capture; ping/status until engine registration; WebGPU shadow guard.
`src/extension/offscreen/offscreen.ts`	The engines: environment invariants, tier-chain loading, anchors, classification, NLI session (direct tokenizer+model), warm-up handlers, status.
`src/extension/offscreen/chunker.ts / pattern-tracker.ts / runtime-anchors.ts`	Sentence chunking (240/20/400/24); Semantic Pattern Analyst (constants, session memory); the 75-anchor index.
`src/extension/offscreen/bridge-classifier.ts`	OffscreenBridgeClassifier — the in-council classifier seat. Exists, tested at the bridge level, deliberately UNWIRED: the v0.19 integration scaffolding.
`src/extension/content/injector.ts`	DOM pipeline; routing; replay tagging; rendering; mediation flow incl. dismissal-first gated proceed.
`src/extension/content/adapters/*`	Platform adapters (claude-ai with sentinels + recovery tier; gemini; chatgpt) and the semantic fallback adapter; base extraction helpers (describeElement, deep extraction).
`src/extension/content/health-monitor.ts`	Cold-dead detection only; the quiet-is-not-failure doctrine header.
`src/extension/content/overlay.ts / biometric-gate.ts`	Dreamstate overlay and status indicator; the friction gate (wait -> WebAuthn/typed -> single-fire proceed).
`src/extension/sidepanel.html / .js`	Dashboard, fossils (incl. Semantic Sentinel and stance rendering by note markers), Settings, live Engine box.
`scripts/postbuild-offscreen.mjs / assemble-extension.mjs / collect-licenses.mjs`	The two mandatory ORT path patches; layout + manifest gate; the self-gating attribution collector.

Table A.1 — Source map.

Appendix B — Enumeration Reference

Categories. SelfHarm, DependencyExploitation, RealityDetachment, Manipulation, PrivacyErosion, AutonomyUndermining, EmotionalExploitation, InformationHazard.

Verdicts. Pass, Emerging, Flag, Escalation, NoPrecedent (per-evaluator outputs combined during resolution).

Levels. PassThrough, SilentObservation, Commentary, Mediation.

Roles. Sentinel, Advocate, Historian, PatternAnalyst, SemanticSentinel. The SemanticSentinel role identifies the embedding rail's evaluation block in fossils, naming the ACTIVE model profile (mpnet-base-v2 or minilm-l6-v2) and,

when adjudicated, the stance attachment.

Stance labels. assert, mention, ambiguous — presentation-scope only.

Note markers. -slm (primary catch), -slm-cooled (governor-suppressed), -slm-concur (silent concurrence beside a kernel verdict), -assert / -mention (stance), -replay (replayed history). Built by one constructor; parsed by the side panel; pinned by tests.

Appendix C — Change History

Version	Change
0.17.1–.5	The Gemini “Luminous” adapter campaign (second edition, Part 1.5): inner-node rule, self-or-descendant matching, wrapper multiplicity guards, deep shadow-root query, element-scoped streaming, scan diagnostics + reachability ALERT.
0.17.6–.14	Deployed-bundle continuation, since ported: adapter refinements and the replay-mediation softening (softenReplayMediation — presentation-only, audited; Section 2.5.2 records the decided scope).
0.18.0–.3	The profile-driven engine: model registry with vector provenance (dims guard, profile-tagged vectors, derived per-profile anchor-cache keys superseding the manual key bump); NLI stance architecture (label-order resolution, golden pairs, templates, decision rule) and its wiring into the semantic rail with -slm-assert/-mention notes and presentation softening; offscreen boot beacon with ping and read-only status (a status probe must never trigger a model load); protocolVersion 2.
0.18.4	Claude adapter rewritten: substring-safe selectors for !-prefixed Tailwind tokens, data-is-streaming as primary anchor, WeakMap element identity (role:eN), guards-before-normalisation, drift/reachability sentinels settling across 3 scans with describeElement evidence. Gemini ALERT settle. Pre-screen worker CSP skip on Gemini with a single boot log. stanceEnabled default true + migration.
0.18.5	Replay correctness: replay flag threaded end-to-end; behavioural gate and reformulation input exclude replays; liveMessageCount for rate metrics (a replayed history can no longer manufacture an 11/min “rate”). Sidepanel defaults parity (the silent stance kill) + literal-parity test + storage repair migration. GUARDIAN_VERSION added to the bump checklist after nine stale releases.
0.18.6	WebGPU shadow guard in boot.js + executionProviders pins on every model load; the Errors page restored as a meaningful diagnostic surface; wasm-pin tripwire suite.
0.18.7	NLI live: the transformers v4 pair-blindness diagnosed from library source and fixed by driving tokenizer + model directly with index-mapped label resolution (logitsToScores); golden-pair failures self-describe (pair + expectation + full triple); pipeline('text-classification') banned by test. First green golden gate on real weights.
0.18.8	Eager warm-up for both rails with profile-named logs and self-test; the embedding tier chain (mpnet preferred / packaged MiniLM floor) with graceful fallback; mpnet activated with its calibration debt explicitly recorded (absent baseThreshold pinned by test); health-monitor stall timer removed — quiet is not failure; cold-dead cascade retained; doctrine + four tests.
0.18.9	Claude reachability loss live: structural recovery tier (user-anchor foothold, LCA / sibling-walk turn-list derivation, stability settle, gated heartbeat, evidence-bearing ALERT, automatic stand-down). Third-party licensing: self-gating collect-licenses build step + licenses/ in the package. Live Engine box (read-only engine status) + attribution pointer.
0.18.10	Concurrence: the semantic rail classifies every turn; kernel-intervened turns get a fire-and-forget silent pass fossilising -slm-concur with the real council rows and stance; one-annotation-per-turn preserved; the pattern tracker fed at flagged moments; buildSlmNote unifies note construction; the rail’s silent catch made loud (and immediately caught a fossil-killing latency-field bug, hardened same release).
0.18.11	Friction gate single-fire + immediate dismissal (the thirteen-Override incident): injector dismisses before any background ack; every gate path locks on first activation; background dedupes resolutions (one Override per gate, 30 s window); step-away integrity pinned. Three tests including the literal thirteen-click recreation.

