GaugeWright Workbench — Architecture & Security

An enterprise architecture and security reference for engineers, security reviewers, and compliance teams. Written to be verified: every claim is grounded in the product repository, and capability claims carry an honest status. For a short summary see the security overview; for a control-by-control crosswalk see the CAIQ-style appendix.

Status legend — **Available** in the shipped product today · **Built** implemented and tested, not yet operationally deployed · **Planned** committed, not built · **Not implemented** absent today.

Frameworks referenced: SOC 2 (AICPA TSC), ISO/IEC 27001, NIST CSF 2.0 / SP 800-53, NIST SSDF (SP 800-218), NIST AI RMF (+ GenAI Profile), OWASP Top 10 for LLM Applications (2025), MITRE ATLAS, SLSA / SBOM, CSA CCM/CAIQ.

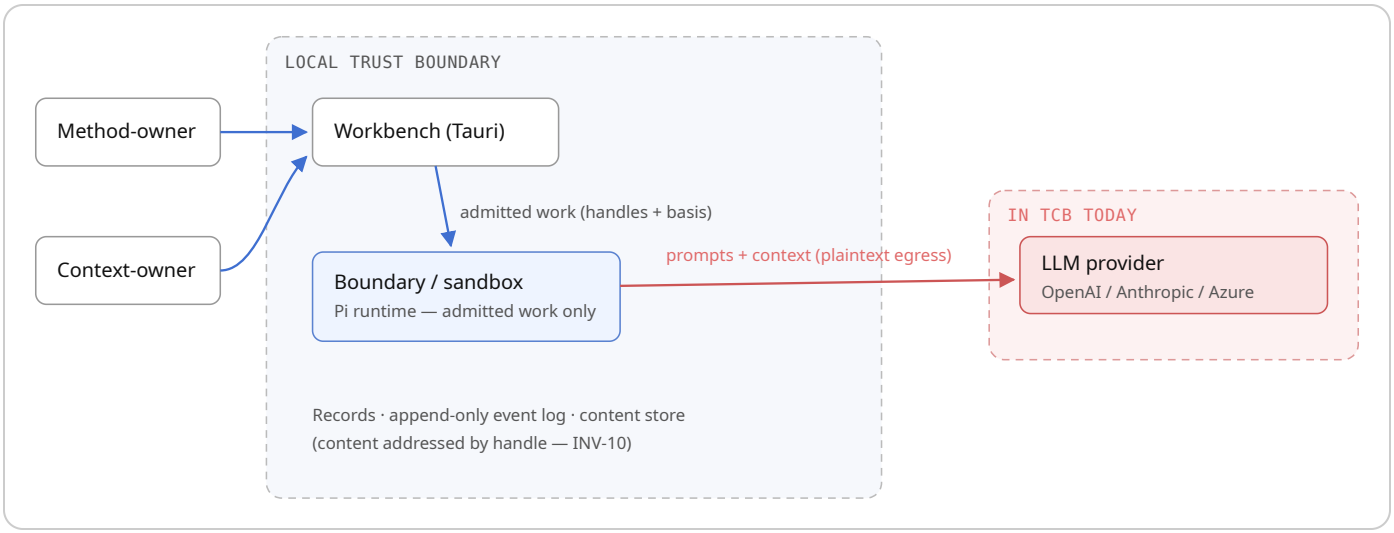
1. Executive trust summary

GaugeWright runs an expert's **method** (an AI agent — instructions, skills, tools) against a client's **private context** (their data) under an enforced **boundary**, so neither the method nor the data leaks to the other party or to the runtime. The boundary is the product; its guarantees are **structural** — expressed as invariants that are **machine-checked** (42 formal models, each with an adversarial "teeth" test), not as policy that can be misconfigured.

