GEMINI

Helix Constitution — Universal GEMINI.md

Field	Value
Revision	
Created	- -
Last modified	- - T : : Z
Status	active

Status summary	Added § . . — Device-independent host-side rendered-UI visual-proof mandate (User mandate - -): every change to ANY user-facing UI surface MUST be proven by DEVICE-INDEPENDENT host-side RENDERED PIXELS before claimed correct — the real component rendered to a PNG ON THE HOST (no device/emulator/running app; Compose→Roborazzi/Paparazzi, web→Playwright/Storybook, SwiftUI→snapshot-testing, equiv per stack) for EVERY screen×state×{light,dark} theme, dual-validated by (i) golden image-diff AND (ii) an OCR/vision oracle reading rendered text+labels+control bounds (NO overlap / label-over-label / clipping / off-screen / collapsed-or-giant-unbounded widget). VALUE/token-equality / property-assertion UI tests are FORBIDDEN as the PROOF a UI is correct (forensic FACT - - : hex/sp/dp value-equality unit tests stayed GREEN while the operator opened a broken giant-button screen) — may supplement, NEVER substitute the rendered-pixel proof; self-validated golden-good/golden-bad analyzer § . . ( ); device-offline is NEVER a valid skip (host-render IS the device-independent path); COMPLEMENTS not replaces § . . /. /. /. live-device recording + § . . OpenDesign tokens + § . . exported-doc visual validation. Propagation gate CM-COVENANT- - -PROPAGATION + recommended gate CM-HOST-RENDERED-UI-VISUAL-PROOF + paired § . mutation. Classification: universal (§ . .). Prior round: § . . REPEALED (operator decision - -): Universal Semgrep static-analysis mandate repealed — Semgrep no longer mandatory; scaffolding, submodules/semgrep, MCP/

pre-commit/PATH wiring, docs_chain semgrep_status context, and CM-COVENANT- - -PROPAGATION/CM-SEMGREP-WIRED gates removed. Added § . . mirror — Universal Media Validation: every recorded artifact MUST pass MEDIA VALIDATION pipeline (OCR/transcription/text parsing vs patterns, self-validated analyzer, structured verdict, pinpoint on FAIL, real-time trigger, § . mutation). Added § . . mirror — Universal Auto-Propagation Hook: post_update_hook.sh detects/registers/installs skills/MCP/hooks/scripts; consumer invokes after every pull. Added § . . mirror — Universal Independent Verification Agent: every output passes INDEPENDENT verifier (§ . . /.), zero-finding GO. All Classification: universal (§ . .). Prior round: Added § . . - mirror entries — Vision-verified recording + HelixQA bridge (User mandate - -): every video recording MUST be processed through a vision/OCR pipeline reading on-screen content against SPECIFY-phase expected patterns; ≤ s frame capture, self-validated analyzer, per-frame PASS/FAIL with evidence path. Rootless container runtime (User mandate -): Podman rootless mandatory; rootful Docker/sudo FORBIDDEN unless § . . ; vasic-digital/containers sole orchestration layer. OpenDesign UI design system (User mandate - -): mandatory UI design-and-refinement system for all user-facing interfaces; design tokens for color/typography/spacing; light+dark themes; extend upstream per § . . . All three Classification: universal (§ . .). GEMINI.md mirrors updated in lockstep per § . . . Prior round: New Gemini CLI carrier of the Helix

Field	Value
	Constitution — added by § . . (universal action-prefix system) so Gemini CLI gets the LAYER- action-prefix recognition floor out of the box, plus the § . . token-efficiency mandate mirror. This file is one of the four canonical agent context carriers (`CLAUDE.md` , `AGENTS.md` , `QWEN.md` , `GEMINI.md`) the constitution maintains per § . . ; the textual identity of the action-prefix recognition block across all four IS the LAYER- universality. This GEMINI.md is a thin carrier — the full universal rule set is in `Constitution.md` (canonical) and the sibling carriers; Gemini CLI reads this file as its context carrier. Back-filled § . . - to lockstep (§ . .).
Issues	none
Issues summary	—
Fixed	§ . . + § . . mirrors
Fixed summary	Carrier created in lockstep with the Constitution.md § . . + § . . additions.
Continuation	—

How inheritance works

This GEMINI.md is the **Gemini CLI carrier** of the Helix Constitution submodule. It is one of the four canonical agent context carriers (`constitution/CLAUDE.md` , `constitution/AGENTS.md` , `constitution/QWEN.md` , `constitution/GEMINI.md`) — these ARE the canonical root per § . . , NOT consumer

extensions. The full universal rule set lives in `constitution/Constitution.md`; the sibling carriers (`CLAUDE.md` / `AGENTS.md` / `QWEN.md`) carry the same content for their respective agents. A consuming project's own repo-root `GEMINI.md` (if any) opens with the inheritance pointer and adds only project-specific rules; THIS file is the canonical carrier and does not.

For agents driven from a consuming project, the project's repo-root carrier SHOULD start with a clearly-marked inheritance pointer:

```
## INHERITED FROM constitution/GEMINI.md

All rules in `constitution/GEMINI.md` (and the `constitution/
        Constitution.md`
it references) apply unconditionally. The project-specific rules below
extend them.
```

Action-prefix recognition (§ . . — LAYER-)

The block below is the **canonical LAYER- action-prefix recognition instruction**, embedded VERBATIM (identical text) into every agent context carrier the constitution maintains. That textual identity across all four carriers IS the LAYER- universality: Gemini CLI reads this block via this carrier and applies the prefix expansion itself, so the action-prefix system works on Gemini CLI with zero extra setup.

2026 06 09

2026 06 09

2026 07 02

§ . . — Universal action-prefix system (`ACTION_NAME ::`) (User mandate, - - ; GRAMMAR_ADDENDUM - - ; arrow form - -). When a user prompt's FIRST non-blank line starts with a recognised action prefix, you MUST: ( ) look the action token up in the action registry `constitution/actions/registry.yaml` (or `$HELIX_ACTION_REGISTRY`); ( ) if it is a registered action, REPLACE the prefix with that action's `expansion` text and apply its `rules` ; ( ) execute the remainder of the prompt under the expanded instruction. **Five EQUIVALENT forms** — same action, same expansion, same execution: ( ) `ACTION_NAME :: <rest>` (bare `::`), ( ) `PREFIX::ACTION_NAME :: <rest>` (namespaced `::`), ( ) `/ACTION_NAME <rest>` (bare slash), ( ) `/PREFIX::ACTION_NAME <rest>` (namespaced slash), ( ) `ACTION_NAME ---> <rest>` (bare arrow) ≡ `PREFIX::ACTION_NAME ---> <rest>` (namespaced arrow). Thus `BACKGROUND :: x` ≡ `DEFAULT::BACKGROUND :: x` ≡ `/BACKGROUND x` ≡ `/DEFAULT::BACKGROUND x` ≡ `BACKGROUND ---> x` ≡ `DEFAULT::BACKGROUND ---> x` . `PREFIX` is an action NAMESPACE; the reserved default namespace is **`DEFAULT`** , and an action runs WITH or WITHOUT the prefix. Grammar (all hold): anchored at the FIRST non-blank line only (mid-prose tokens never match); the action token AND the namespace are UPPERCASE-only `[A-Z][A-Z0-9_]*` (lowercase never matches); the namespace separator `::` inside the token carries NO surrounding spaces (`PREFIX::ACTION_NAME`), DISTINCT from the action-body separator `" :: "` (one ASCII space on each side of `::` — avoids C++ `Foo::Bar` , YAML `key: value` , URLs) in forms / , the arrow-body separator `" ---> "` (one ASCII space on each side) in form , and the slash-body separator (one space) in forms / ; stacked prefixes (`A :: B :: rest`) apply outer-to-inner, left-to-right (expand `A` , re-scan, expand `B` , then the residual is the task); a leading `\` escapes

the prefix for the `::`, arrow, and slash forms (`\BACKGROUND :: x`, `\BACKGROUND ---> x`, `\/BACKGROUND x` — treat literally, strip the backslash, NO expansion) so action names can be discussed. **Conflict rule (slash form):** `/ACTION_NAME` (form) is honored as the action ONLY when `ACTION_NAME` (case-folded) does not collide with a built-in/host slash command (registry `slash_bare: auto` + `slash_conflicts: [..]`); form (`/PREFIX::ACTION_NAME`) is ALWAYS unambiguous and always honored. An unknown token that matches the grammar shape (any of the forms) but is NOT registered is NEVER silently expanded or silently dropped — ask which registered action was meant (§ . . / § . .) or treat it as a literal prompt, NEVER invent an expansion (§ . .); any prompt not satisfying the grammar is an ordinary prompt and the system is a no-op. The registered action **`BACKGROUND`** expands to: *"The following prompt that we will provide MUST BE executed in background in parallel with all main work streams using the subagents-driven development approach! All work done MUST PRODUCE rock solid evidence covered with hard physical proof(s) that all done is working as expected and as specified without any false results and without any bluff!"* (composes § . . / § . . subagent-driven, § . . / § . . parallel streams, § . . background execution, § . . / § . . / § . . captured physical evidence, § . anti-bluff). A second registered action **`REMINDER`** re-surfaces previously-scheduled, CRITICAL, status-UNCERTAIN work: FIRST verify the ACTUAL current status from captured evidence (never assume done or not-done — § . . no-guessing), THEN act on the delta — report the captured proof if genuinely complete, resume from the exact point if partial, action it NOW if not started, or surface the block per § . . /§ . . — always

> producing a status verdict (composes § . . / § . . / § . . / § . . / § . . / § . .). The system is UNIVERSAL (every CLI agent reads this block via its context carrier per § . .), extensible (new action = new registry row), decoupled + reusable (§ . .), and loads out-of-the-box. Classification: universal (§ . .). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-action-prefix`, `--ignore-prefix`, `--no-registry`, `--invent-expansion-OK`, `--single-layer-only` flag.

Token-efficiency mandate (§ . .)

§ . . — **Token-efficiency mandate (research-derived + operator mandate, - -).** Every project worked on by AI coding agents MUST cut token spend (input AND output) toward - **% of current (a - % reduction)** WITHOUT degrading quality/performance/safety or breaking any existing mechanism, via a composable, safety-ranked measure set: ( ) **prompt-cache the static governance prefix** — the always-loaded governance forms a byte-stable cache-breakpointed prefix with no volatile bytes ahead of it; cache reads cost ~ . × base input (the dominant cost driver — measured ~ K tokens of governance re-sent every turn, externally corroborated by Claude Code issue); caching is transparent so it removes no rule, weakens no gate, changes no verdict — only billing (PRIMARY, biggest + safest lever); the operative discipline is to **batch governance edits and keep CLAUDE.md + the constitution byte-stable mid-session** — every governance edit busts the prompt

cache (BASELINE.md finding); ( ) **subagent model-tiering + output-to-file** — mechanical non-judgment work (search/grep/status/doc-export/read-only probes) to a Haiku-class model, the strong model RESERVED for all reasoning/verdicts/fix-design (§ . .)/code-review (§ . .)/demotion (§ . .), large output persisted to a file not an inline – K-token transcript; the cheap model never emits a PASS so § . . + anti-bluff are untouched; ( ) **thin always-loaded INDEX + on-demand detail** — concise index (one line per fix/anchor, EACH carrying the literal `11.4.N` token so propagation gates pass) with the canonical full text kept gate-scanned in `constitution/Constitution.md` and reachable in one hop — a de-duplication realising § . . , never a deletion; ( ) **CodeGraph/retrieval-first over full-file loading** (§ . . / § . . /§ . .); ( ) **output-token reduction** — terse status + `effort:"low"` on the mechanical allowlist only; ( ) **tool-call batching + no re-reads**; ( ) **compaction/context-editing for long sessions. Mandatory measured proof:** a token-accounting harness measures tokens-per-development-cycle BEFORE vs AFTER on a frozen deterministic workload from the authoritative `usage` object (input/cache_read/cache_creation/output split; NEVER `tiktoken`, NEVER the client-side cost estimate), reproduced N times (§ . .), pass = AFTER ≤ % of BEFORE OR the measured best-safe reduction with a cited cold-cache reason; the AFTER run MUST show ZERO regression on the pre-build sweep + meta-test mutation sweep + propagation gates + a strong-model reasoning probe + a cache-warm proof (`cache_read_input_tokens > 0`) — cost reduction with quality regression is a § . FAIL. The headline number is the *measured* reduction, never the design estimate (§ . . / § . .). No measure may break/degrade any existing mechanism, and the rule is structured so none can. Composes § . . /. /. /.