Table C.1 — Change history relevant to this guide. For 0.13.x–0.17.0, see the second edition’s Appendix C.

Appendix D — Technical Glossary

Term	Definition
Alpha-index	A structured severity object for a resolved category; the basis of the response level and the unit fossilised.
Council / Bouncer / Governor	The evaluator panel; the deliberation-to-candidate adaptor; the restraint layer (rate windows, cooldowns, circuit-break).
Strata / Fossil / Amnesia Net	The on-device, text-free, decaying memory; one persisted observation; the scheduled decay.
Fail-open	On any internal failure — worker, engine, tier, or rail — the turn passes unrendered; the Guardian never blocks on its own failure.

Term	Definition
Model profile / provenance	The registry entry describing an on-device model; every vector carries its profile id and cross-profile comparison is a hard error.
Tier chain / floor	The embedding load order: preferred quality tier first, packaged floor last; the floor must be a shipped profile by construction.
Anchor / chunking / category offset	A curated phrase defining a point of a category's meaning-space (weighted top-3 similarity); sentence-level splitting with per-category max-pooling; a per-category threshold addition where anchors border benign discourse.
Reformulation / trajectory / persistence bonus	Near-identical re-asks (cosine >= 0.86, twice+); a near-monotonic per-category rise; the threshold reduction (-0.06, floor 0.33) either grants.
Stance (assert / mention / ambiguous)	NLI adjudication of flagged content: declared, or quoted/discussed, or unresolved. Softens presentation only; never self-harm wording; never a verdict.
Golden pairs	The benign sanity set every NLI session must score correctly before the stance rail arms; failures name pair and probability triple.
Primary / concurrence	The semantic rail's two modes: verdict-bearing on kernel-passed turns; silent, record-bearing on kernel-intervened turns.
Constitutional path	The invariant spine of every visible intervention: gate -> fossil -> audit -> annotate. Suppressed and concurrent catches still fossilise.
Replay turn / replay softening	A re-scanned history turn: evaluated and fossilised normally, exempt from budgets and pattern memory; a replayed mediation softens to commentary, audited.
Reachability ALERT / recovery tier	The user-rows-yes assistant-rows-zero diagnostic, now evidence-bearing; the provider-name-free structural fallback it activates.
Quiet is not failure	The health-monitor doctrine: an adapter with >= 1 detection is presumed competent; only cold-dead cascades; drift belongs to the sentinels.
Calibration debt	The explicitly recorded gap between the active embedding space and the space its thresholds were tuned in; carried visibly (absent registry field, pinned by test) until the harness rerun retires it.
One-log diagnosis	The instrumentation doctrine: a firing sentinel or failing gate must name elements, pairs, or scores such that one log is a complete bug report.
Single-fire	A user decision control may invoke its consequence exactly once; the v0.18.11 friction-gate contract.

Table D.1 — Technical glossary.

Ethics First. Always.