Zero-to-Prototype in 12 Guarded Calls: Building Audit-Grade Software (Healthcare Case Study)

Status: **PRE-RELEASE** monaa-guard v0.1.0 is built locally in Rust (Apache-2.0) and is not yet published to crates.io. Commands, outputs and benchmark figures in this document are either marked as examples or as author-reported and unverified. Nothing here is certified test evidence.

Scope disclaimer. Software enforces deterministic pattern-redaction and execution-blocking rules only; it does NOT certify HIPAA, DPDP Act 2023, or clinical regulatory compliance. Consult qualified legal counsel.

1. Universal Idea-Exploration Matrix

Step	Method
1. Pain Isolation	Replace vague feature lists with one deterministic input-to-output bottleneck. Example: a Clinical Triage and Prior-Authorization Intake Router that consumes HL7 FHIR R4 JSON.
2. Hard Compliance Envelope	Separate advisory AI summarization from deterministic fail-closed rules. Reference frameworks: HIPAA Safe-Harbor (Privacy Rule, Security Rule); India DPDP Act 2023 § 6 (unbundled consent); HITRUST and ISO 42001 governance; zero-PHI outbound with BYOK keys.
3. Zero-Cost Setup	Run `monaa-guard init "FHIR Prior-Auth & Triage Prototype"`. This generates `SPEC.md`, `TASKS.md`, `SKILL.md` and `.mcp.json`. No quota is spent.

2. Architecture

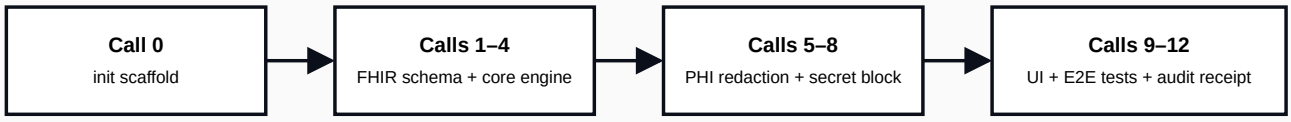


Figure 1: 3×4 Micro-Sprint Funnel (Call 13: paid-tier upgrade, see https://monaa.tech)

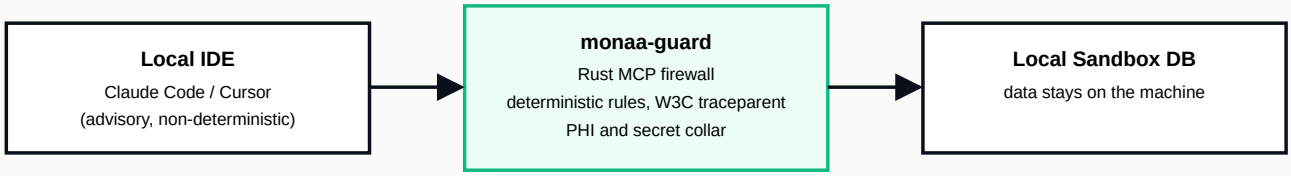


Figure 2: Advisory vs. Deterministic Air-Gap (design diagram; no latency claim shown)

3. Sprint 1: Core FHIR Prototype (Calls 1–4)

Call	Prompt
1. `read_spec_and_define_fhir_types`	Read `SPEC.md` and create strict typed models (`PatientIntake`, `VitalBounds`, `PriorAuthDecision`) with deterministic validation: reject NaN and negative vitals; zero external network calls.
2. `write_red_unit_tests`	Following TDD Red-Green-Refactor, write 6 failing unit tests covering pediatric and adult vital thresholds, missing DPDP § 6 consent rejection, and malformed ICD-10 codes.
3. `implement_deterministic_triage_core`	Implement the deterministic triage and prior-auth rule engine so all 6 unit tests pass. Keep AI strictly advisory: summary narration only, zero autonomous clinical override.
4. `run_test_suite_sprint1`	Execute the test suite and lint checks. Output a 5-line Sprint 1 verification receipt. *See Section 7: this is a template, not a result.*

4. Sprint 2: PHI Firewall, Breach Intercept and Audit Ledger (Calls 5–8)

Call	Prompt
5. `implement_phi_scrubber`	Implement deterministic HIPAA Safe-Harbor (18 identifiers) and India DPDP Act 2023 § 6 PII redaction before any advisory summary string is constructed. Assert that no raw SSN, Aadhaar or MRN survives.
6. `adversarial_secret_and_phi_test`	Run inside a sandbox, never against real systems. Simulate an agent attempting to read `.env` credentials or run an unguarded `rm -rf` cleanup. Capture the policy hard-block receipt. *Latency figures: see Section 6.*
7. `wire_mcp_2026_07_28_endpoints`	Expose `triage/evaluate` and `prior_auth/check` over stateless JSON Schema Draft `2020-12` (MCP 2026-07-28) with W3C `traceparent` propagated through `params._meta`.
8. `run_security_suite_sprint2`	Run the full security and redaction tests. Record actual warning and outbound-byte counts from the run. *See Section 7.*

5. Sprint 3: Clinician Intake UI, E2E Proof and Ship (Calls 9–12)

Call	Prompt

9. `build_clinician_intake_ui`	Create a single-page accessible intake dashboard showing vitals, the deterministic rule disposition, and a visible "Non-Clinical Decision Support" banner.
10. `connect_ui_to_guarded_backend`	Bind form submission to the local guarded engine, with fail-closed validation on missing fields.
11. `run_e2e_clinical_scenarios`	Execute 3 synthetic patient journeys: Routine Approval, Escalated Review, Missing-Consent Block.
12. `generate_release_bundle`	Package the final `README.md` and emit a session summary receipt. *See Section 7.*

6. Benchmark Status

UNVERIFIED The author reports the following figures from a local Apple M2 loopback microbenchmark of a monaa-guard v0.1.0 release build: p50 6 µs, p99 27 µs. No benchmark artifact is included with this document. Treat them as author-reported until the method below is run and published.

Target: p99 < 50 µs. **Reproduction method to publish with any figure:** hardware model and OS; build profile and flags; exact call measured; sample count; warm-up policy; raw output file.

Token-cost figures are omitted from this edition. They will be added with the billing-export file and the prefix-cache configuration that produced them.

7. Run-to-Verify Templates

These are not test results. Each block below shows the form of output a run should produce. Replace every value with the actual result of running the command.

Call 4: Sprint 1 receipt (template)

```
Sprint 1 verification receipt
  unit tests:   [run]/[total] passed
  lint warnings: [count]
  network calls: [count]
  guarded calls used: [n]
```

Call 8: Security suite (template)

```
Sprint 2 security receipt
  warnings:           [count]
  outbound PHI bytes: [count]
  blocked attempts:   [list from run]
```

Call 12: Release receipt (template)

```
Release bundle
  files written: [list]
```

```
  guarded calls used: [n] of [quota shown by the CLI]
```

8. Installation (Pre-Release)

```
# Pre-release v0.1.0. Verify the repository and tag before running.
cargo install --git https://github.com --tag v0.1.0

monaa-guard init "FHIR Prior-Auth & Triage Prototype"
monaa-guard proxy --upstream <cmd>
monaa-guard activate <cert>
monaa-guard doctor   # example output template; run it to see actual values
```

The repository URL, tag and command names are placeholders until the release is published. Confirm them against the actual repository before use.

9. Plans and Pricing

For current Free vs. Pro tier quotas and licensing, see **https://monaa.tech**.