



# EUGENE PLATFORM

## Consolidated Programme Briefing

---

Overview, Scope, Current-State Architecture, Gaps and Technical Recommendations

|                |                                              |
|----------------|----------------------------------------------|
| PREPARED FOR   | CSL Behring — Business Development & R&D     |
| PROGRAMME      | Eugene Knowledge Graph & Agentic AI Platform |
| DATE           | May 2, 2026                                  |
| CLASSIFICATION | Confidential — Internal Use Only             |
| VERSION        | 2.1 (Book Edition)                           |

## TABLE OF CONTENTS

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>FRONT MATTER</b>                                    | <b>3</b>  |
| About This Document . . . . .                          | 3         |
| Document Map . . . . .                                 | 3         |
| <b>SECTION 1 — OVERVIEW OF EUGENE</b>                  | <b>3</b>  |
| 1. What Eugene Is . . . . .                            | 3         |
| <b>SECTION 2 — SCOPE AND CAPABILITIES</b>              | <b>4</b>  |
| 2. What Eugene Can Answer . . . . .                    | 4         |
| 3. Data Coverage . . . . .                             | 5         |
| <b>SECTION 3 — DETAILED CURRENT-STATE ARCHITECTURE</b> | <b>6</b>  |
| 4. High-Level Architecture . . . . .                   | 6         |
| 5. End-to-End Request Flow . . . . .                   | 8         |
| 6. Data Model . . . . .                                | 11        |
| 7. Authentication & Authorisation . . . . .            | 11        |
| 8. AWS Deployment Topology . . . . .                   | 13        |
| <b>SECTION 4 — GAPS IDENTIFIED</b>                     | <b>14</b> |
| 9. Audit Outcome . . . . .                             | 14        |
| 10. Critical Findings . . . . .                        | 14        |
| 11. High-Severity Findings . . . . .                   | 14        |
| 12. Medium-Severity Findings . . . . .                 | 15        |
| 13. Operational, Data, and Experience Gaps . . . . .   | 16        |
| 14. Architectural Strengths to Preserve . . . . .      | 17        |
| <b>SECTION 5 — TECHNICAL RECOMMENDATIONS</b>           | <b>17</b> |
| 15. Next-Generation Target Architecture . . . . .      | 17        |
| 16. Four-Phase Roadmap . . . . .                       | 20        |
| 17. User-Experience Enhancement Programme . . . . .    | 21        |
| 18. Non-Functional Targets . . . . .                   | 21        |
| 19. Programme Risks and Mitigations . . . . .          | 22        |
| 20. Recommendation . . . . .                           | 22        |
| <b>APPENDICES</b>                                      | <b>22</b> |

**Appendix A — Source Documents . . . . . 22**

**Appendix B — Glossary . . . . . 23**

## FRONT MATTER

### About This Document

This briefing is the single, consolidated reference for the Eugene biomedical knowledge graph and agentic AI programme at CSL Behring. It is intended for client and stakeholder presentations and replaces the need to navigate between the Project Validation Document, the Stage 1 Assessment & Audit Report, the Developer Guide, and the next-generation architecture proposal. The five sections that follow tell a complete technical story: what Eugene is, what it does today, how it is built, where the gaps are, and what we recommend doing about them.

Every architectural claim in this document is grounded in source material in the docs/ folder. Where a measurement, a finding, or a remediation is cited, it traces to one of the four primary sources above. Diagrams are reproduced from the canonical Mermaid sources in the Developer Guide, augmented with the next-generation architecture preserved in CSL\_Arch\_nextGen.pdf.

### Document Map

- Section 1 — Overview of Eugene: who it is for, what it does, and why it matters to CSL Behring.
- Section 2 — Scope and Capabilities: what the platform answers today, what data sources are integrated, and what use cases are in production.
- Section 3 — Detailed Current-State Architecture: the five running services, the data model, the request flow, the authentication design, and the AWS deployment topology.
- Section 4 — Gaps Identified: the twenty-one findings of the Stage 1 audit (CRITICAL, HIGH, MEDIUM) plus operational, data, and experience gaps observed in this assessment.
- Section 5 — Technical Recommendations: the next-generation target architecture, the four-phase remediation roadmap, the user-experience enhancement programme, and the non-functional commitments.

■ *Confidential. Prepared for CSL Behring stakeholders. Reproduction or distribution outside CSL Behring requires programme-lead approval.*

## SECTION 1 — OVERVIEW OF EUGENE

### 1. What Eugene Is

Eugene is an enterprise-grade, AI-powered biomedical knowledge graph platform purpose-built for CSL Behring's competitive intelligence and research analytics operations. It allows non-technical users — Business Development analysts, R&D; scientists, and Medical Affairs professionals — to pose complex, multi-hop biomedical questions in plain English and to receive synthesised, evidence-backed answers in seconds. The underlying graph integrates authoritative public data (USPTO patents, PubMed literature, ClinicalTrials.gov records) with internal CSL data assets, delivering a single conversational surface across what was previously a fragmented set of databases and tools.

#### 1.1 Mission Statement

Eugene exists to compress the time required to answer biomedical competitive-intelligence and research questions from days of manual database navigation to seconds of conversational interaction. The Project Validation Document positions Eugene as a platform whose explicit objective is that a non-technical user should pose a multi-hop biomedical question in plain English and receive a synthesised, evidence-backed answer in real time.