Propagation gate `CM-COVENANT-114-141-PROPAGATION` (literal `11.4.141`) + recommended gate `CM-TOKEN-EFFICIENCY` + paired § . mutation (inject a pre-breakpoint volatile token → cache collapses → measured reduction falls below bar → gate FAILs). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-token-efficiency`, `--no-cache-governance`, `--assert-reduction-without-measuring`, `--tier-down-reasoning`, `--inline-all-governance`, `--tiktoken-estimate-OK` flag.

§ . . — Universal code-review mandate — every change reviewed, always, no exception (User mandate, - -). Verbatim operator mandate: "ALL changes we do MUST pass through the code review step!!! ALWAYS!!!" EVERY change made to ANY repository this Constitution governs — without exception — MUST pass through an INDEPENDENT code-review step BEFORE it is accepted, committed, or built. NO change class is exempt: source code, fixes, tests, gates, meta-test mutations, documentation, doc-tooling, build/CI scripts, configuration, governance files (Constitution / CLAUDE / AGENTS / QWEN), conductor main-stream edits, sub-agent output, refactors, single-line edits — if a diff exists, it gets an independent review. This is the ABSOLUTE form of § . . (code-review agent gate after a batch, before pre-build sweep + main build): § . . strips every implicit scoping § . . 's "after all fixes/changes/implementations are done" phrasing could leave open — no "just a doc edit", no "just a one-liner", no "the author already self-reviewed (§ . .)", no "trivial change" carve-out. **Independence is load-bearing** — the reviewer MUST be structurally separated from the author (a dedicated code-review agent, subagent-driven by default per § . . / § . . , or a distinct human), NEVER the author re-reading their own work; § . . 's multi-pass self-evaluation is the AUTHOR-side discipline and PRECEDES (never satisfies) § . . . **The review is**

itself anti-bluff (§ . / § . .) — a rubber-stamp "looks good" is not a review; it MUST genuinely analyse correctness, safety (no host § / data § / target-hardware § . . regression), will-it-really-work (no solve-A-create-B), end-user behaviour (§ . / §), and test-genuineness (§ .), its conclusions captured evidence per § . / § . . , and **it iterates to a clean GO per § . .** — any finding (BLOCKING / nit / warning) re-arms the loop, acceptance only on ZERO new findings + ZERO warnings with rock-solid physical evidence. Honest boundary (§ . .): a passing review does NOT replace § . . four-layer runtime-signature verification nor the § . . full-suite retest — it is one of MULTIPLE STRONG LAYERS, and the FIRST one every change crosses. Composes § . . / § . . / § . / § . . / § . . / § . . / § . . / § . . / § . . / § . . / § . . / § . . Classification: universal (§ . .) — the consuming project supplies its reviewer-dispatch mechanism + change-acceptance seam (commit wrapper / merge queue / PR gate) per § . . . Propagation gate `CM-COVENANT-114-142-PROPAGATION` (literal `11.4.142`) + recommended gate `CM-EVERY-CHANGE-REVIEWED` (every accepted change carries a fresh independent-review-completed marker for its diff, produced by a reviewer distinct from the author, before acceptance/commit/build) + paired § . mutation (strip the literal → propagation gate FAILs; accept a diff with no independent-review marker → `CM-EVERY-CHANGE-REVIEWED` FAILs). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-review`, `--no-review`, `--trivial-change-exempt`, `--doc-edit-exempt`, `--self-review-suffices`, `--review-after-commit` flag.

§ . . — Real-user-journey mandate for video-streaming-app full-automation tests (User mandate, - -). Verbatim operator mandate: "All video streaming apps ... require to choose some title and to press proper UI button to start or resume playing! Proper UI interaction to play exact show with proper content and subtitles is MANDATORY! Without it we just eventually play on nd display sample, and that's it mostly! THIS MUST BE ADDED as MANDATORY RULE regarding testing of any video streaming app in general with full automation tests! Universal in root constitution + a consuming project's extensions." Any full-automation test that asserts a video player / streaming application plays content MUST drive the REAL end-user journey through the app's OWN UI — launch → BROWSE the actual catalog → choose a SPECIFIC title → press the real Play/Resume button → confirm THAT chosen content is genuinely playing, with its correct subtitles, on the intended routing target. A test that bypasses the journey with a sample / demo / built-in-loop clip, a deep-link / `am start -a VIEW` / intent shortcut, a synthetic or pre-staged stream, or any path that does NOT exercise the app's own browse-select-play UI is a § . PASS-bluff at the user-journey layer: it validates ROUTING (that *something* reaches the display) while leaving the user-visible behaviour the operator cares about — "the show I picked actually plays" — unproven (the operator's "we just eventually play on nd display sample, and that's it mostly" gap). The mandate (ALL hold): ( ) **Real journey, not a shortcut** — launch → catalog browse → specific-title selection → real Play/Resume press through the app's own UI (§ . . UI-driven), NEVER a deep-link / intent / sample / loop-clip shortcut; ( ) **Chosen-content confirmation** — the PASS proves the SPECIFIC selected title is playing (not merely that pixels move on the target) via the § . . liveness battery on the § . . real-content path + the § . . subtitle content-correctness oracle for the chosen title's captions; ( ) **Non-**

introspectable UIs use the pixel oracle — when the app's accessibility hierarchy is blank/unreliable (TV-Compose / leanback / canvas / GL), DRIVE input + ASSERT content via the § . . CV/OCR pixel oracle, never a hierarchy-only tool; ( ) **Login via the credential single-source** (§ . .), never hardcoded, never logged; ( ) **Honest SKIP, never a faked PASS** — where the autonomous journey is genuinely infeasible (hard human-only login / CAPTCHA, geo-block per § . . , secure-surface pixel-blanking per § . .) the test is `operator_attended` SKIP-with-reason per § . . + § . . with a tracked migration item — NEVER a metadata-only / sample-played / routing-only PASS. Honest boundary (§ . .): "the routing fired so the title is playing" is a guess — only the chosen-content liveness + subtitle oracle on the real journey proves it. Composes § . . / § . . / § . . / § . . / § . . / § . . / § . . / § . . Classification: universal (§ . .) — the consuming project supplies its concrete app roster, login-credential source, routing target, and UI-driving / pixel-oracle harness per § . . . Propagation gate `CM-COVENANT-114-143-PROPAGATION` (literal `11.4.143`) + recommended gate `CM-VIDEO-REAL-JOURNEY-TEST` (every video-streaming-app playback test drives the real browse-select-play journey + confirms the chosen content + subtitles, or SKIPs-with-reason) + paired § . mutation (replace a real-journey test's browse-select-play path with a deep-link / sample shortcut → gate FAILs; strip the literal → propagation gate FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--sample-playback-ok`, `--skip-real-journey`, `--deep-link-suffices`, `--routing-only-pass`, `--no-title-selection` flag.

§ . . — Tracked/recorded-device availability-following mandate (User mandate, - -). Direct operator mandate: every device the project is tracking / following / recording (test / debug / manual-testing device, across every reachable transport — USB / wireless ADB / SSH / serial / network introspection API) MUST be availability-FOLLOWED — its connection state continuously monitored, any drop handled, never silently abandoned and never presented as a continuous recording. The § . . always-on recorder KNOWS a tracked device is absent (its per-device loop guards on the reachable state) but, lacking a following discipline, merely spins idle — no data captured, no offline event logged, no resume, no escalation — so the recording corpus presents a continuous timeline with a silent hole, a § . PASS-bluff at the recording-integrity layer (it claims continuous capture while no data exists for the gap). On a tracked device leaving its reachable state the system MUST, automatically: (a) **DETECT** the drop and **log an honest offline event** into the recording corpus (§ . . — a silent gap presented as continuous capture is a fabricated-continuity bluff; § . . — a silent recording gap defeats always-on recording); (b) **WAIT** for the device to return using the project's ALREADY-DEFINED reconnection timings — never invented numbers (§ . .), the SAME grace / reconnect / poll budgets the project's recovery path already uses; (c) **RE-ATTACH** — resume recording / tracking the moment the device returns and log an honest online / resume event; (d) **ESCALATE** to the project's sanctioned device-recovery path (per § . . feature class `device_recovery`) if the device does not return within the defined timeout — through the sanctioned, authorization-gated recovery entry point ONLY, never bypassing its gate and never performing a destructive recovery (e.g. a power-cycle) autonomously without that authorization (§ . . / § . . — high-blast-radius recovery is gated; while blocked the system keeps following the device, logging the

blocked-escalation honestly). A tracked-device drop that produces a silent corpus hole, a never-resumed recording, or a never-escalated permanent absence is the bluff this anchor forbids. Composes § . . (always-on recording — § . . closes its drop-handling gap) / § . . (`device_recovery` sink-side positive evidence) / § . . (honest offline / online events, reused-not-invented timings) / § . . (watchdog children reaped on stop) / § . . + § . . (gated, non-autonomous escalation). Classification: universal (§ . .) — the consuming project supplies its concrete tracking transport, device set, already-defined reconnection timings, and sanctioned recovery entry point per § . . . Propagation gate `CM-COVENANT-114-144-PROPAGATION` (literal `11.4.144`) + recommended gate `CM-DEVICE-AVAILABILITY-FOLLOWED` + paired § . meta-test mutation (strip the watchdog wiring / let an absence go unlogged → gate FAILs; strip the literal → propagation gate FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-availability-following` , `--silent-recording-gap-OK` , `--no-reconnect-wait` , `--invent-reconnect-timing` , `--skip-recovery-escalation` , `--autonomous-power-cycle-OK` flag.

