Summit IntelGraph

Architecture Brief

TECHNICAL ARCHITECTURE BRIEF — STACK, INTEGRATION & DEPLOYMENT

PLATFORM OVERVIEW

Summit IntelGraph is a full-stack graph intelligence and AI analysis platform built on a modern distributed architecture designed for auditability, multi-tenant isolation, and on-premise or cloud deployment. The platform is TypeScript/Node.js throughout the API layer, React 18 on the frontend, and Python for AI/ML pipelines. The v4.0.3 release represents 7,300+ commits across 422 releases with active CI/CD enforcement.

CORE STACK

Layer	Technology	Purpose
Frontend	React 18 / Vite / Material-UI / Cytoscape.js	Graph visualization, investigation UI, temporal slider
API	Node.js / Express / Apollo GraphQL	Unified API layer, real-time subscriptions, auth
Graph DB	Neo4j (APOC + GDS plugins)	Entity relationships, event networks, community detection
Relational DB	PostgreSQL + pgvector	Structured data, audit logs, vector embeddings for entity resolution
Time-Series	TimescaleDB	Telemetry, metrics, temporal edge data
Queue / Stream	Redis → Kafka/Redpanda	Background jobs, high-velocity ingest pipeline
AI / ML	LangChain / OpenAI / local LLM	GraphRAG Copilot, entity extraction, narrative simulation
Orchestration	Maestro (BullMQ)	AI pipelines, scheduled jobs, graph Gardener
Observability	Grafana / Prometheus / OpenTelemetry	Performance monitoring, cost tracking, alerting
CI/CD	GitHub Actions / Jules AI Agent	Automated build, test, release pipeline

INTEGRATION POINTS

Data ingest	REST API ingest endpoint; CSV/JSON/GraphML batch loader; STIX 2.1 native parser; Kafka consumer for stream ingest; air-gap loader for offline environments
Authentication	JWT-based auth with role/clearance claims; OIDC/SAML integration for enterprise SSO; per-tenant credential scoping
Export / API	Full GraphQL API with Apollo Playground; REST endpoints for programmatic access; JSON and CSV export; case note export in markdown and HTML
Existing tooling	STIX 2.1 ensures interoperability with existing threat intel platforms (MISP, OpenCTI, commercial TIPs); Kafka integration enables connection to existing SIEM/SOAR pipelines

On-premise deployment	Docker Compose for single-node; Helm charts for Kubernetes multi-node; no external network calls required in air-gap mode; all AI inference can run locally with open-weight models

SECURITY ARCHITECTURE

Multi-tenant isolation	Every query includes tenant predicate injection. ESLint rule fails CI if a Cypher query is missing tenant scope. Dual-tenant integration test in CI asserts zero cross-tenant data leakage.
Clearance-level LBAC	Middleware proxy intercepts all Neo4j queries and injects clearance predicates. Four levels: UNCLASSIFIED, SENSITIVE, CONFIDENTIAL, SECRET. Claim sourced from JWT.
Audit trail	Append-only PostgreSQL audit log. Application role has INSERT/SELECT only — no UPDATE or DELETE. Nightly SHA-256 chain hash for tamper evidence. /health/audit-integrity endpoint for verification.
Hypothesis segregation	Confirmed intelligence and AI-generated predictions stored in separate graph layers (:Entity vs :Hypothesis). No mechanism for hypothesis data to appear in confirmed intelligence queries without explicit promotion with audit entry.
Secret management	GitHub branch protection with required secrets inventory check. No secrets in codebase — enforced by CI security scan. Environment-variable injection for all credentials.

DEPLOYMENT OPTIONS

Local / dev	make up — Docker Compose brings up full stack in one command. All services containerized. Smoke test via make smoke.
On-premise	Helm chart deployment to customer Kubernetes cluster. No external network calls in air-gap mode. All AI inference via locally-hosted model or disconnected API.
GovCloud	Deployable to AWS GovCloud or Azure Government. No proprietary cloud dependencies — all components are open-source infrastructure.
MSSP / multi-tenant	Multi-tenant architecture supports multiple client environments from a single deployment with strict data isolation.