## 1.2 Stakeholder Map

| Role                  | Stakeholder Group                     | Responsibility                               |
|-----------------------|---------------------------------------|----------------------------------------------|
| Executive Sponsor     | CSL Behring Leadership                | Strategic direction; budget approval         |
| Product Owner         | R&D; / Competitive Intelligence       | Requirements prioritisation; user acceptance |
| Platform Engineering  | AI / Data Engineering Team            | Architecture, development, deployment        |
| End Users             | Scientists, Analysts, Medical Affairs | Query, interpret, and act on insights        |
| Security / Compliance | IT Security; Legal                    | Authentication, data governance review       |
| Infrastructure        | Cloud Platform Team                   | AWS environment management                   |

## 1.3 Business Value

Eugene delivers five business outcomes today:

- Accelerated competitive intelligence — analysts query drug pipelines and patent landscapes in seconds rather than hours of manual database navigation.
- Reduced research latency — the agent eliminates the need for manual Cypher authorship, removing a hard skill barrier between scientists and the graph.
- Centralisation of biomedical knowledge — USPTO, PubMed, ClinicalTrials.gov, DrugBank, and internal CSL data are reachable through a single conversational surface.
- Scalable cloud-native infrastructure — AWS ECS Fargate, ALB-fronted, Terraform-managed in two environments (development and production).
- Auditability — every agent reasoning chain decomposes into structured MCP tool calls that are individually loggable and explainable.

# SECTION 2 — SCOPE AND CAPABILITIES

## 2. What Eugene Can Answer

### 2.1 Representative Questions

The following questions, drawn from the validation document and demonstrated end-to-end on the platform today, illustrate the breadth of capability. Each is answerable by a non-technical user in under thirty seconds:

- Which organisations hold patents on drugs targeting the same indication as our lead compound, and what is their clinical trial status?
- Find all organisations in Germany with active Phase III trials in rare disease.
- Which drugs target EGFR and what trials are sponsored by which organisations?
- Trace the connection between AstraZeneca and the KRAS gene through patents and trials.
- What are the relationships of Hemophilia A in the graph?
- What are the drug aliases for Adderall?

- Generate a table of drugs, diseases and clinical trials researched by Biogen Inc.
- What recent PubMed studies mention ABL1?

## 3. Data Coverage

### 3.1 Data Sources Integrated

The Mermaid pipeline below — reproduced from the developer guide — shows how each source is downloaded, transformed into Entity / Relationship records, expanded with type nodes, and merged into Neo4j:

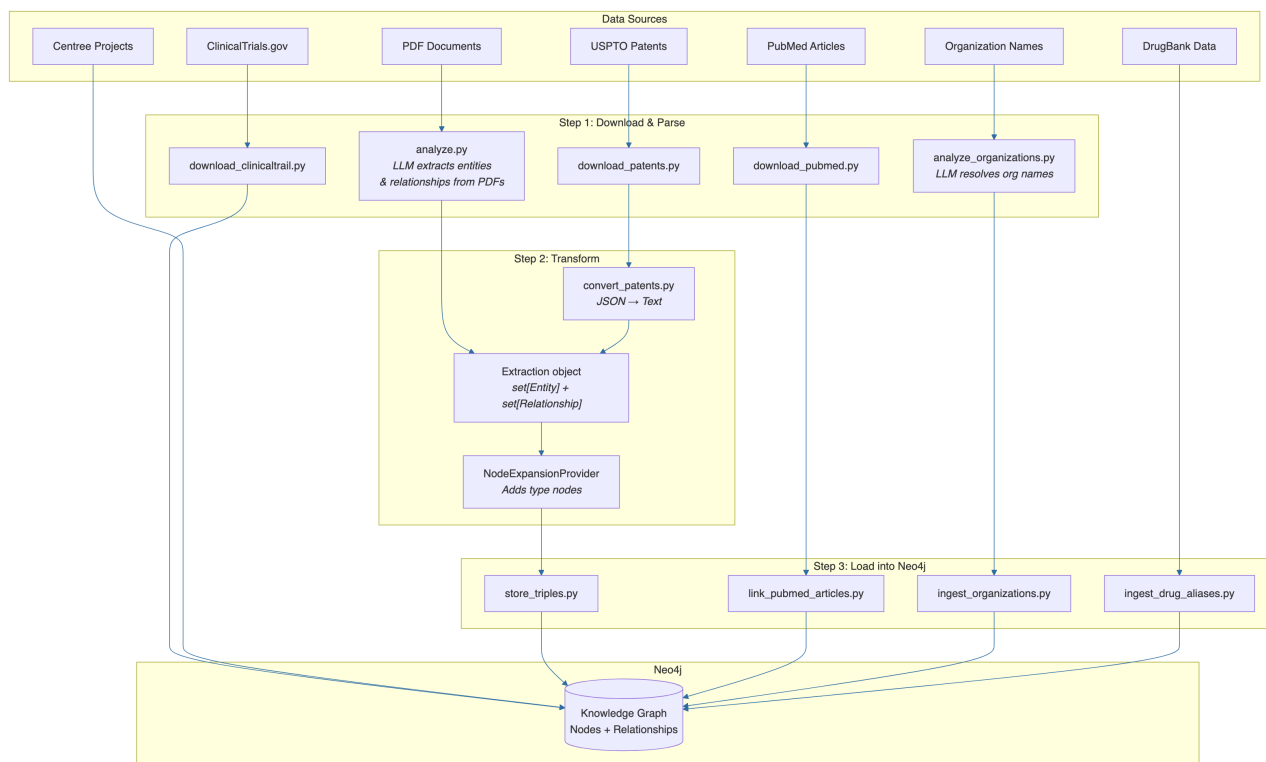


Figure (Mermaid). Data ingestion pipeline (download → transform → load), from DEVELOPER\_GUIDE.md §5.