- **Shipped today** is a single-party **desktop workbench**: local orchestration, encrypted local storage, kernel-enforced method isolation, an append-only audit log. **Available**
- **Inference is remote.** The agent's reasoning calls the third-party LLM provider you configure; your prompts and the in-scope context are sent to that provider over the network. The model provider is in the trust boundary today. **Available** **Confidential inference Planned**
- **Hosted, relayed, attested, enterprise-identity** capabilities range from code-complete to design-only, and are not operationally available. **Built** **Planned**
- **No third-party attestations yet** — no SOC 2, ISO 27001, or penetration test. Committed and prioritized. **Planned**

2. System context & stakeholders

Principal	Role	Trusts
Method-owner	Builds and packages the agent	The boundary not to leak their method to the client
Context-owner (client)	Provides private data	The boundary not to leak their data to the method-owner or runtime
Runtime / operator	Executes the agent (local today; hosted later)	Is *not* trusted with payload — handles convey no access
End-user (public hosting)	Identified principal inside a consultant's authority	The consultant's scoping
LLM provider (external)	Performs inference	**In the trust boundary today** (sees prompts + context)
Relay (external)	Routes encrypted bytes between authorities	Cannot read payload (`INV-14`)



System context. The local trust boundary contains the workbench, sandboxed runtime, and stores. The highest-scrutiny flow is the plaintext egress to the external LLM provider, which is in the trust boundary today.

Authority & scope (`INV-1`): every durable fact names the authority responsible for it and the scope it touches; no authority writes outside its scope. Projects and tenants are isolated by scope, not by row filters. **Available**

3. Architecture & deployment views

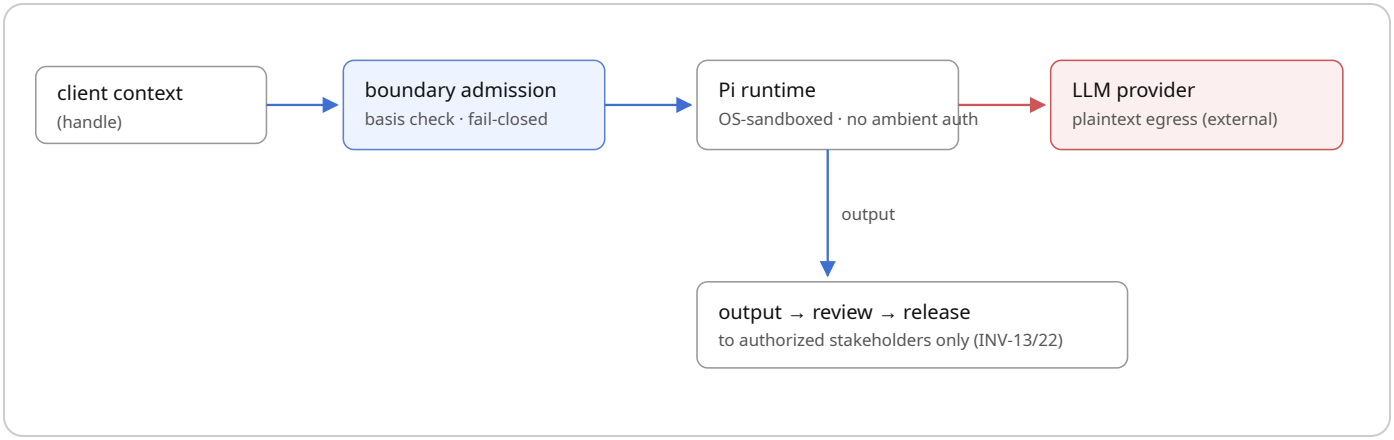
Components. A Rust core (pure, event-sourced reducers) + an application layer (stores, identity, crypto, networking) + a Tauri desktop shell + the **Pi** LLM runtime spawned as a sandboxed subprocess (`--mode rpc`) + a Node sidecar for SAML XML-dsig verification. The Rust backend is a Cargo workspace (`crates/core`, `store`, `workspace`, `boundary`, `pi-bridge`, `app`).