§ . . — Independent multi-angle impact-research per change (User mandate, - -). For EVERY fix / change / new feature, INDEPENDENT impact-research agents (subagent-driven § . . /§ . . , structurally separate from the author — § . . self-eval PRECEDES, never satisfies — adversarial "refute-the-change" stance to defeat the documented LLM confirmation-bias failure mode) MUST research the change AND its connected/dependent features across a CLOSED SET OF EIGHT ANGLES — ( ) correctness/logic, ( ) regression (what existing working feature could break — via call-graph impact enumerating every direct + transitive caller and contract-dependent feature,

each mapped to its test), ( ) latent/dangerous-code — the recents class (runtime-only failure, interface/ABI/contract mismatch, concurrency/data-race, resource lifecycle), ( ) security (taint/injection/secret/unsafe-API), ( ) performance (latency/memory/thermal under load vs baseline), ( ) host/data/target-hardware safety per § /§ / § . . , ( ) cross-feature interaction (shared state/timing/hardware contention), ( ) business-logic conformance (matches the spec AND the connected features' contracts, not merely compiles). Forensic anchor (FACT, project-internal): the recents / -button-nav path shipped a latent AIDL-interface-contract mismatch that PASSED code review yet failed at RUNTIME ONLY (ANR on a recents-reaping gesture) because no review angle interrogated the interface contract / concurrency-lifecycle / silently-broken connected feature — the § . . SOURCE→ARTIFACT→RUNTIME→USER-VISIBLE gap caught only after the fact; § . . shifts that discovery LEFT. Each angle MUST name the TOOL(S) used + obtain the required DEPTH (the diff, the full changed unit, its declared contract, the spec section, every connected feature — NEVER the diff lines alone) + emit a captured-evidence conclusion per § . . /§ . . ("looks fine" forbidden, § . .); a genuinely-N/A angle is recorded `NOT_APPLICABLE: <reason>` per § . . , never silently skipped. Output = a per-change impact-research REPORT (one section per angle: tool + evidence path + verdict) + a single GO/NO-GO that BLOCKS acceptance/commit/build on ANY unmitigated risk in ANY angle; the change is fixed/mitigated + affected angles re-researched, iterating to a CLEAN GO (zero unmitigated risk + zero warning) per § . . before the § . . review gate. Honest boundary (§ . .): a GO proves cross-angle internal-consistency + bounded blast-radius — it does NOT replace § . . runtime-signature verification nor § . . full-suite retest; it is one of MULTIPLE STRONG LAYERS, the FIRST research pass every change crosses.

Composes/strengthens
§ . . /. /. /. /. /. /. /. /.
§ /§ /. /. /§ . . Classification: universal
(§ . .). Propagation gate `CM-COVENANT-114-145-PROPAGATION` (literal `11.4.145`) + recommended gate `CM-IMPACT-RESEARCH-PER-CHANGE` + paired § . meta-test mutation (strip the literal → propagation gate FAILs; accept a change with a report missing an angle or an unmitigated NO-GO → `CM-IMPACT-RESEARCH-PER-CHANGE` FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-impact-research`, `--single-angle-suffices`, `--author-researches-own-change`, `--diff-skim-OK`, `--no-go-overridable`, `--trivial-change-no-research` flag.

§ . . — Reproduce-first test + same-test-confirms-fix + mandatory extend-to-all-cases workflow (User mandate, - -). Every reported problem MUST be handled by a NAMED three-step test workflow — § . . does NOT re-author its component disciplines, it NAMES + BINDS them into the operator's exact per-defect sequence and adds ONLY the two emphases the components leave implicit. **(STEP — REPRODUCE-FIRST + INVESTIGATE)** before any fix, author the § . . RED test as a § . . RED-baseline-on-the-broken-artifact (reproduce the defect on the CURRENT pre-fix artifact, capture defect-present physical evidence per § . . /§ . . /§ . .); NEW EMPHASIS (D) that same RED test is ALSO a deliberate investigation instrument per § . . (A) — it MUST gather ADDITIONAL forensic data characterising the defect (triggers, boundaries, input/topology scope, adjacent failure modes) that feeds BOTH the fix design AND the STEP- extend scope; a RED test used only as a binary present/absent gate with no characterisation satisfies § . . but NOT § . . STEP . **(STEP — SAME-TEST-CONFIRMS-FIX)** after the fix the SAME test source (the § . . polarity switch flipped `RED_MODE=1→0`) confirms the defect

ABSENT — RED-on-broken then GREEN-on-fixed, both captured, on a clean target per § . . /§ . . , validated-first-after-redeploy per § . . , deterministic per § . . ; no separate happy-path test substitutes (the § . . /§ . . PASS-bluff). **(STEP — EXTEND-TO-ALL-CASES, mandatory per-fix)** NEW EMPHASIS (D) immediately after STEP the test MUST be FANNED OUT across the full case-space of the SAME functionality — all flows, valid + invalid + boundary (§ . . empty/max/off-by-one) + concurrent (§ . . contention) + failure-injection (§ . . chaos) + topology variants (§ . .) — confirming no issue of any kind, with PROVABLE enumerated coverage per § . . (a listed case-set with per-case outcome, never "no other issues found"); a REQUIRED step, NOT deferred to a release-cycle discovery sweep and NOT reduced to a single guard; each user-visible case carries rock-solid physical evidence per § . . + is registered into the § . . standing regression-guard suite; a newly-discovered fan-out defect triggers § . . test-interrupt + re-enters STEP . (ANTI-BLUFF, all steps) every PASS is rock-solid CAPTURED physical evidence per § . . (§ . . /§ . . / § . .) — metadata-only / config-only / absence-of-error / grep-without-runtime PASS forbidden (§ . / § . .); unclear validation method ⇒ deep-research-before-declaring-untestable per § . . /§ . . / § . . ; an operator-found defect the green suite missed triggers § . . . Honest boundary (§ . .): STEP reduces the unknown-unknown surface but does not prove zero remaining defects (§ . .) — un-exercised cases stated as honest gaps, never silently implied clean. Composes § . . /. /. /. /. /. /. /. /. § /§ . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-146-PROPAGATION` (literal

`11.4.146`) + recommended gate `CM-REPRODUCE-FIRST-THEN-EXTEND` (every closed defect's fix carries a § . . reproduce-first polarity test with captured defect-characterisation [STEP] + its `RED_MODE=0` GREEN confirmation [STEP] + an enumerated per-functionality extend case-set registered into the § . . suite [STEP]; a closure with the reproduce→confirm pair but NO enumerated extend case-set FAILs) + paired § . meta-test mutation (strip the literal → propagation gate FAILs; close a defect with only the reproduce→confirm pair and no extend-case-set → `CM-REPRODUCE-FIRST-THEN-EXTEND` FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-reproduce-first`, `--fix-without-red`, `--skip-extend-to-all-cases`, `--reproduce-confirm-suffices`, `--defer-extend-to-release-sweep`, `--single-guard-suffices`, `--no-defect-characterisation` flag.

§ . . — Crashed-agent respawn-until-complete + no-work-loss registry mandate (User mandate, - -). Verbatim operator intent: "any agent that crashed because of something MUST BE respawned and finish its work at some point! We MUST NOT lose any work, forget about or have it corrupted!" Forensic case study (FACT): dispatched subagents died on TRANSIENT causes (`API Error: Server is temporarily limiting requests` rate-limit killed at once, `API Error: socket connection closed unexpectedly` killed , one left PARTIAL edits — two source files written, the dependent SystemUI sliders + doc + test NOT done), each re-dispatched by pure conductor vigilance with NO mechanical guarantee — the moment conductor context is lost / compacted / the conductor itself crashes, an in-flight-but-dead work unit is silently dropped (lost / forgotten / corrupted). Every dispatched agent/subagent MUST be tracked through its full lifecycle so a crash NEVER loses, forgets, or corrupts its work; a crashed agent is NOT a completed agent (§ . . — "it probably finished before dying" is

a forbidden guess; abnormal termination is positive evidence the unit is OPEN). The mandate (ALL hold): **(a) durable agent REGISTRY** — every dispatched agent has an append-only machine-readable registry entry (stable id, task + § . . file-scope, declared output path(s) + § . . / § . . evidence dir, dispatch timestamp, status from the CLOSED SET `{dispatched | in-flight | crashed | respawned | complete}`), reusing the § . . sync substrate (JSONL event stream + atomic status snapshot; "an entry with no dispatch-event cannot show `complete`"); the registry is the SINGLE SOURCE OF TRUTH for "is this work owed?" and an unregistered dispatch is itself a violation; **(b) mechanical CRASH-DETECTION + RESPAWN-until-complete** — any abnormal termination (transient rate-limit/socket-close, any non-completion exit, or a terminal error after the runtime's own retries are exhausted) flips the entry to `crashed` + keeps the unit OPEN, and the unit is respawned (fresh agent claims the same id + scope + preserved state) and re-respawned until an agent reaches `complete`; respawn is the safe/reversible/bounded decision taken autonomously per § . . (never blocks the loop for a human), transient causes use the project's ALREADY-DEFINED backoff budgets (never invented, § . .), a deterministically-reproducible non-transient terminal error is investigated per § . . before further respawn (never a blind retry loop); **(c) PARTIAL-STATE preserve → § . . -check → resume-or-clean-restart** — the crashed agent's uncommitted edits + output doc are PRESERVED (never silently discarded → lost, never blindly committed → corruption), the respawn runs the § . . quiescence check on the preserved tree (every modified file accounted-for vs scope, no mutation/ `// always pass` / `_mutated_*` residue, no half-written/torn artifact), then EITHER RESUMES idempotently when it passes OR CLEAN-RESTARTS from a known-good base (§ . pre-op backup, reversible § . .) when inconsistent — nothing lost, nothing corrupted; per-agent `git worktree` isolation (§ . . L /

§ . .) keeps the partial tree from contaminating other streams; **(d) "crash ≠ done" COMPLETION criterion** — a unit is DONE only when an agent reaches `complete` with its required captured evidence/output landed (§ . . / § . . + the § . . verdict-carries-evidence-path rule); the endless-loop done-condition (§ . . /§ . . /§ . . /§ . .) MUST NOT read satisfied while any entry is `dispatched` / `in-flight` / `crashed` / `respawned` -not-yet- `complete`, the zero-idle survey (§ . .) treats every non- `complete` entry as an OPEN item to reclaim, and a registry showing `complete` without landed evidence is a § . PASS-bluff at the agent-lifecycle layer. Honest boundary (§ . .): the registry + respawn guarantee work is not lost/corrupted, NOT that it is correct — the respawned output still crosses § . . + § . . /§ . . review + § . . retest; § is the durability layer beneath those, the agent-side analogue of § . . (same detect→wait/backoff→re-attach/respawn→escalate shape). Composes § . /. /. /. /. /. /. /. /. /. § . /§ . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-147-PROPAGATION` (literal `11.4.147`) + recommended gate `CM-CRASHED-AGENT-RESPAWN-TRACKED` + paired § . meta-test mutation (strip the literal → propagation gate FAILs; mark a `crashed` entry as the loop's done-condition `complete` without landed evidence, OR drop a dispatched agent without a registry entry → `CM-CRASHED-AGENT-RESPAWN-TRACKED` FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-agent-registry`, `--crash-equals-done`, `--no-respawn`, `--discard-partial-state`, `--blind-commit-partial`, `--forget-dead-agent`, `--loop-done-with-crashed-entries` flag.