| Source             | Status | Description                             |
|--------------------|--------|-----------------------------------------|
| USPTO Patents      | Loaded | Patent records, filing dates, assignees |
| PubMed             | Loaded | Journal articles, abstracts, citations  |
| ClinicalTrials.gov | Loaded | Trial records, phases, sponsors         |
| Internal CSL data  | Loaded | Proprietary biomedical and deal data    |
| DrugBank           | Loaded | Drug mechanisms, indications, status    |
| Organisation DB    | Loaded | Pharma / biotech company records        |

### 3.2 Graph Scale

The Eugene knowledge graph contains approximately 484,000 nodes organised across nine principal labels and approximately 21 million typed, directed edges. Memory is tuned per environment, with production running 16 GB pagecache and 4–8 GB heap on dedicated EC2.

| Node Label | Approx. Count | Key Properties                                      |
|------------|---------------|-----------------------------------------------------|
| Drug       | 4.6 k         | node_id, name, mechanism_of_action, status, aliases |

| Node Label       | Approx. Count | Key Properties                                      |
|------------------|---------------|-----------------------------------------------------|
| Disease          | 17.1 k        | node_id, name, icd_code, mesh_id, synonyms          |
| GeneProtein      | 27.7 k        | node_id, name, hgnc_id, entrez_id, symbol           |
| ClinicalTrial    | 49.3 k        | node_id, nct_id, title, phase, status, sponsor      |
| Patent           | 50 k +        | patent_id, title, filing_date, grant_date, assignee |
| Organization     | 26.1 k        | node_id, name, hq_country, org_type, aliases        |
| Publication      | 100 k +       | node_id, pmid, title, journal, publication_date     |
| Annotation       | —             | source, text_span, confidence                       |
| Therapeutic Area | ~ 500         | node_id, name                                       |

### 3.3 Capability Inventory

- Twenty REST endpoints covering identity, foundation queries, drug/disease/patent/PubMed/organisation domains, similarity, and stats.
- Twelve MCP tools mediating the agent's access to the API: `fetch_identity`, `fetch_by_label`, `fetch_similar`, `lookup_node_by_value`, `fetch_node_details`, `fetch_drug_aliases`, `fetch_facts`, `fetch_node_relationships`, `fetch_paths`, `has_reachable_path`, `find_organization_names`, `find_organization_assets`.
- Two-hop graph traversal with hard-cap (`MAX_SUPPORTED_HOPS=2`) to prevent fan-out.
- Token-by-token streaming via Server-Sent Events for ChatGPT-class user experience.
- Dual-LLM provider auto-detect (OpenAI GPT-4.1 and Anthropic Claude Sonnet) with one-environment-variable swap.
- Microsoft Entra ID single sign-on plus Eugene-issued HS256 JWT for service-to-service propagation.

## SECTION 3 — DETAILED CURRENT-STATE ARCHITECTURE

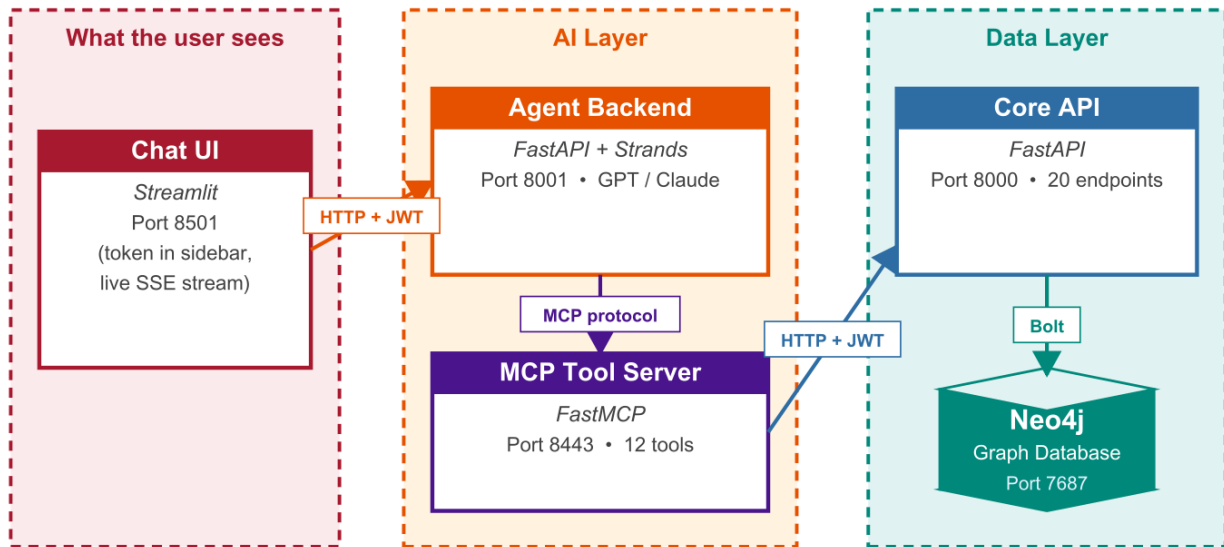
### 4. High-Level Architecture

#### 4.1 Five-Service Topology

Eugene is composed of five running services arranged in a five-tier microservice architecture, following Domain-Driven Design with hexagonal (ports-and-adapters) layering inside the Core API. The diagram below reproduces the canonical architecture from the Developer Guide.