Deployment mode	Description	Status
Local desktop	Orchestration + storage on your machine; federation opt-in; inference calls your configured LLM provider	**Available**
Multi-authority federation	Expert and client collaborate across machines; cert-pinned TLS; relay routes opaque bytes only	**Available**
Hosted multi-tenant	Cloud-hosted relay + compute for consultants' deployments	**Planned**
Attested compute	AMD SEV-SNP confidential VM + Azure Key Vault Secure Key Release; both parties verify the measurement	**Verifier Built** **Live Planned**
Public hosting / embed	Browser-embeddable agent for end-users; per-session isolation, origin allowlist + budget caps	**Planned**

4. Data: classification, flow, residency, retention

Data model (`ADR 0007`). Four kinds, kept distinct: **Records** (durable declarations), **Streams** (append-only events/commands/observations; events are product truth), **Content** (protected payload — prompts, tool results, outputs, transcripts — addressed by **handle** only, `INV-10`), and **Projections** (rebuildable views, never authority, `INV-5`). **Available**

Where data lives. An append-only **SQLite** event log + a **git-backed content store**. Single-machine by default; multi-machine via relay; hosted/attested is roadmap. **Available (local)**



Data flow during a run. Context is admitted under a fail-closed basis; the sandboxed runtime has no ambient authority; egress to the LLM provider is plaintext (the known exposure); output release is gated to authorized stakeholders.

By product default a project's **network egress is open** (a run can reach the model out of the box); an operator **opts into per-project network isolation** (`network_isolated`, default off), which restores kernel-enforced network containment. There is no per-host model-endpoint allowlist proxy yet, so isolation today is all-or-nothing (no egress vs. open egress), not a filtered allowlist. **Available**

Retention & erasure (`ADR 0008`). Revocation stops future use without rewriting the past (`INV-18`); **erasure** tombstones content payload while preserving audit handles/metadata. Modeled (`content-erasure.qnt`, teeth `CANT_UNERASE`) and implemented in the reducer. **Available** — bulk/admin erasure UI and a GDPR DPA are **Planned**.

5. Security architecture → control families (the invariant crosswalk)

The differentiator: each protection invariant maps to a recognized control family **and** to a machine-checked proof. This turns assertions into evidence.

Invariant	Guarantee	Control family (SOC 2 / ISO / NIST)	Proof · teeth
`INV-10`	Handles don't grant access	Confidentiality (CC6.1) · A.8.3 · AC-3	`boundary.qnt` SAFE_EGRESS
`INV-12`	Method & context reads both explicit	Least privilege (AC-6)	`boundary.qnt`

Invariant	Guarantee	Control family (SOC 2 / ISO / NIST)	Proof · teeth
`INV-13`	Cross-authority needs source + target	Confidentiality · ZTA (800-207)	`federation.qnt` · SIGNATURE_FORGERY
`INV-14`	Relays aren't payload authorities	Confidentiality	`federation.qnt` · RELAY_READS_PAYLOAD
`INV-22`	Confidentiality goal	SOC 2 Confidentiality	`engagement-taint.qnt` SOUND
`INV-11`	Execution consumes admitted work	Least privilege · ZTA	`run-admission.qnt` · SKIP_ADMISSION
`INV-20`	Fail-closed	Access control (CC6.x)	`fail-closed.qnt` · FAIL_OPEN
`INV-24`	Method is edit-authored	Integrity · Change mgmt (CC8.1)	`method-integrity.qnt` · USE_WRITE_LEAKS
`INV-6/7/8`	Append-only, ordered, replayable	Processing integrity · Audit (AU-9)	event-store semantics
`INV-19`	Idempotent admission	Processing integrity	`idempotency.qnt` · NO_DEDUP

CI enforces both that each invariant *holds* and that its tooth *bites* (flips the probe true and asserts failure). **Available**

6. Identity & access management

- **OIDC** — id-token verifier (JWKS signature, `iss` / `aud` / `exp` / `nbf`, claim → authority mapping), fail-closed; verified per-commit against self-hosted Keycloak. **Available**
- **SAML 2.0** — verification delegated to a hardened Node sidecar behind the same seam; **single-use assertion** replay defense. **Available**
- **SCIM** — inbound provisioning/de-provisioning. **Built**; outbound sync **Planned**
- **RBAC/ABAC** — role assignments + an ABAC policy evaluator. **Built**; admin console **Planned**

- **MFA** — Not implemented in the product (org-level enforcement roadmap)
- **Build-vs-buy** — own SSO/SCIM, no broker in the auth path (`ADR 0056`); trust source is SOC 2 + a SAML-scoped pen test (Planned)

Live interop with specific IdP vendors (Okta, Entra, Google) and deploy-time secret wiring are Planned. The shipped desktop product is local and needs no account.