§ . . — **Workable-item integrity (status+type+id) + comprehensive structured description + bidirectional external-tracker sync + BLOCKED unblock-choices mandate (User mandate,** – – **).** BINDS + STRENGTHENS the workable-item discipline (§ . . status / § . . type / § . . id / § . . Operator-blocked / § . . description-clarity / § . . + § . . SQLite SSoT / § . . docs_chain) into ONE integrity contract spanning DB docs external tracker, adding the operator's three emphases: **(D) no item without a valid status + valid type + stable id — on ALL three surfaces** (an item missing any of the three in the DB, the rendered docs, OR the external tracker FAILs the validator — release-blocker; the id is the cross-surface binding key); **(D) comprehensive structured description per item** — WHAT it is (§ . . ≥ -word/≥ -char clear meaning) + HOW it manifests + HOW to reproduce (§ . . /.) + ACCEPTANCE CRITERIA (§ . . /. captured-evidence verdict), in the § . . DB `description` column AND the docs AND the tracker, never a stub/§ . . -anti-pattern fragment; **(D) BLOCKED items carry WHY + enumerated unblock CHOICES** — tightens § . . : every `Operator-blocked` item (incl. `Blocked` / `BLOCKED` documented alias normalised to canonical `Operator-blocked`, never a silent fork § . .) MUST enumerate the closed list of decisions/actions that would unblock it (`[A]…·[B]…·[C]…`, mirroring § . . 's – -option shape); a BLOCKED item with no enumerated choices FAILs; **(D) regular never-missed bidirectional DB docs tracker sync** — the § . . /. git-tracked SQLite DB is the SSoT, docs + tracker DERIVED; full sync (`db-to-md` / `md-to-db` byte-identical round-trip § . . + tracker push) runs regularly + on every change, § . . drift-proof fingerprint (sha of the sorted item keyset, NOT mtime) gates freshness, § . . docs_chain-bound so drift is mechanically

caught not vigilance-dependent; **(D) generic external-tracker push** carries statuses (collapsed onto the tracker's native set per § . . /. when fixed, precise value preserved in a header, never lost), types, assignee (§ . . participant handle, UNSET defaults from a project env var, never hardcoded/logged § . .), and sub-tasks, IDEMPOTENT (dry-run-then-real, match-by-stable-key `[<ID>]` prefix / id custom-field, present⇒UPDATE/ absent⇒CREATE, rate-limited, credential-redacted, sink-side `created=N updated=M failed=0` proof § . .), the MACHINERY project-agnostic § . . (consumer registers its tracker/list/field-map at runtime). Anti-bluff § . : every sync + validator pass carries captured evidence § . . /. ; honest boundary § . . — guarantees well-formed items + agreeing surfaces, NOT that the underlying work is correct (still crosses § . . /. /.). Composes § . /. /. /. /. /. /. /. /. /. /. /. § . . Classification: universal (§ . .) — consumer supplies its DB path, id prefix, tracker service + list/board id + field map + default-assignee env var, docs_chain context per § . . . Propagation gate `CM-COVENANT-114-148-PROPAGATION` (literal `11.4.148`) + recommended gates `CM-ITEM-INTEGRITY-STATUS-TYPE-ID` / `CM-ITEM-COMPREHENSIVE-DESCRIPTION` / `CM-BLOCKED-UNBLOCK-CHOICES` / `CM-TRACKER-SYNC-IDEMPOTENT` + paired § . mutation (gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--item-without-status`, `--item-without-type`, `--item-without-id`, `--stub-description-OK`, `--blocked-without-choices`, `--skip-tracker-sync`, `--one-way-sync-OK`, `--non-idempotent-tracker-push`, `--hardcode-assignee` flag.

§ . . — Per-workable-item testing-diary mandate (User mandate, - -). Every workable item MUST carry an append-only TESTING DIARY — a chronological record of every test run against it — distinct from § . .

`item_history` (lifecycle STATE transitions) + § . . Reopens (reopen cycles): the diary records TEST EXECUTIONS (which may or may not change status). The mandate (ALL hold): **(a)** a `test_diary` table additive to the § . . /. SQLite SSoT, one row per run soft-keyed to the item id, capturing ISO- -UTC `date_time` + `tested_by` (closed set `User|Operator|AI-agent|HelixQA`) + `result` (§ . . `PASS|FAIL|SKIP` + detail) + in-depth `observations` (long markdown, facts or explicit `UNCONFIRMED:` / `PENDING_FORENSICS:` § . . , the background a future fix needs) + `action_taken` + `status_changed` +from/to (did this run change status + WHY) + § . . `evidence_path` + `feature_class`, with a SCHEMA CONSTRAINT making a `PASS` row WITHOUT a non-empty evidence path impossible (a PASS-bluff rejected by the schema itself); **(b)** BOTH an in-depth diary doc + a derived at-a-glance summary VIEW (total/pass/fail/skip + last-verdict + last-run + status-changes + distinct testers/feature-classes, DERIVED never duplicated § . .) feeding § . . risk-ordering; **(c)** four-format per-item `Diary` / `Diary_Summary` exports § . . + § . . - drift-proof-fingerprinted (sha of the sorted diary keyset) + § . . docs_chain-bound so a diary row not exported/pushed FAILs a gate (never-missed); **(d)** external-tracker SUB-TASK model — the item task's description stays the item (§ . . D , NOT polluted with diary text), each run a CHILD sub-task (type `task`) with a `{TODO|In-progress|Completed}` lifecycle (collapsed per § . . /.), observations + action + an Evidence: line (path not raw artefact § . . /.) + a diary-entry idempotency key, reusing the § . . D idempotent rate-limited credential-redacted sink-side-proven push, mapper project-agnostic § . . ; **(e)** MINIMAL-LLM deterministic bash/Go tooling (zero LLM in the data path — the `observations` prose is authored by whoever ran the test; tooling only stores/renders/pushes/validates), % test-covered § . . (unit + integration-against-the-real-

tracker with an honest § . . SKIP-when-token-absent never a faked PASS + export + HelixQA Challenge + paired § . . mutation) so a diary PASS-bluff is mechanically impossible. Honest boundary § . . — the diary guarantees a complete auditable per-item test-history, NOT that any single run's verdict is correct (rests on its own § . . /. /. evidence). Composes § . . /. /. /. /. /. /. /. /. /. /. § . . Classification: universal (§ . .) — consumer supplies its DB path, diary doc layout, export formats, tracker sub-task field map, docs_chain context per § . . . Propagation gate `CM-COVENANT-114-149-PROPAGATION` (literal `11.4.149`) + recommended gates `CM-TEST-DIARY-SYNC` / `CM-DIARY-PASS-REQUIRES-EVIDENCE` + paired § . mutation (gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-testing-diary` , `--diary-without-evidence` , `--no-diary-summary` , `--diary-without-tracker-subtask` , `--llm-in-diary-data-path` , `--diary-export-optional` flag.

§ . . — Mandatory deep multi-angle web research per change/issue, before declaring fixed or structural (User mandate, - -). Verbatim operator mandate: "For every single issue we fix or improvement we make — besides tight systematic-debugging, fixing, code review by independent agents, comprehensive tests — we MUST ALWAYS do deep web research from various angles! No matter how big/small/simple/complex, dig deep the internet for articles, technical documentation, APIs, open-source code — EVERYTHING that can help make the best possible solution OR confirm we don't have a more serious problem we're unaware of! We MUST do everything possible to STOP the constant issue-reopening and finally start closing items as fixed+working! Do this ALWAYS in parallel with the main work stream!" For EVERY fix / improvement / change / closure — no matter how big, small, simple, or complex — the agent MUST, IN ADDITION to

tight systematic-debugging (§ . .), the fix, independent-agent code review (§ . . / § . .), and comprehensive multi-layer tests (§ . . (b) / § . .), perform a DOCUMENTED **deep multi-angle web research pass** digging the internet from VARIOUS angles — articles, official + vendor technical documentation, API references, standards, issue trackers, maintainer guidance, reusable open-source code — to BOTH (i) discover the best possible solution AND (ii) confirm there is NOT a more serious underlying problem the team is unaware of. Forensic case study (FACT, - -): a subtitle-on-secondary-display goal verdicted `Won't-fix: structurally-impossible` (§ . . secure-surface pixel-blanking) was OVERTURNED by a deep multi-angle research pass that surfaced the real OCR-of-the-PRIMARY-display path — a "we can't" became a shipped capability; the canonical anti-pattern of a structural verdict reached for want of research. The mandate (ALL hold): **(A) No closure-as-fixed/structural WITHOUT a documented deep-research pass** — no item may be marked `Fixed` / `Implemented` / `Completed` (§ . .) OR classified `structurally-impossible` won't-fix (§ . .) until a deep multi-angle pass is documented; § . . makes § . . 's pass UNCONDITIONAL (every item, however trivial, at the closure AND structural-verdict gates). **(B) Multiple angles, not a single lookup** — ≥ genuinely-distinct angles (official-docs / standards / known-bug-trackers / alternative-approach / failure-mode / security / performance / platform-constraint / open-source-precedent) so a single-source confirmation-bias miss (§ . .) cannot pass; a one-link drive-by is NOT deep multi-angle. **(C) Confirm-no-bigger-problem, not just find-a-fix** — explicitly seek evidence the fix does not mask a deeper defect AND the closure/verdict is genuinely safe; "we found nothing worse" requires the enumerated search (§ . .). **(D) Latest-source + cited** — LATEST authoritative versions per § . . (never training-data/memory), each cited by

URL + access date in the research artefact AND the closure commit footer (`Deep-research <date>: <urls>` OR the literal `NO external solution found — original work` per § . .). **(E) Reopen-breaking is the PURPOSE** — STOP the constant Fixed Reopened churn (§ . . / § . .); a reopen whose root cause a deep pass would have surfaced is a § . . miss; composes § . . (demotion-evidence) + § . . (structural verdict now REQUIRES the cited-authorities pass). **(F) ALWAYS in parallel with the main stream** — background subagent-driven (§ . . / § . . / § . . / § . .) concurrent with main fix/build/test work, NEVER serialising it or stalling the loop (§ . . / § . . / § . . / § . .). **(G) Apply ASAP to every workable item** — retroactively + going forward across the § . . / § . . SSoT; items closed without a documented pass are re-audited in the § . . / § . . release-gate sweep. Honest boundary (§ . .): the pass reduces the unknown-unknown surface + breaks the most common reopen causes — it does NOT prove zero remaining defects (§ . .) and does NOT replace § . . runtime-signature verification, § . . / § . . review, or § . . retest; it is the research layer every fix/closure/structural-verdict additionally crosses, and "we probably don't have a bigger problem" without the enumerated multi-angle search is a guess (§ . .), never a finding. Composes § . . / § . . / § . . /

closure-commit-footer convention per § . . . Propagation gate `CM-COVENANT-114-150-PROPAGATION` (literal `11.4.150`) + recommended gate `CM-DEEP-RESEARCH-PER-ISSUE` (every closed/structural-verdicted item carries a documented multi-angle deep-research artefact + cited-source closure footer, run in parallel, before the closure/structural verdict is accepted) + paired § . mutation (strip the literal → propagation gate FAILs; close an item or reach a `structural` verdict with no documented multi-angle research artefact / cited footer → `CM-DEEP-RESEARCH-PER-ISSUE` FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-deep-research`, `--single-source-suffices`, `--trivial-change-no-research`, `--close-without-research`, `--structural-without-research`, `--serialise-research`, `--research-from-memory-OK` flag.