## Eugene Platform — High-Level Architecture

(per docs/DEVELOPER\_GUIDE.md — flowchart LR)



The same architecture is also captured in the developer guide as a Mermaid flowchart, reproduced below verbatim from docs/DEVELOPER\_GUIDE.md:

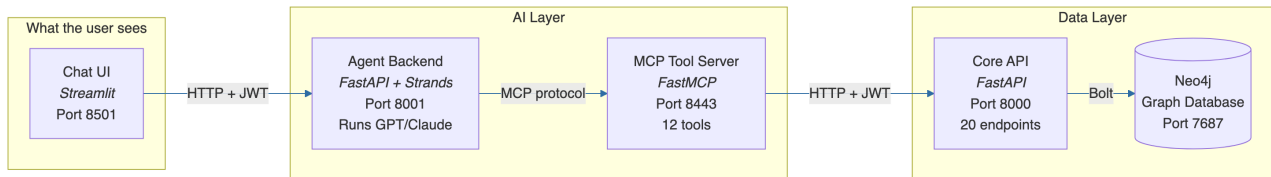


Figure (Mermaid). High-level architecture flowchart, from DEVELOPER\_GUIDE.md §2.

## 4.2 Tier Responsibilities

| Tier                 | Service                | Responsibility                                 |
|----------------------|------------------------|------------------------------------------------|
| Presentation         | eugene-agent-ui        | User interaction; live token rendering         |
| Agent                | eugene-agent-ws        | Reasoning loop; tool selection; LLM inference  |
| Tool / Orchestration | eugene-mcp             | Schema-validated tool surface; JWT enforcement |
| Data API             | eugene_ws              | Canonical graph API; 20 REST endpoints         |
| Persistence          | Neo4j 5.26 CE + Milvus | Graph + vector storage                         |

## 4.3 Technology Stack



## Technology Stack — Layered Inventory

|            |                 |                      |                |                 |                       |            |
|------------|-----------------|----------------------|----------------|-----------------|-----------------------|------------|
| Frontend   | Streamlit       | Next.js + TypeScript | Cytoscape.js   | Chakra UI       |                       |            |
| Agent      | Strands (ReAct) | Anthropic Claude     | OpenAI GPT-4.1 | FastMCP client  |                       |            |
| Tool / API | FastMCP         | FastAPI              | Pydantic v2    | httpx / aiohttp |                       |            |
| Data       | Neo4j 5.26 CE   | APOC                 | GDS            | Milvus          | sentence-transformers |            |
| DevOps     | Docker          | ECS Fargate          | ECR            | Terraform       | GitHub Actions        |            |
| Quality    | Black           | Ruff                 | isort          | Pytest          | MLflow                | pre-commit |

The Core API itself is organised by biomedical domain into the following router groups, faithful to the Mermaid graph in the developer guide:

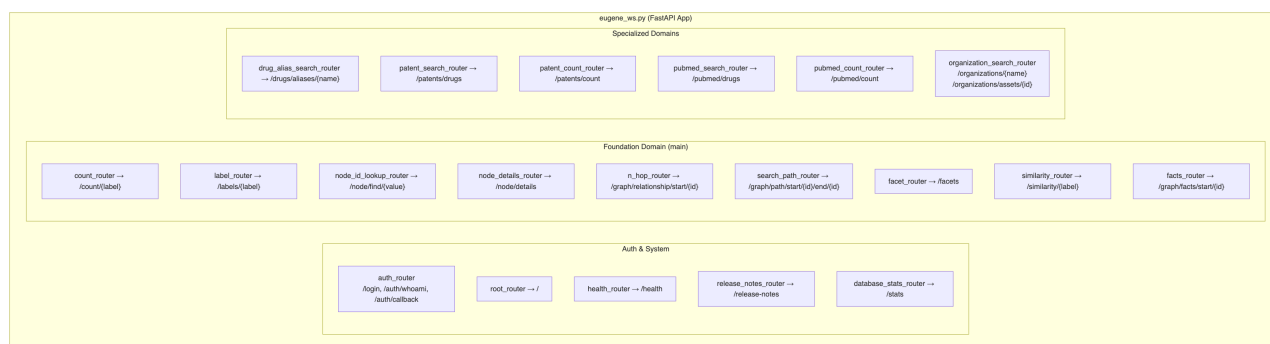


Figure (Mermaid). Core API router groups, from DEVELOPER\_GUIDE.md §6.