7. Threat model

Methodology: **STRIDE** for the platform, **OWASP LLM Top 10 (2025)** and **MITRE ATLAS** for the AI surface.

STRIDE	Mitigation	Status
Spoofing	Actor authenticity verified before admission (`INV-21`); OIDC/SAML; signed governance envelopes	Available
Tampering	Append-only immutable events (`INV-6`); AEAD at rest; method surface read-only at kernel (`INV-24`)	Available
Repudiation	Per-actor append-only audit + SIEM export; *cryptographic non-repudiation not yet shipped*	Available Planned
Information disclosure	Handles don't grant access (`INV-10`); both reads explicit (`INV-12`); opt-in per-project network isolation (egress open by default); relay opacity (`INV-14`)	Available
Denial of service	Idempotent admission (`INV-19`); entitlement/budget caps; *platform rate-limiting limited*	Partial
Elevation of privilege	No ambient authority (`INV-11`); retries can't widen scope (`INV-17`); fail-closed (`INV-20`); kernel sandbox	Available

OWASP LLM Top 10 (2025)	Posture	Status
LLM01 Prompt injection	No ambient authority; acts only on admitted work; tool calls gated; egress open by default with opt-in per-project network isolation	**Available**
LLM02 Sensitive-info disclosure	The known exposure: prompts + context reach the LLM provider in plaintext. Disclosed, not hidden; confidential inference Planned	**Disclosed**
LLM06 Excessive agency	Execution consumes admitted work only (`INV-11`); kernel sandbox bounds tool/file/network reach	**Available**
LLM07 System-prompt leakage	Method definition runs read-only, kernel-enforced; a work chat cannot read/rewrite it (`INV-24`)	**Available (Linux/macOS)**
LLM05 Improper output handling	Output review lifecycle (`SOUND_RELEASE`); release gated on stakeholder taint	**Built**
LLM10 Unbounded consumption	Entitlement gate + per-engagement budget caps	**Built**
LLM03 Supply chain	Pinned `Cargo.lock`; reproducible image + measurement digest. No SBOM / CVE scanning yet	**Partial**
LLM04 / LLM08 / LLM09	Poisoning / vector weaknesses / misinformation — out of current scope (no first-party training/RAG/fact-checking)	**Not implemented**

MITRE ATLAS: the sandbox + admitted-work model addresses ML-model exfiltration and unauthorized-use tactics; model-extraction via the provider remains a residual until confidential inference.

8. Cryptography & key management

- **At rest** — **AES-256-GCM** (AEAD via `ring`; no OpenSSL), `nonce(12)‖ciphertext‖tag(16)`, fresh nonce per call; tamper/wrong-key fails the auth tag. **Available**
- **Key management** — **envelope encryption**: a 256-bit DEK wrapped by a KMS KEK in **Azure Key Vault** via a `KeyWrap` seam; KMS integration verified live. **Built**
- **Signatures** — real **P-256 ECDSA** (pure-Rust) for governance envelopes and attestation reports. **Available**
- **Attestation** — real **AMD SEV-SNP** quote verifier: ARK→ASK→VCEK chain + ECDSA-P384/SHA-384 signature + `report_data` freshness + measurement allow-list; tested against **real Milan vectors**. **Built** — quote generation needs a confidential VM (**Planned**)
- **TLS** — cert-pinned egress seam for attested model endpoints. **Planned**