§ . . — Project-prefixed release-tag/version-naming mandate (User mandate, - -). Verbatim operator mandate: "Every release tag and version name we create — on the main repository and on every Submodule we own — MUST be prefixed with the project's release prefix, e.g. `myproject-1.0.0-dev-0.0.1`. The prefix MUST come from `HELIX_RELEASE_PREFIX` in our `.env` if it is set, otherwise from the lowercased project root directory name. The SAME prefix MUST be used across the main repo and all owned Submodules in one release so a release is greppable across every repository." Every release tag AND every version name created on the main repository AND on every owned-by-us submodule (§ . .) MUST be prefixed with the project's release prefix, form `<PREFIX>-<version>` (canonical example: `myproject-1.0.0-dev-0.0.1`), so a release is identifiable + greppable across every repository it spans (one `git tag -l '<PREFIX>-*'` enumerates the whole release surface). **Prefix resolution order (closed-set, deterministic — § . . no-guessing):** ( ) `HELIX_RELEASE_PREFIX` from the project's `.env` — authoritative when set; `.env` is git-ignored per

§ . . and the variable is documented in the tracked `.env.example` (a § . . re-obtain mechanism, never committed); ( ) fallback = the lowercased snake_case form of the project root directory name (no spaces) per § . . — used whenever the env var is unset/empty, so a prefix is ALWAYS resolvable from the checkout with zero operator input. **The prefix MUST be IDENTICAL across the main repo and all owned submodules within a single release** — a release that tags the main repo `<PREFIX>-1.2.0` while tagging an owned submodule with a different/unprefixed value is a § . . violation (the cross-repo grep no longer enumerates the release). Version codes increment monotonically within the prefix (`<PREFIX>-…-0.0.1` → `<PREFIX>-…-0.0.2` → …), never reset, never skipped. Honest boundary (§ . .): the prefix guarantees a release is identifiable + uniform across every repository, NOT that its contents are correct — the tag is still created only after the § . . full-suite retest GREEN and reaches every upstream via the § . . merge-onto-latest-main path (NEVER a force-push), fanned out per § . . Composes § . / § . . / § . . / § . . / § . . (the release-scope terminal condition is a published, prefixed tag). Classification: universal (§ . .) — the consuming project supplies its concrete prefix value + the `HELIX_RELEASE_PREFIX` env var per § . . . Propagation gate `CM-COVENANT-114-151-PROPAGATION` (literal `11.4.151`) + recommended gate `CM-RELEASE-PREFIX-NAMING` (every release tag/version on the main repo + every owned submodule carries the resolved `<PREFIX>-` prefix, identical across the release) + paired § . meta-test mutation (strip the literal → propagation gate FAILs; create an unprefixed or differing-prefix release tag → `CM-RELEASE-PREFIX-NAMING` FAILs; gate-code = separate work item). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is

a release blocker. No escape hatch — no `--no-release-prefix`, `--unprefixed-tag`, `--prefix-optional`, `--differing-submodule-prefix` flag.

§ . . — Crashlytics-recorded-data continuous monitoring + systematic-debug + regression-test-coverage mandate (User mandate, - -). Verbatim operator intent: "For every project that has Firebase Crashlytics enabled / wired, we MUST continuously monitor ALL of the Crashlytics-recorded data — crashes, ANRs, performance traces, and non-fatals — systematically debug each, fix and improve, and cover everything with validation and verification tests. This MUST be checked regularly, with no false results and no bluff of any kind!" Every project that has Firebase Crashlytics enabled/wired (SDK linked into a shipping artifact, crash+non-fatal+ANR reporting active) MUST treat the Crashlytics console as a first-class captured-evidence channel from real end-user devices and continuously process every datum it records — an open Crashlytics issue with a green test suite is a § . PASS-bluff at the field-telemetry layer (a real user hitting a broken feature while the suite reports green). STRENGTHENS+COMPLETES § . . : § . . owns the periodic REVIEW + dedup + Issue-creation pass that SURFACES Crashlytics/Analytics/Performance findings; § . . owns what happens to each surfaced item AFTER — the systematic-debug → fix/improve → regression-test-coverage lifecycle that drives it to a proven regression-immune closure (§ . . finds it; § . . fixes it + proves it stays fixed; both mandatory, neither substitutes). The mandate (ALL hold): ( ) **continuous monitoring of ALL four surfaces** — fatal crashes, ANRs, performance traces/regressions, AND non-fatals (skipping any one is a PASS-bluff; non-fatals are the silent class — every catch/fallback/recovered-error path is a real degraded UX and MUST be triaged, not ignored because the app did not crash); ( ) **systematic-debugging of each (reproduce-before-fix)** per § . . Iron Law + § . .

RED-baseline-on-the-broken-artifact (the recorded stacktrace/ANR-trace/non-fatal-context IS the § . . /. captured defect-present evidence) BEFORE the fix — a fix without a prior reproducing test is a § . . /. violation; unreproducible ⇒ deep-research-before-declaring-untestable § . . /. ; ( ) **a fix/improvement for every confirmed issue** against its proven root cause (§ . . /. /.), "acknowledged in console" is not a fix, muting a recurring issue without a tracked rationale is a § . . /. violation; ( ) **validation+verification regression-test coverage per closed issue** — in the SAME commit as the fix register a permanent § . . regression guard (§ . . polarity test: `RED_MODE=1` captures the recorded defect on the pre-fix artifact, `RED_MODE=0` standing GREEN guard asserting ABSENT) exercising the same user-reachable path the stacktrace identifies, with rock-solid captured evidence § . . /. /. /. + a closure log citing the console issue id/URL + root-cause + fix commit + the validation/verification test paths; a Crashlytics issue marked resolved WITHOUT a falsifiable regression test is FORBIDDEN (the silent-recurrence vector, § . . operator-escape class applied to field telemetry); ( ) **regular cadence** reusing the § . . five-trigger set (pre-build/pre-flash/pre-distribute/pre-tag blocking; daily + post-deployment-burn-in non-blocking) — a one-time sweep never repeated is a violation; ( ) **no false results, no bluff** (§ . /. /. — "no new issues" requires the enumerated monitored-surfaces+window § . . ; "fixed" requires the RED→GREEN flip § . . /. ; a guard whose § . mutation does not FAIL it is a bluff gate). Honest boundary § . . : processing every recorded issue breaks the silent-recurrence vector — it does NOT prove zero remaining field defects nor replace § . . /. ; an issue is "closed" only when its guard is GREEN on a clean artifact, never when the console mark is flipped.

Classification: universal (§ 11.4.10.1.1.1

(§ . .) — with NO single feature left out. ( ) **Total categorized coverage** — per-feature table organized Component→Category, reconciled against the actual codebase (a code-present feature missing from the table, or a row with no code, is a § . . /§ . . bluff). ( ) **Per-feature fields** — Component / Feature / Category / Implementation / Wiring (genuinely end-user-reachable, not merely compiled, § . .) / Real-use / Tests-coverage (four-layer § . . (b)) / Validation (PASS / FAIL / SKIP / PENDING_FORENSICS / OPERATOR-BLOCKED per § . .) / **Video-recording confirmation** (path to the real-use video or honest gap marker). ( ) **Mandatory per-feature real-use video** — every user-visible "confirmed" claim backed by a recorded real-use video of a genuine end-user scenario (real prompts → real LLM/service responses → real results), NEVER a frozen/stale frame (§ . .), NEVER faked/mocked/demo-loop/bluff response or LLM error passed off as success (§ . . /§ . .), stored at the project-declared recording path (§ . .); a confirmed row with no real video or a bluff video = § . PASS-bluff; autonomous-infeasible ⇒ honest § . . /§ . . SKIP + tracked migration item, NEVER a faked confirmation. ( ) **Video-analysis remediation loop** — every video analysed; any defect it surfaces triggers § . . systematic-debug → fix → § . . retest → re-record → clean GO per § . . before "confirmed" (a video exposing a broken feature is a § . . test-interrupt). ( ) **Always-in-sync** — § . . -class roster/corpus-backed, § . . docs_chain-bound + § . . drift-proof fingerprint (sha of sorted feature-key roster AND sorted video-artefact roster, NOT mtime), re-syncs out-of-the-box; stale = violation. ( ) **Four-format export** — HTML + PDF + **DOCX** (this doc class ADDS DOCX to the § . . HTML+PDF set; other classes unchanged), in sync per § . . . ( ) Follows § . . /. /. /. /. /. . ( ) **MP**

format REQUIRED. All video confirmations MUST be in `.mp4` format (H.). Window-specific capture ONLY (§ . . (A)). Vision validation REQUIRED (§ . . (D)). `.cast` files are supplementary only. Honest boundary § . . — guarantees a complete video-confirmed always-synced ledger, NOT per-feature correctness (rests on each row's § . . /. /. evidence) and NOT a § . . retest substitute. Classification: universal (§ . .). Composes § . . /. /. /. /. /. /. /. /. /. /. /. § . . Propagation gate `CM-COVENANT-114-153-PROPAGATION` (literal `11.4.153`) + recommended gates `CM-FEATURE-STATUS-COMPLETE` + `CM-FEATURE-STATUS-VIDEO-CONFIRMED` + paired § . mutation.

Canonical authority: constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-feature-status`, `--feature-without-video`, `--frozen-video-OK`, `--bluff-response-video-OK`, `--skip-video-analysis`, `--feature-ledger-incomplete-OK`, `--no-docx-export`, `--allow-feature-ledger-drift` flag.

§ . . — Window-scoped capture + fresh-corpus rotation for feature/QA recordings (User mandate, – –). Verbatim: "recording you perform does record the window containing apps and services, not the whole desktop or monitor screen!" + "All old recording files MUST BE removed when new one starts!" Refines § . . /. /. /. recording discipline with two capture-hygiene invariants. **(A) Window-scoped, NOT whole-screen** — every feature/QA video MUST capture ONLY the window/surface of the app/service under test (GUI window / CLI-TUI terminal pane / web tab-viewport / device-emulator-simulator frame), NEVER the whole desktop/monitor or unrelated windows; whole-desktop capture leaks operator-private content (§ . . /.), dilutes the § . . liveness/freeze oracle, and breaks the § . . OCR/

ROI content oracle. Target the window/region by stable identity (window id/title, device serial, browser context, tmux target) per § . . — never a fixed full-screen device index. Platform genuinely cannot capture below whole-screen ⇒ honest § . . SKIP + tracked migration item, never a whole-screen pass-off. **(B) Fresh-corpus rotation** — when a new recording run for a scope begins, the agent's OWN prior in-scope stale recordings at the raw recording path MUST be removed FIRST so the live corpus reflects the current run (§ . . not-stale + § . . roster-freshness). Honest boundary (§ . . + § .): "remove old" = the agent's own prior recordings for the SAME scope/project ONLY — NEVER another project's/scope's/operator-authored files; uncertain ⇒ surface, don't delete (§ . .); committed `docs/qa/<run-id>/` evidence (§ . .) is the durable record, NOT rotated. Classification: universal (§ . .). Composes § . . /. /. /. /. /. /. /. /. /. § . /. . Propagation gate `CM-COVENANT-114-154-PROPAGATION` (literal `11.4.154`) + recommended gate `CM-WINDOW-SCOPED-FRESH-CORPUS-RECORDING` + paired § . mutation.

Canonical authority: constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--whole-screen-capture-OK`, `--skip-window-scope`, `--keep-stale-recordings`, `--no-corpus-rotation`, `--full-desktop-recording` flag. **(C) MP auto-conversion REQUIRED.** Any `.cast` file produced MUST be auto-converted to `.mp4` via `agg` + `ffmpeg` immediately after capture. The `.mp4` is the primary evidence; `.cast` is supplementary only.