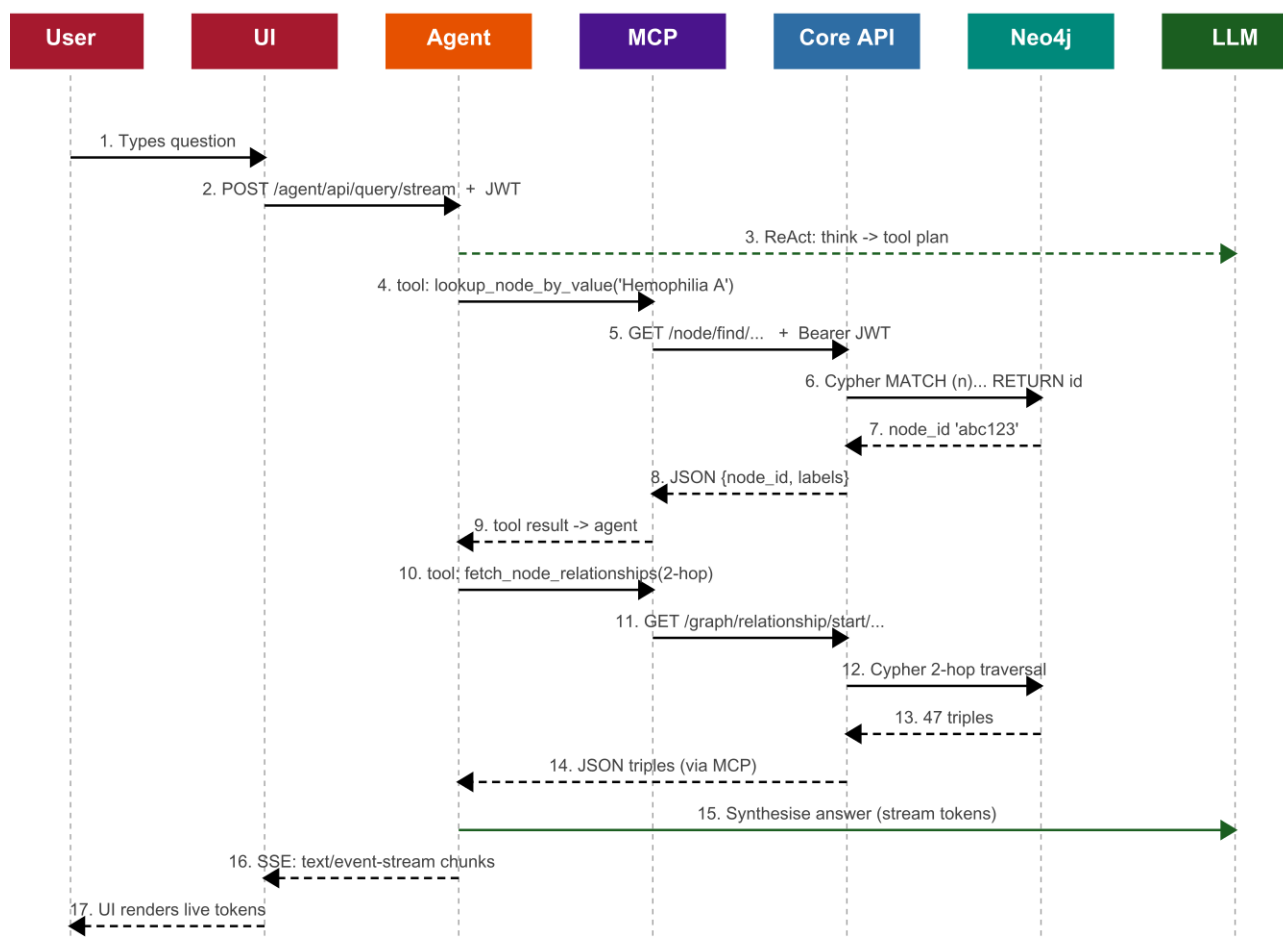
The stack is intentionally Python-centric end-to-end (3.12 / 3.13), with FastAPI for HTTP services, Strands for the agent loop, FastMCP for the tool surface, Neo4j 5.26 Community Edition with APOC and GDS for the graph, and Milvus for vector similarity. Containerisation is Docker; orchestration is AWS ECS Fargate; infrastructure is Terraform-managed.

## 5. End-to-End Request Flow

### 5.1 Sequence Walk-through

When a user submits a question, control flows through every tier of the platform. The diagram below traces the canonical seventeen-step path for the example query "What are the relationships of Hemophilia A?":

## End-to-End Query Sequence — User question to streamed answer



The developer guide records the same flow as a UML sequence diagram, showing the layered DDD pipeline inside the Core API (Router → Orchestrator → Provider → Adapter → Neo4j → mappers):