9. Audit, logging, monitoring & incident response

- **Audit log** — per-actor, append-only `{actor, action, target}` in a reserved scope; references only (`INV-10`); position-ordered, filterable. **Available**
- **SIEM export** — `HttpAuditSink` POSTs each entry as JSON to a customer-configured collector (Splunk/Datadog/webhook) over rustls. **Available**
- **Tamper-evidence** — append-only is enforced *semantically*, **not yet cryptographically** (no signature/merkle chain). **Planned**
- **Monitoring / alerting / incident response** — **Not implemented** (no production observability, runbooks, or IR procedures in tree today)
- **Uptime / SLA** — **Planned**

10. AI governance (NIST AI RMF)

- **Govern** — AI use is bounded by the same authority/scope/admission model as everything else; method changes are edit-authored and audited (`INV-24`).
- **Map** — the LLM provider is named in the trust boundary; data-to-model flow is documented (§4). BYO-credentials (`ADR 0053`) lets the LLM relationship be the customer's own subprocessor. **Built (core)**

- **Measure** — protection properties machine-checked (§13); no model-performance / bias / hallucination evaluation today (out of scope).
- **Manage** — output review gates release; budget caps bound consumption; per-project network isolation is opt-in (egress open by default).

ISO/IEC 42001 (certifiable AI management system) is named as a future target, not a current certification. **Planned**

11. SDLC & software supply chain

CI gates every push (`.github/workflows/ci.yml`): `cargo fmt --check`, `cargo clippy -D warnings`, `cargo test --workspace`, web typecheck + unit, the SAML sidecar tests, `quint typecheck` + invariant/teeth model checks, and `scripts/audit-gate.py` (a tracker cannot close a gate while any coverage row is unfinished). Tier-1 adds a real networked relay + governance handshake; a Compose lane exercises NAT-isolated federation. **Available**

Mapped to NIST SSDF: version control + `Cargo.lock` (PO1.1); fmt/clippy enforced (PO2.1, PS2.1); provenance via reproducible image + digest (PO2.2, **Built**); proptests (PS3.1); coverage-gate review records (PS4.1).

Supply-chain expectation	State
Pinned dependency lockfile	**Available** (`Cargo.lock`)
Reproducible build + measurement digest	**Built** (`flake.nix`, `image-digest.yml`)
SBOM (CycloneDX/SPDX)	**Not implemented**
SLSA provenance attestation	**Not implemented**
Dependency CVE scanning (cargo audit / Dependabot)	**Not implemented**
SAST / secret scanning in CI	**Not implemented**
Code signing / notarization of desktop builds	**Not implemented** (builds are unsigned)

These are known, named gaps — not oversights — and are the most actionable near-term hardening items.

12. Compliance posture & roadmap

Item	State
SOC 2 Type II	**Planned** (highest-priority; trust source for own-built SSO)
ISO 27001	**Not implemented** (no roadmap committed)
Penetration test (SAML-scoped)	**Planned**
DPA + published subprocessor list	**Planned**
GDPR right-to-erasure	**Built** (erasure model); DPA **Planned**

Subprocessors (see also the CAIQ appendix): your configured LLM provider(s); Microsoft Azure (Key Vault, confidential VM — Built/Planned modes); Stripe (billing — Planned); customer-operated IdP. With BYO-credentials the LLM provider is the *customer's* subprocessor.

13. Assurance evidence

- **42 Quint formal models** covering the protection invariants; CI verifies each invariant holds and each adversarial "teeth" probe bites. **Available**
- **Property tests** in the pure core tie the Rust reducers to the models. **Available**
- **Live verifications**: Keycloak OIDC per-commit; Azure Key Vault KMS; real SEV-SNP Milan vectors; NAT-isolated federation. **Available** **Built**
- **No independent third-party penetration test or audit yet.** **Planned**

Security contact: jack@gaugewright.com · Source & formal specs: github.com/jamesjscully/un-tie · Reviewed against spec rev 2026-06.

Change log — 2026-06: first enterprise edition (arc42 + security/AI overlay; control crosswalk; STRIDE + OWASP LLM Top 10 threat model; honest supply-chain and compliance gaps).