§ . . — Project-name-prefixed feature/QA recording filenames (User mandate, - -). Verbatim: "All recorded videos MUST START with prefix: the PROJECT NAME (ALWAYS USE THE PROJECT NAME). Project name MUST be obtained according to the constitution's own project-name resolution." Every recorded video the project produces — every feature/QA

real-use recording (§ . .), every window-scoped capture (§ . .), every always-on device recording (§ . .), every raw or curated artefact at the project-declared recording path (§ . .) + the committed `docs/qa/<run-id>/` trail (§ . .) — MUST have a filename that STARTS WITH the PROJECT-NAME prefix, ALWAYS; an unprefixed recording is a § . . violation (a multi-project/scope corpus on one host per § . . /. becomes un-greppable + un-attributable — the § . . identify-and-grep failure on the recording-corpus axis). **Prefix resolution (closed-set, deterministic — § . . , IDENTICAL to § . .):** ( ) `HELIX_RELEASE_PREFIX` from `.env` (authoritative, git-ignored § . . , documented in tracked `.env.example` § . .) else ( ) lowercased snake_case project-root dir name § . . — ALWAYS resolvable, zero operator input. SAME prefix for EVERY recording in a checkout so `ls '<PREFIX>---'*` enumerates the corpus; canonical form `<PREFIX>---<feature-or-scope>---<run-id>.<ext>` (`---` delimits the prefix unambiguously); MUST equal the § . . -resolved release-tag prefix (divergence is itself a § . . violation — one project, one name). Honest boundary (§ . .): the prefix guarantees attribution + greppability, NOT content validity (still § . . /. /.) and does NOT relax § . . 's window-scope/rotation (rotation removes the agent's OWN `<PREFIX>---*` only; foreign/operator files surfaced not deleted § . . /§ .). Classification: universal (§ . .). Composes § . . /. /. /. /. /. /. /. /. /. § . . Propagation gate `CM-COVENANT-114-155-PROPAGATION` (literal `11.4.155`) + recommended gate `CM-RECORDING-PROJECT-NAME-PREFIX` + paired § . mutation.

Canonical authority: constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--no-recording-prefix`, `--recording-without-project-name`, `--unprefixed-recording`, `--prefix-optional-for-recording`, `--differing-recording-prefix` flag.

§ . . — All CI/CD automation (GitHub Actions / GitLab pipelines / equivalents) MUST be disabled (User mandate, - -). Verbatim operator mandate: "Any GitHub actions or GitLab pipelines MUST BE disabled! Add this critical mandatory rule / mandatory constraint into the root constitution Submodule, commit and fetch all its changes to all upstreams and make sure we respect and follow this rule (we do apply it) ASAP!!!" Every repository this Constitution governs — main repo, this constitution submodule, every owned + nested submodule we author and push — MUST ship with ALL server-side CI/CD automation DISABLED: no push to any owned upstream may trigger a GitHub Actions run, GitLab pipeline, or equivalent (Jenkins/CircleCI/Travis/Drone/Woodpecker/Bitbucket/Azure, any `on: push` / `schedule` / `workflow_dispatch`). GENERALISES + makes ABSOLUTE the § . . Layer- posture (remote CI DISABLED, workflow preserved at a `…disabled-local-only` non-`.yml` name a provider ignores) across ALL governed repos; enforcement migrates to the LOCAL § . . git-hook ritual + § . . pre-tag sweep, never a remote runner. ALL hold: **(A)** zero active `.github/workflows/*.yml|yaml` / `.gitlab-ci.yml` / `.gitlab/**` / equivalent at the ROOT of any governed repo/submodule (the only place a provider executes); **(B)** "disabled" = a push triggers ZERO runs — delete OR rename to a non-trigger name (§ . . `.disabled` / `.disabled-local-only`); a live-`on:` + `if:false` workflow still queues runs, NOT compliant; **(C)** scope = repos we author+push — vendored/third-party nested configs below the root (AOSP `external/**`, `prebuilts/**`, vendored submodules) are INERT (a provider never runs a non-root config), OUT of scope, MUST NOT be mass-edited (§ . . vendor-exempt); test = "does a

push to OUR upstream trigger a run?" yes⇒disable, inert⇒document+leave (§ . . verify-not-assume); **(D)** no new CI may be added (release blocker); **(E)** pre-push verify `git ls-files | grep -E '^\.github/workflows/.*\.ya?ml$|^\.gitlab-ci\.yml$'` empty for authored repos, § . . -class PreToolUse guard + gate enforce mechanically. Honest boundary (§ . .): file-level disabling stops FILE-triggered runs, NOT provider-side server settings (org-default required workflows, branch-protection required checks, provider scheduled exports) — the operator turns those off; the agent documents what it cannot reach, never claims unachieved completeness. Composes § . . /. /. /. /. /. /. /§ . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-156-PROPAGATION` (literal `11.4.156`) + recommended gate `CM-NO-ACTIVE-CI` + paired § . meta-test mutation (strip the literal → propagation gate FAILs; add a root `.github/workflows/x.yml` to an authored repo → `CM-NO-ACTIVE-CI` FAILs). **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--allow-ci`, `--enable-workflow`, `--keep-pipeline`, `--remote-ci-OK`, `--ci-exempt` flag.

§ . . — GEMINI.md maintained in lockstep with CLAUDE.md / AGENTS.md / QWEN.md (User mandate, - -). Verbatim operator mandate: "Make sure with CLAUDE.md, AGENTS.md, QWEN.md we maintain GEMINI.md too! Add this mandatory fact / rule to root constitution Submodule we are inheriting / extending - CONSITUTION.md, CLAUDE.md, QWEN.md, AGENTS.md, GEMINI.md and other related relevant files!" Forensic FACT (- -): when § . . /§ . . were authored, Constitution/CLAUDE/AGENTS/QWEN carried the family through § . . but GEMINI.md had silently drifted to § . . — mandates (§ . . -) never propagated — a § . propagation-bluff (a Gemini-CLI agent reads a stale Constitution). GEMINI.md is a FIRST-

CLASS governance context carrier EQUAL to CLAUDE.md/AGENTS.md/QWEN.md, never optional/best-effort. ALL hold: **(A)** five-carrier lockstep — no governance change is complete until GEMINI.md carries it alongside the other three mirrors; GEMINI.md is added to the § . . propagation + cross-reference set explicitly; **(B)** no silent drift — GEMINI.md lagging the other mirrors' highest rule is a § . . violation (§ . . -class), back-fill required; **(C)** equal status — GEMINI.md restates the SAME literal `11.4.N` anchors the propagation gates require (§ . .), fleet count INCLUDES GEMINI.md; **(D)** consumer projects' own CLAUDE/AGENTS/QWEN/GEMINI bind too (§ . .). Honest boundary (§ . . .): the § . . - GEMINI.md back-fill is a tracked release-blocking remediation; claiming GEMINI.md "in sync" while the back-fill is incomplete is itself a § . . violation. Composes § . . /. /. /. /. /. /. / § . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-157-PROPAGATION` (literal `11.4.157`, GEMINI.md INCLUDED) + recommended gate `CM-GEMINI-MD-LOCKSTEP` + paired § . meta-test mutation. **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-gemini-md`, `--gemini-optional`, `--gemini-lag-OK`, `--four-carrier-suffices` flag.

§ . . — Intensive all-feature/flow/edge-case video-recording + read-the-screen content-verification mandate (User mandate, - -). Every project MUST be covered by intensive automated testing that exercises + RECORDS every feature/flow/use-case/edge-case (valid/invalid/boundary/concurrent/failure-injection § . .), each recording showing the feature GENUINELY WORKING with REAL results (real prompts→real responses→real outputs § . .) and NO false/simulated/stale/frozen result (§ . . /. /.) — a recording showing an error/bluff/non-working feature is a FINDING (→§ . . ( )/

§ . . fix→retest→re-record), never a confirmation. The testing System MUST ACTUALLY READ every shown log line / message / UI label / dialog / toast / status text and VERIFY it is a genuine working result (OCR/ROI § . . /. + confidence floor for pixel surfaces; direct capture for terminal/log; § . . ( ) self-validated golden-good/golden-bad analyzer) — "a video was produced" is NOT evidence, "the System read the screen + confirmed a genuine result" is. HelixQA (§ . .) MUST drive this exercise→record→read→score pass, PASS only on a read-confirmed genuine result + captured artefact path (§ . .). Default recording save path = `$HOME/Downloads` (host user's home Downloads, resolved at runtime never hardcoded) unless a project declares an override per § . . ; § . . project-prefix + § . . window-scope/rotation + § . . git-ignored raw corpus + § . . curated docs/qa evidence apply. **Vision analysis MANDATORY for every recording** — after every recording, the agent MUST read the terminal output / video content and verify the feature actually works. A recording without a vision-confirmed verdict is a § . . violation. Composes § . . /. /. /. /. /. /. /. /. /. /. § . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-158-PROPAGATION` (literal `11.4.158`) + recommended gates `CM-INTENSIVE-RECORDING-COVERAGE` + `CM-RECORDING-CONTENT-READ-VERIFIED` + paired § . mutation. **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--skip-recording-coverage`, `--video-without-content-read`, `--happy-path-recording-suffices`, `--recording-path-anywhere`, `--unread-recording-OK`, `--skip-helixqa-read` flag.

§ . . — Mandatory window-specific video recording + vision validation mandate (User mandate, - -). Every feature test, validation,

verification, challenge, and QA session that produces video evidence MUST comply with ALL of the following: **(A) Window-specific recording ONLY** — every video MUST record ONLY the target application window (Terminal pane, TUI app, browser tab, emulator frame), NEVER the whole desktop/monitor screen; use macOS `screencapture -l<window_id>`, Linux `xdotool` + `ffmpeg`, or equivalent; whole-desktop capture leaks operator-private content (§ . .), dilutes the § . . liveness oracle, and breaks § . . OCR/ROI; platform genuinely cannot capture below whole-screen => honest § . . SKIP + tracked migration item. **(B) MP format REQUIRED** — `.mp4` (H. , `movflags +faststart`, `pix_fmt yuv420p`); `.cast` files are supplementary only; auto-convert via `agg` + `ffmpeg` per § . . (C). **(C) Project-name prefix REQUIRED** — filename MUST start with project name in snake_case from `HELIX_RELEASE_PREFIX` in `.env` (§ . .) or lowercased project root dir name (§ . .); format `<project_name>-<feature>-<timestamp>.mp4`; unprefixed = violation. **(D) Mandatory vision validation** — after EVERY recording, the agent MUST read the terminal output / video content and verify: (i) feature ACTUALLY WORKS, (ii) LLM responses are REAL, (iii) all tests show PASS, (iv) no "TODO implement" / "simulate" / "for now" patterns, (v) output demonstrates end-user working feature; MUST produce a verdict (PASS/FAIL) with evidence path (§ . .). **(E) Terminal window cleanup** — after each recording, dismiss/close ONLY the Terminal window used for that recording (window-specific close via `osascript` / `xdotool`); MUST NOT close windows belonging to other processes. **(F) Real results ONLY** — every recording MUST show REAL working features; errors/empty output/simulated responses trigger fix→retest→re-record. **(G) Re-runnable evidence** — the command shown MUST be re-runnable to produce the same results. **(H) Fresh-corpus rotation** — remove agent's own prior in-scope stale recordings FIRST (§ . .). **(I) Content verification MANDATORY — not duration-based** — the

value of a recording is NOT its duration but its CONTENT; a recording MUST demonstrate the ACTUAL feature being used with REAL results; before accepting ANY recording, the agent MUST verify: expected output patterns ARE present (e.g., test PASS lines, API response data, feature-specific output), the feature ACTUALLY WORKS as demonstrated (not just "something ran"), LLM responses are REAL content (not simulated, not placeholder, not empty), every claim of "working" is backed by visible evidence; a -second recording showing a feature working correctly is MORE valuable than a -second recording of empty terminal; duration is NOT a proxy for quality. **(J) Expected-content specification REQUIRED** — before recording, the agent MUST specify what content SHOULD appear (expected patterns, expected test results, expected API responses); after recording, the agent MUST verify these patterns ARE present; if expected content is MISSING, the recording is REJECTED regardless of duration. **(K) Content-verification recording workflow** — mandated workflow: ( ) SPECIFY expected content patterns, ( ) RECORD the feature execution, ( ) EXTRACT all text from the recording, ( ) VERIFY expected patterns are present, ( ) CHECK for simulated/placeholder content, ( ) ACCEPT only if ALL patterns found AND zero bluffs detected, ( ) REJECT and re-record if ANY pattern missing or bluff detected. **(L) Root cause analysis REQUIRED for rejected recordings** — when a recording is rejected (missing expected content, bluff detected, or empty capture), the agent MUST investigate WHY before re-recording per § . . ; determine the root cause (timing issue, wrong command, tool failure, etc.) and fix it; simply re-recording without understanding WHY the first attempt failed is a § . . violation. **(M) Real-time monitoring RECOMMENDED** — for complex features, use real-time monitoring that analyzes output DURING recording (not after); this catches issues immediately and allows corrective action before the recording completes. Classification: universal (§ . .). Composes