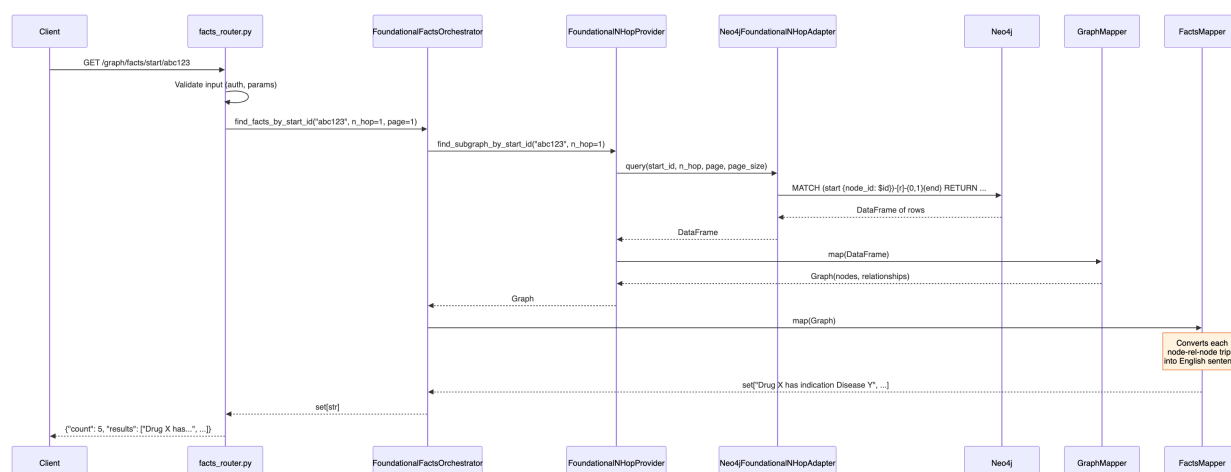
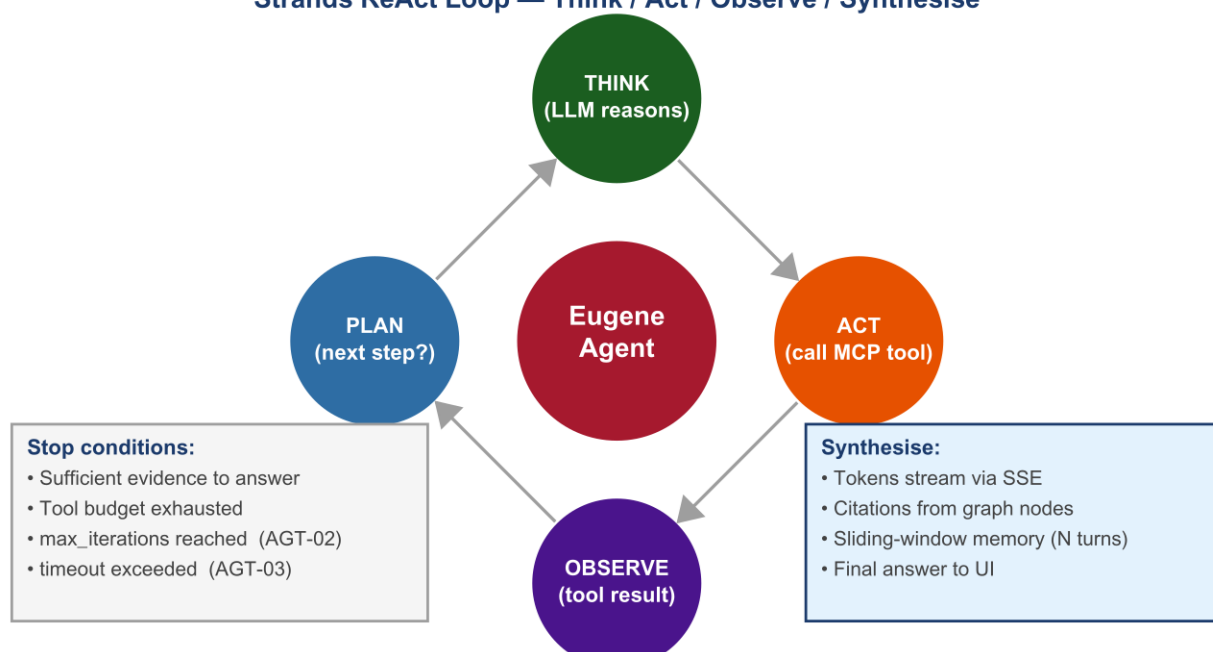


Figure (Mermaid). Request sequence for GET /graph/facts/start/{id}, from DEVELOPER\_GUIDE.md §4.

## 5.2 ReAct Loop Inside the Agent

## Strands ReAct Loop — Think / Act / Observe / Synthesise



Inside the Agent Backend, the Strands framework drives a ReAct loop: the LLM reasons (Think), invokes an MCP tool (Act), receives a structured result (Observe), and repeats until it has sufficient evidence or its tool budget is exhausted, at which point it synthesises a streaming answer.

The developer guide expands the same loop into a per-tool flowchart showing how the agent reasons, calls an MCP tool, returns to reason, and ultimately streams the final answer back to the UI:

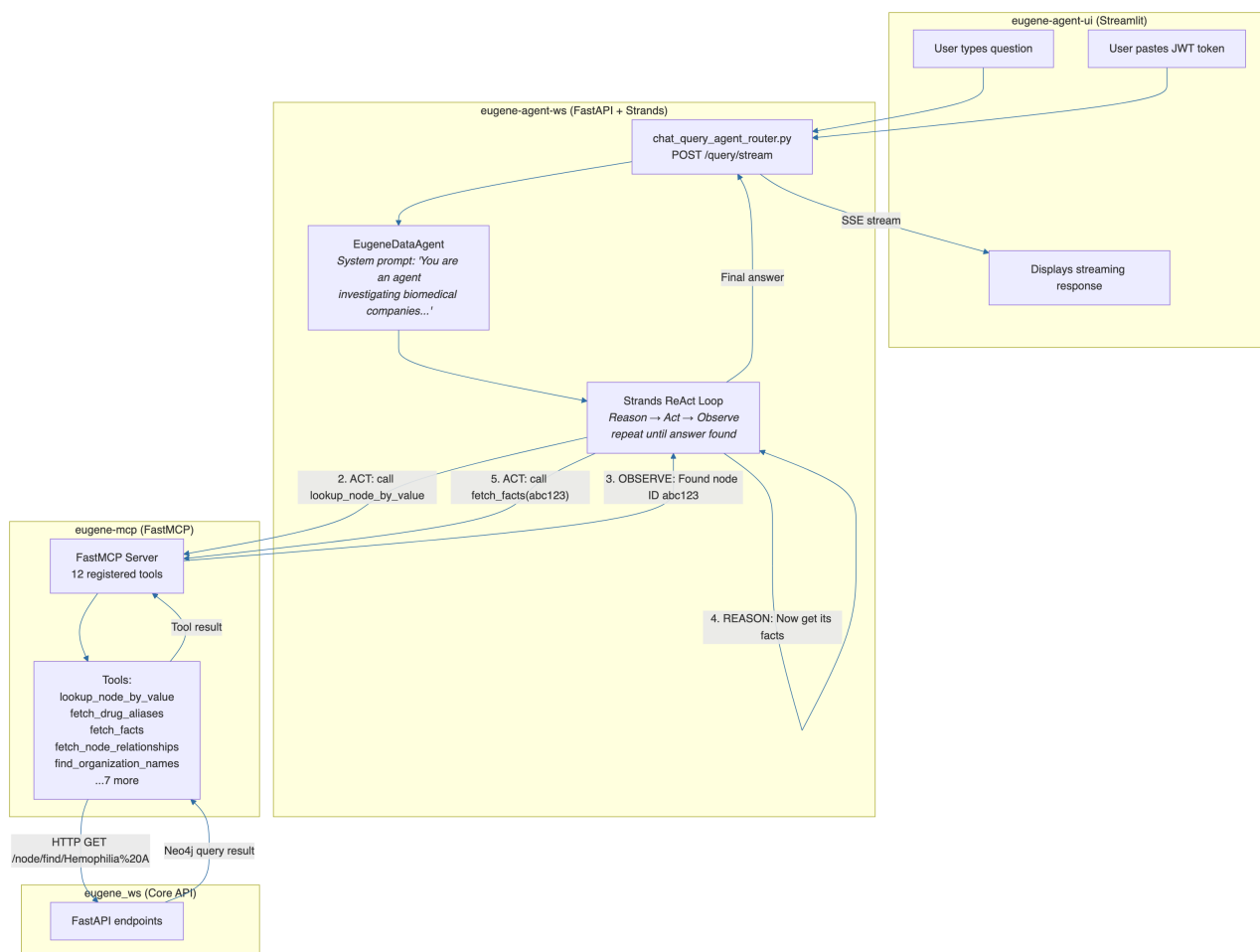
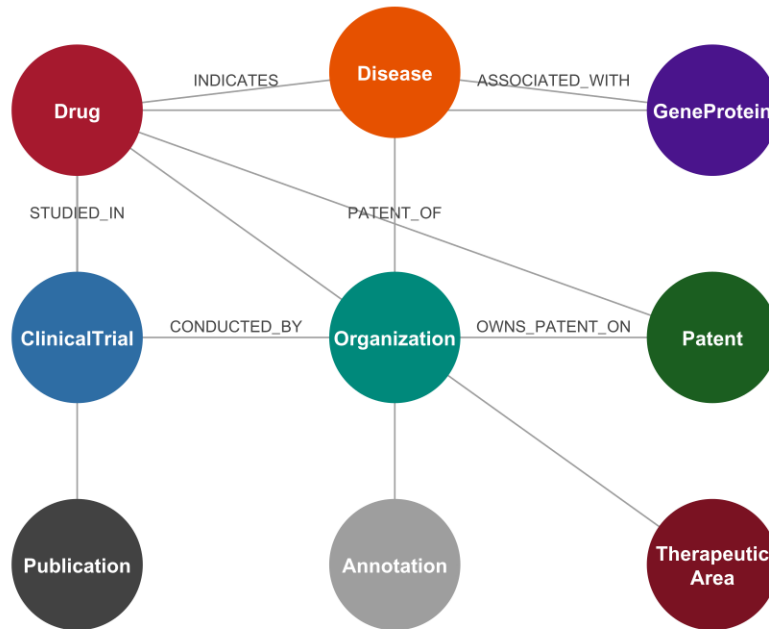


Figure (Mermaid). Agent layer with Strands ReAct loop, from DEVELOPER\_GUIDE.md §7.

## 6. Data Model

### 6.1 Graph Schema

Eugene Knowledge Graph — Schema (9 node labels, principal edges)



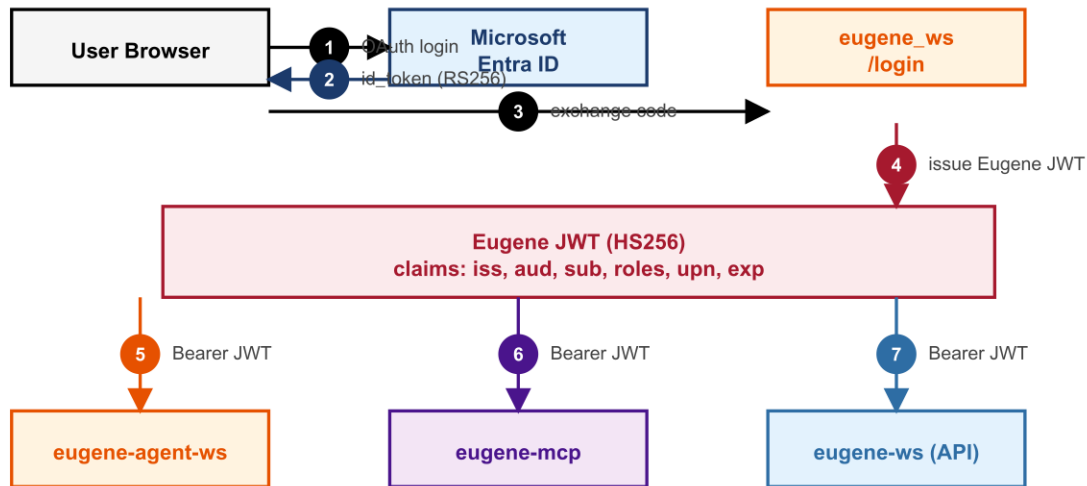
*Approx. 484k nodes • 21M edges • Bolt protocol • parameterised Cypher only*

