

GaugeDesk — Security Overview

*For IT and security teams evaluating GaugeWright as a vendor. Written to be verified, not sold; the formal model lives in the [architecture document](#) and the product's *specs*/.*

Status legend — Available in the shipped product today · Built implemented and tested in code, not yet operationally available · Planned committed, not yet built.

[Save as PDF](#) [Download PDF](#) [Markdown](#)

What it is

A desktop workbench for building and running AI agents (the "method") against private data (the "context") under an enforced boundary. **It orchestrates locally; the agent's reasoning is performed by a third-party LLM provider you configure.** Hosted, relayed, and attested deployment modes are roadmap.

Where your data goes

Stage	What happens	Status
Inference	The agent's reasoning calls the LLM provider you configure (e.g. OpenAI, Anthropic, Azure OpenAI). Your prompts and the in-scope context are sent to that provider over the network. You authenticate the provider; its retention/training terms are the provider's, not ours. No local-only inference today.	Available
At rest	Content is addressed by handle, not stored inline. Envelope encryption (AES-256-GCM, data key wrapped by a KMS key) is implemented and verified live against Azure Key Vault for server/hosted deployments. Confirm the at-rest posture for your specific mode.	Built (server)
In transit	Cross-machine traffic uses certificate-pinned TLS; the rendezvous relay routes only opaque encrypted bytes and cannot read payload. Cross-party messages are signed and verified before admission.	Available

The one fact to take away: local orchestration is not local inference. GaugeDesk does not remove your LLM provider from the trust boundary. If data may not leave for a third-party model, you need a provider you've contracted (bring-your-own account) or the future confidential-inference mode, which is not yet available.

Isolation & access control

- **Authority / scope model.** Every fact names a responsible authority and the scope it touches; no authority writes outside its scope. Projects and tenants are isolated by scope, not row-level filters. Available

- **Handles don't grant access.** Holding a reference conveys no read; payload access requires a separate, explicit basis evaluated at the boundary. Checks are **fail-closed** — uncertainty denies. Available
- **Method is read-only at runtime, kernel-enforced.** A running agent cannot rewrite its own prompt, policy, or tools. Enforced by an OS sandbox (bubblewrap on Linux, Seatbelt on macOS); every write path, including a shell, is blocked at the kernel. **Windows: enforcement pending — runs fail-closed (refused) until the backend ships.** Available (Linux/macOS)

Identity & access management

OIDC and SAML sign-in, SCIM provisioning/de-provisioning, coarse role-based access (owner/admin/member/viewer/billing), and org-level MFA enforcement are **built and tested** (OIDC verified per-commit against Keycloak). Live interop with specific IdP vendors (Okta, Entra, Google) and deploy-time wiring are **planned**. The shipped desktop product today is local and needs no account. Built

Audit & integrity

- Every durable fact is an **append-only, immutable event**, totally ordered within its scope; corrections are new events, never edits. State is the deterministic replay of events.
- Audit entries carry actor / action / target references only (no payload) and export to your SIEM via a customer-configured webhook.
- **Honest limit:** append-only is enforced *semantically*, not yet with cryptographic tamper-evidence or non-repudiation. Cross-party log trust today rests on the operator, not a signature chain. append-only Available
tamper-evidence Planned

Confidential / attested compute

Designed and partially built. The AMD SEV-SNP attestation **quote verifier** is implemented and tested against real hardware vectors (validates the ARK → ASK → VCEK chain and report signature). Live confidential-VM hosting, the KMS Secure-Key-Release wiring, and confidential *inference* (removing the model provider from the trust boundary) are **not yet operational**. Verifier Built Live attestation Planned

Software assurance

The protection model's invariants are **machine-checked** as formal models (Quint), each paired with an adversarial "teeth" test that confirms the check fails when the protection is removed (fail-open, skip-auth, method-write-leak, ...). The core reducers are pure and property-tested against those models. No independent third-party penetration test has been performed yet. Formal verification Available Pen test Planned

Compliance & certifications

None available to cite today. SOC 2 Type II, a Data Processing Agreement with a published subprocessor list, and an independent penetration test are committed and prioritized but not yet delivered. Subprocessors will include your configured LLM provider(s) and, for hosted/attested modes, Microsoft Azure. None today Planned

Current stage

What you can deploy today is the **local desktop workbench** — single-party, local orchestration, remote inference. The hosted, relayed, attested, and enterprise-identity capabilities above range from code-complete-and-tested to design-only, but are **not operationally available** and carry no live certification yet. The security *model* is verified; operational *readiness* for a regulated mid-market deployment is still in progress.

Security contact: Jack@GaugeWright.com · Source & formal specs: github.com/jamesjscully/gaugedesk ·
Reviewed against spec rev 2026-06.

Products

[GaugeDesk](#) [WhippleScript](#)

Developers

[GaugeDesk docs](#) [WhippleScript docs](#) [GitHub](#)

Solutions

[For AI service agencies](#) [Pricing](#)

Trust

[Security & architecture](#) [Compliance \(CAIQ\)](#)

Company

[About](#) [Contact](#)

© 2026 GaugeWright LLC