§ . . /. /. /. /. /. /. /. /. /. /.
§ . . . Propagation gate `CM-COVENANT-114-159-PROPAGATION` (literal `11.4.159`) + recommended gate `CM-WINDOW-VIDEO-VALIDATED` + paired § . mutation. **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--whole-screen-ok`, `--cast-only`, `--skip-vision-validation`, `--no-cleanup`, `--simulated-recording-ok`, `--unprefixed-recording` flag.

§ . . — Vision-verified recording + HelixQA bridge mandate (User mandate, - -). Compact summary: every video recording for feature/QA evidence MUST be processed through a vision/OCR pipeline that reads on-screen content and confirms expected results BEFORE acceptance; the recording system MUST provide a bridge feeding captured frames to HelixQA's test infrastructure (or equivalent) for automated read-the-screen verification against SPECIFY-phase expected patterns (§ . . (J)); capture frames at ≤ s intervals, run OCR/vision with self-validated analyzer (§ . . ( )), compare text against patterns, produce per-frame PASS/FAIL with evidence path, surface failures immediately for re-record (§ . . (L)). Project-configured parameters per § . . . Honest boundary: does NOT replace § . . quality analysis nor § . . runtime-signature; FLAG_SECURE uses § . . proxy oracle. Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-160-PROPAGATION` + recommended gate `CM-VISION-VERIFIED-RECORDING-BRIDGE` + paired § . mutation.

§ . . — Rootless container runtime mandate (User mandate, - -). Compact summary: every project MUST use Podman rootless (or equivalent) for ALL containerized workloads — rootful Docker/sudo FORBIDDEN unless § . . platform constraint documented; `vasic-digital/containers` Submodule (§ . .) sole

orchestration layer via `pkg/boot` / `pkg/compose` / `pkg/health`; extend upstream per § . . for missing capabilities; integration tests boot infra on-demand. Honest boundary: does NOT replace § . . credentials nor § . hygiene. Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-161-PROPAGATION` + recommended gate `CM-ROOTLESS-CONTAINER-RUNTIME` + paired § . mutation.

§ . . — OpenDesign UI design system mandate (User mandate, - -). Compact summary: every project with user-facing interfaces MUST use OpenDesign (https://github.com/nexu-io/open-design) as mandatory UI design system — NOT ad-hoc CSS; install as dependency, use token system for all color (light+dark from brand assets), typography, spacing, component tokens; extend upstream (§ . .) for unsupported patterns; every UI component ships light+dark + token-diff regression tests via visual regression; no element overlap/font collision/label overlay. Honest boundary: does NOT replace § . . functional testing nor § . . /. UI testing. Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-162-PROPAGATION` + recommended gate `CM-OPENDESIGN-UI-SYSTEM` + paired § . mutation.

§ . . — Universal Media Validation (User mandate). Every recorded artifact MUST pass MEDIA VALIDATION pipeline — OCR/transcription/text parsing, compare vs SPECIFY-phase patterns (§ . . (J)), self-validated analyzer (§ . . ( )), structured verdict, pinpoint on FAIL, REAL-TIME trigger option. Paired § . mutation on golden-bad fixture. Classification: universal. Propagation gate `CM-COVENANT-114-163-PROPAGATION` + paired § . mutation.

§ . . — Universal Auto-Propagation Hook (User mandate). Every constitutional pull triggers `post_update_hook.sh` — detects changed files, registers skills/

MCP/hooks/scripts, logs summary. Consumer invokes after every pull. Classification: universal. Propagation gate `CM-COVENANT-114-164-PROPAGATION` + paired § . mutation.

§ . . — **Universal Independent Verification Agent (User mandate).** Every code/media/docs/config output passes independent verifier (§ . . /.), zero-finding GO (§ . .). CODE: review+build+mutations+runtime-signature. MEDIA: pipeline+genuine+format. DOCS: exports+revision. CONFIG: syntax+schema+leaks. Self-validated § . mutation. Classification: universal. Propagation gate `CM-COVENANT-114-165-PROPAGATION` + paired § . mutation.

§ . . — **REPEALED (operator decision, - -).** The Universal Semgrep static-analysis mandate is repealed — Semgrep is NO LONGER mandatory. Static analysis remains encouraged, not mandated. Scaffolding, `submodules/semgrep`, MCP/pre-commit/PATH wiring, docs_chain `semgrep_status` context, and the `CM-COVENANT-114-166-PROPAGATION` / `CM-SEMGREP-WIRED` gates removed. Anchor . . retired. See constitution `Constitution.md` § . . .

§ . . — **Big-work-item feature work-stream lifecycle (User mandate, - -).** Every BIG work item (new feature OR drastic/large fix) MUST be its own isolated feature work-stream — own sibling project copy, own `feature/<slug>` branch, own per-feature flashable builds + feature tags, SEPARATE from trunk until operator manually approves after testing, trunk merged INTO every stream regularly. (A) one big item = one sibling project copy (`<project>_<slug>`); small changes stay on trunk. (B) disk-feasible CoW/reflink clone MANDATORY (`cp -a --reflink=auto` on btrfs/XFS/ZFS/APFS, else `git worktree`); naive deep copies FORBIDDEN; per-stream `out/` + shared ccache; ~ -disk verified (§ . .) before relied on (§ . .). (C) own branch + greppable tags (§ . . `<base>-feat-<slug>`), NEVER merged to trunk until § . . GREEN on

clean target → § . . merge-first → § . . ff-only. (D) trunk merged INTO every live stream per trunk-tag/daily (merge, never rebase). (E) feature branch + tag cascades to EVERY touched submodule (§ . .). (F) single-builder build QUEUE (§ . . /. . /. .) + finite-device queue (§ . .); idle `out/` GC'd (§ . .). (G) every change crosses full gauntlet (§ . . /. /. /. + anti-bluff). (H) stream registry (§ . . /.) + atomic `.partial`→rename + § . . resume + § . retire-backup; bounded § /. . . Honest boundary (§ . .): isolation+disk-feasibility+greppable-identity+merge-discipline, NOT correctness (still § . . /.). Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-167-PROPAGATION` (literal `11.4.167`) + gates `CM-FEATURE-WORKSTREAM-COW-CLONE` / `-NO-MERGE-UNTIL-APPROVED` / `-TRUNK-SYNC-CADENCE` / `-SUBMODULE-CASCADE` + paired § . mutations. **Canonical authority:** constitution `Constitution.md` § . . .

§ . . — Exported-document independent content + textual + full-visual validation mandate (User mandate, - -). Every generated/exported document (HTML/PDF/DOCX/any § . . export-scope format) MUST pass INDEPENDENT validation by a reviewer structurally separate from the generator (§ . . /. , NEVER author self-check), on BOTH source AND every exported artifact, ALWAYS, across THREE layers: ( ) CONTENT — export faithfully carries source intent+data, nothing dropped/truncated/garbled; ( ) TEXTUAL — human-readable, NO raw markup/diagram-source (Mermaid `gantt` / `graph` / `flowchart` / `sequenceDiagram` etc.)/ unrendered code-fence leaking as body text (the exact - - raw-gantt-in-PDF failure); ( ) FULL VISUAL — diagrams render as IMAGES not source, layout intact, no overlap/cut-off/garble, verified by RENDERING the export (`pdftotext` catches raw source + `pdfimages` confirms rendered images + `pdftoppm`→OCR confirms human-readable content) with

captured evidence § . . /. /. . Anti-bluff: rubber-stamp ≠ validation; analyzer self-validated golden-good/golden-bad § . . ( ) (golden-bad seeded with raw `gantt` MUST FAIL); iterate to clean GO § . . . Forensic FACT (- -): a "green" Mermaid fix shipped raw gantt source as readable text into user PDFs because validation grepped HTML for `class="mermaid"` and NEVER checked the exported PDF — operator caught it (§ . .). Honest boundary (§ . .): confirms the exported artifact is faithful/readable/visually-rendered — does NOT prove source content is correct (§ . . /.) nor replace the § . . mtime freshness gate (this is the READABILITY/FIDELITY layer the mtime gate cannot see); FLAG_SECURE/un-rasterisable surfaces document the gap § . . + § . . proxy, never a faked PASS. Classification: universal (§ . .). Composes § . . /. /. /. /. /. /. /. /. § . . Propagation gate `CM-COVENANT-114-168-PROPAGATION` (literal `11.4.168`) + recommended gate `CM-EXPORTED-DOC-VISUALLY-VALIDATED` + paired § . mutation. **Canonical authority:** constitution `Constitution.md` § . . .

§ . . — Mandatory comprehensive test-type coverage with anti-bluff captured evidence (User mandate, - -). Every project MUST cover a CLOSED enumerated test-type set, each to as-close-to- % as the domain permits, every PASS citing rock-solid captured PHYSICAL evidence (§ . . /. /.), zero false results/zero bluff (§ . /. /.): ( ) unit (only layer mocks allowed § . . (A)); ( ) integration (real System, infra via containers submodule § . .); ( ) e e; ( ) full-automation (autonomous, re-runnable, no manual step § . . /. /. , deterministic § . .); ( ) Challenges (challenges submodule banks § . . (B)); ( ) HelixQA (helix_qa submodule — written test banks/suites per app/service/platform + comprehensive

fully-autonomous QA sessions § . .); ( ) DDoS/load-flood (clean refuse or graceful degrade); ( ) security (authz/injection/secret-leak § . . /crypto/CVE/default-deny + security submodule); ( ) stress+chaos (load+contention+boundary+failure-injection, categorised recovery § . .); ( ) concurrency/atomicity (no lost updates, transactional atomicity, idempotent replay); ( ) race-condition/deadlock (race detector + lock-order analysis; no blocking op in shared-lock region §); ( ) memory (leak census + peak-RSS ceiling + h/N-iter soak, no unbounded growth); ( ) benchmarking/performance (p / p /p + throughput vs recorded baseline). The ONLY permitted absence is an honest § . . SKIP-with-reason, never a silent gap; four-layer § . . (b) enforcement + coverage ledger (§ . . /.); missing required type or OPERATOR_ATTENDED_ONLY = release blocker. Strict enumerated expansion of § . . . Composes § . . /. /. /. /. /. /. /. /. /. /. /. /. § . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-169-PROPAGATION` (literal `11.4.169`) + recommended gate `CM-MANDATORY-TEST-TYPES-COVERED` + paired § . mutation. **Canonical authority:** constitution `Constitution.md` § . . .

§ . . — Device-independent host-side rendered-UI visual-proof mandate (User mandate, - -). Compact summary: every change to ANY user-facing UI surface (screen/component/view/widget/layout/theme on any platform — Android-Compose, iOS-SwiftUI/UIKit, web, desktop, TUI) MUST be proven by DEVICE-INDEPENDENT host-side RENDERED PIXELS before it may be claimed correct — the real component rendered to a PNG ON THE HOST (NO device, NO emulator, NO running app) via a host-render harness (Compose→Roborazzi/Paparazzi on the JVM; web→Playwright/Storybook snapshot; SwiftUI→snapshot-testing; equivalent per stack), for EVERY relevant screen×state×{light,dark} theme, with the PNG validated by BOTH (i) golden image-diff (per-pixel/perceptual

PASS/FAIL) AND (ii) an OCR/vision oracle reading rendered text+labels+control bounds to assert legibility + NO overlap / NO label-over-label / NO clipping / NO off-screen control / NO collapsed-or-giant-unbounded widget (the § . . layout invariants PROVEN on real pixels). Forensic FACT (- -): UI work was "validated" by VALUE-EQUALITY unit tests (a hex colour string / an sp/dp size integer / a design-token field == a constant) that NEVER RENDERED A PIXEL — GREEN while the operator opened a broken unbounded giant-button screen. A value/token-equality / property-assertion UI test is FORBIDDEN as the PROOF a UI is correct (it may SUPPLEMENT, NEVER SUBSTITUTE the rendered-pixel proof); a "green" UI suite with no rendered-pixel artefact for the changed surface is a § . PASS-bluff at the visual layer. Self-validated golden-good/golden-bad analyzer § . . ( ), thresholds calibrated on the project's own fixtures § . . . "device offline" is NEVER a valid skip — host-render IS the device-independent path; honest § . . SKIP only where the platform genuinely has no host-render harness (tracked migration item, never a value-only fallback). Honest boundary § . . : host-render proves the COMPONENT renders correctly device-independently per commit — it does NOT prove the running app on real hardware (that remains § . . /. /. /. 's live-device recording + read-the-screen layer, which § . . COMPLEMENTS not replaces — both mandatory), does NOT replace § . . OpenDesign token/theme governance (§ . . PROVES the rendered RESULT of those tokens), DISTINCT from § . . exported-document visual validation. Classification: universal (§ . .). Composes § . . /. /. /. /. /. /. /. /. /. § . . Propagation gate `CM-COVENANT-114-170-PROPAGATION` (literal `11.4.170`) + recommended gate `CM-HOST-RENDERED-UI-VISUAL-PROOF` + paired § . mutation. **Canonical authority:**

constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--value-assertion-proves-ui`, `--skip-rendered-pixel-proof`, `--token-equality-suffices`, `--device-offline-skip-visual`, `--single-theme-only`, `--unvalidated-image-diff-OK`, `--no-ocr-layout-oracle` flag.

When in doubt

- Read `constitution/Constitution.md` for the canonical text of every rule.
- Cross-reference the sibling carriers (`CLAUDE.md` / `AGENTS.md` / `QWEN.md`) — they carry the same universal content for their respective agents.
- If still unclear, ask the operator. Do NOT guess. Do NOT bluff (§ . .).

§ . . — Mandatory comprehensive human-readable workable-item descriptions (User mandate, - -). Every workable item (ATM-NNN ticket) in the project's workable-items database, Issues.md, Fixed.md, Issues_Summary.md, Fixed_Summary.md, and ALL derived documents MUST carry a comprehensive human-readable description that explains the item in plain, non-technical language suitable for non-developers (project managers, stakeholders, team leads, business partners). The description MUST be - sentences minimum, written so that ANY person — regardless of technical background — can understand: (a) WHAT the item is (the feature, fix, or task in simple terms); (b) WHY it matters (the user/business value); (c) HOW it works (the mechanism, without code references); (d) WHO benefits (the end user or stakeholder); (e) WHAT the expected outcome is (the measurable result). The description is MANDATORY — an item without a human-readable description is a § . violation at the documentation layer. Descriptions MUST NOT use jargon, acronyms without expansion, code

references, or technical implementation details as the primary explanation. Technical details MAY appear as supplementary context AFTER the plain-language explanation. The `description` column in the SQLite database (`docs/workable_items.db`) is the SINGLE SOURCE OF TRUTH for item descriptions; all derived documents (Issues.md, Fixed.md, summaries, exports) MUST carry the SAME description text. Pre-build gate `CM-HUMAN-READABLE-DESCRIPTION` (when implemented) walks every item in the DB and asserts the `description` field exists, is non-empty, and is ≥ characters. Paired § meta-test mutation strips a description → gate FAILs. Classification: universal (§). Composes § (item-status tracking), § (item-type tracking), § (column alignment), § (summary clarity), § (SQLite SSoT), § (item integrity), § (universal export). Propagation gate `CM-COVENANT-114-171-PROPAGATION` (literal `11.4.171`) + paired § mutation.

§ — Mandatory production-readiness planning with realistic timeline projection (User mandate, - -). Every project under this Constitution MUST maintain a living production-readiness planning document that: (a) provides REALISTIC timeline projections for all product milestones (MVP, beta, production-ready) based on actual development velocity data (items completed per week, reopening rate, blocking dependencies); (b) identifies ALL danger zones, weaknesses, and risk areas with mitigation strategies; (c) accounts for HW delivery timelines, licensing delays, and external dependency risks; (d) defines the critical path and identifies which items can slip without affecting MVP; (e) includes workstation/build-capacity analysis and optimization recommendations; (f) is updated at least monthly or whenever the item count changes by ≥ %; (g) is cross-referenced against the workable-items database for accuracy. The planning document MUST live at `docs/research/production_planning_<YYYYMMDD>/ANALYSIS.md` with

HTML+PDF exports per § . . . Honest boundary (§ . .): projections are estimates based on measured velocity — they guarantee the methodology is sound, NOT that specific dates will be met. Classification: universal (§ . .). Composes § . . (no-guessing — projections based on data, not wishful thinking), § . . (full retest before tag), § . . (iteration discipline), § . . (SQLite SSoT), § . . (four-layer verification). Propagation gate `CM-COVENANT-114-172-PROPAGATION` (literal `11.4.172`) + paired § . mutation.

§ . . — Containerized + distributed build mandate (User mandate, - -). EVERY build of EVERY component (source compile, artifact/package/installer/container-image production, codegen, asset render — for ANY language/platform: Go, Android/Gradle, desktop, web, native, firmware) MUST run INSIDE a specialized build container provisioned via the `digital.vasic.containers` submodule (§ . .) — NEVER on the bare host. The build containers MUST be DISTRIBUTED to the designated remote build host(s) (e.g. `thinker.local`) via the SAME containers-submodule distribution mechanism the infra uses (§ . . Distributor / remote compose over SSH, § . . rootless), so the build EXECUTES on the remote build host (offloading the developer/main host); once the build completes the produced artifacts MUST be brought BACK to the originating main host (scp/rsync/volume copy) for use/flashing/distribution. Building outside a container, or on the bare host, is FORBIDDEN — a release blocker (the "works on my machine" / unreproducible-build class § . . exists to prevent). The build-host target + build-container definitions are config-injected (§ . . , never hardcoded in the submodule); a missing build-container capability is added by EXTENDING the containers submodule upstream (§ . .), never by an ad-hoc host build. Honest boundary (§ . .): the

containerized+distributed build guarantees reproducibility + host-isolation + capacity offload — it does NOT replace the § . . full-suite retest, § . . four-layer artifact→runtime verification, or § . . installable-asset evidence (those still run against the brought-back artifact). Classification: universal (§ . .). Composes § . . (containers submodule — sole orchestration layer), § . . (rootless runtime), § . . (extend-don't-reimplement), § . . (config injection / decoupling), § . . (build-resource stats), § . . (iteration speedup — persistent caches in the container), § . . (no-commit-while-build-writes-artifacts), § . . (installable-asset evidence on the brought-back artifact), § . . (artifact→runtime verification), § . (host memory ceiling — offloading the build preserves the main host). Propagation gate `CM-COVENANT-114-173-PROPAGATION` (literal `11.4.173`) + recommended gate `CM-CONTAINERIZED-DISTRIBUTED-BUILD` (every build runs via the containers submodule on the remote build host; a bare-host build is detected + FAILs) + paired § . mutation. **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--build-on-host`, `--skip-container-build`, `--local-build-ok`, `--no-distributed-build`, `--bare-host-build` flag.

§ . . — Shared-host process-ownership verification mandate (User mandate, - -). On any shared host where other projects/users/agents may run concurrently, every inspection of or action on a process, build, daemon, port, lock, or system-state signal MUST first positively VERIFY the target is OURS before diagnosing or acting. Attribute a process to us ONLY by a strong discriminator — cwd inside this project's tree, argv naming our scripts/artifacts/repo path, a PID we launched + recorded (§ . .), or a lock/port path under our tree — NEVER a loose `pgrep java/gradle/node` name match (it matches every

project on the host). NEVER kill/signal/restart a process not positively ours (other projects' gradle/java/podman/zig are off-limits — operator-workload-safety, § / § . .); the § . daemon-kill remediation MUST be scoped to OUR daemons. When our work waits on host contention from another project's process, surface it (§ . . /§ . .) — block-don't-break. Every pgrep/ps filter MUST exclude self-matches + other-project matches. Forensic anchor (FACT - -): a %-CPU java+gradlew on the shared host was a separate `catalogizer` project's gradle test (cwd=Projects/catalogizer), nearly mis-attributed to our build. Composes § . . / § . . /§ . . /§ . . /§ / § . /§ . . . Classification: universal (§ . .). Propagation gate `CM-COVENANT-114-174-PROPAGATION` (literal `11.4.174`) + recommended gate `CM-PROCESS-OWNERSHIP-VERIFIED` + paired § . mutation. **Canonical authority:** constitution submodule `Constitution.md` § . . . Non-compliance is a release blocker. No escape hatch — no `--assume-process-ours`, `--loose-pgrep-ok`, `--kill-any-matching-process`, `--skip-ownership-check` flag.

§ . . — Conflict-free multi-track work-division + exactly-once claim registry + capability-aware deadlock-proof device-lock (User mandate, - -). Two mandatory multi-track arbitration layers, conflict-free by construction: (A) exactly-once work-item/logical-group claim registry (extends § . . /§ . . ; disjoint file-scope § . . L + worktree/CoW § . . ; no codebase loss § . /§ . . / § . . /§ . .); (B) capability-aware deadlock-proof device-lock (§ . . single-owner; all-or-nothing + non-blocking + TTL-reap breaks all Coffman conditions; append-only JSONL + atomic snapshot § . .); (C) universal/decoupled (§ . . / § . . /§ . .) + evidence-based resource-tuning (§ . . /§ . . , never guessed).

Classification: universal (§ . .). Gates CM-WORK-DIVISION-EXCLUSIVE-CLAIM + CM-DEVICE-LOCK-DEADLOCK-FREE + CM-COVENANT- - -PROPAGATION + paired § . mutation. Composes § . . § . . § . .