The graph carries nine principal node labels — Drug, Disease, GeneProtein, ClinicalTrial, Patent, Organization, Publication, Annotation, and Therapeutic Area. Principal relationship types are TARGETS, INDICATES, ASSOCIATED\_WITH, OWNS\_PATENT\_ON, CONDUCTED\_BY, STUDIES, PUBLISHED\_IN, SYNONYM\_OF, and PARENT\_OF. All Cypher queries use parameterised binding; no string interpolation appears anywhere in the source.

## 7. Authentication & Authorisation

### 7.1 Two-Layer Identity Model

## Authentication Flow — Entra ID → Eugene JWT → Service Mesh



Local mode (ENVIRONMENT=local) bypasses steps 1–3 and issues a development JWT directly.

The developer guide describes the same identity flow as a sequence diagram, showing both the local-mode token issue path and the per-request validation path:

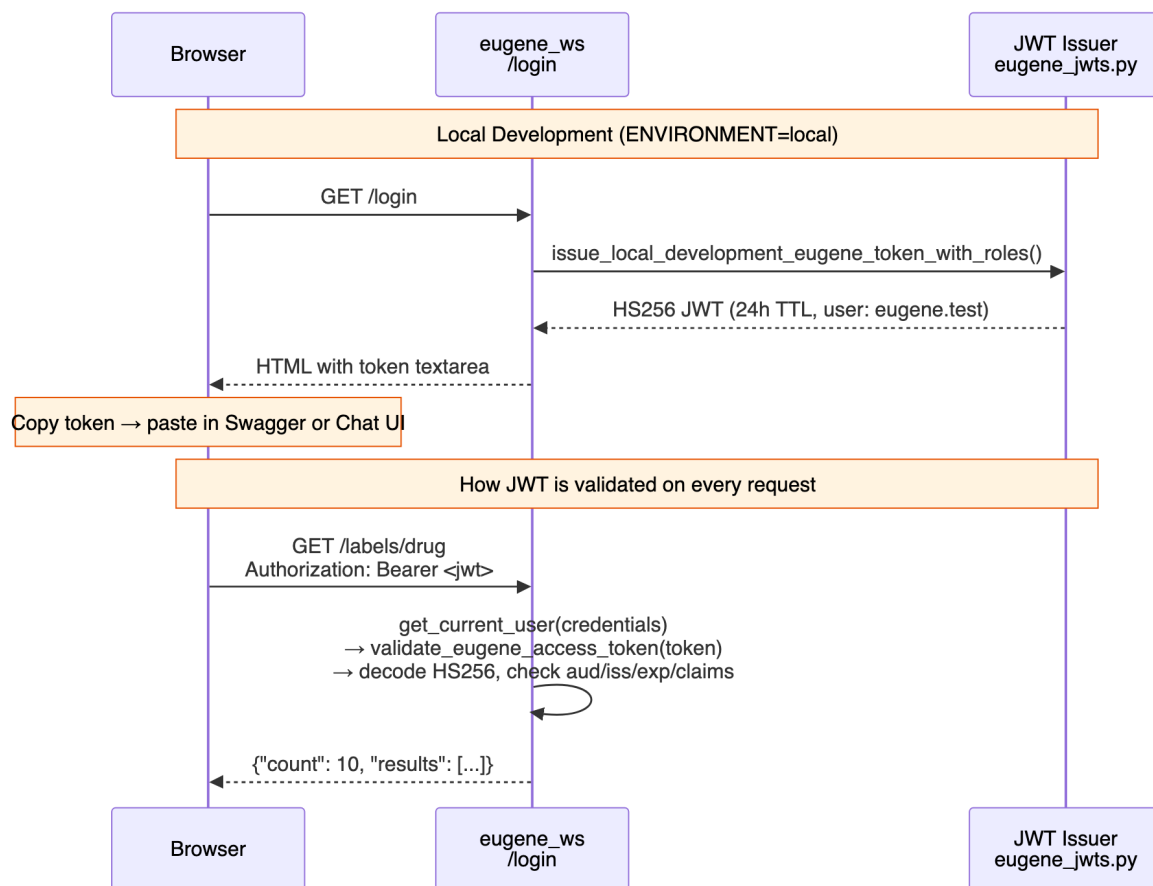


Figure (Mermaid). JWT login and per-request validation, from DEVELOPER\_GUIDE.md §11.

Eugene operates a two-layer identity model. Layer 1 is enterprise SSO via Microsoft Entra ID using OAuth 2.0 Authorization Code flow. Layer 2 is an Eugene-issued HS256 JWT that propagates between services. After Entra ID returns an ID token, the Core API issues an Eugene JWT (24-hour TTL) with claims iss = https://eugene.ai.cslg1.cslg.net/{tenant\_id}, aud = api://eugene/{client\_id}, sub = user UPN, and roles. The

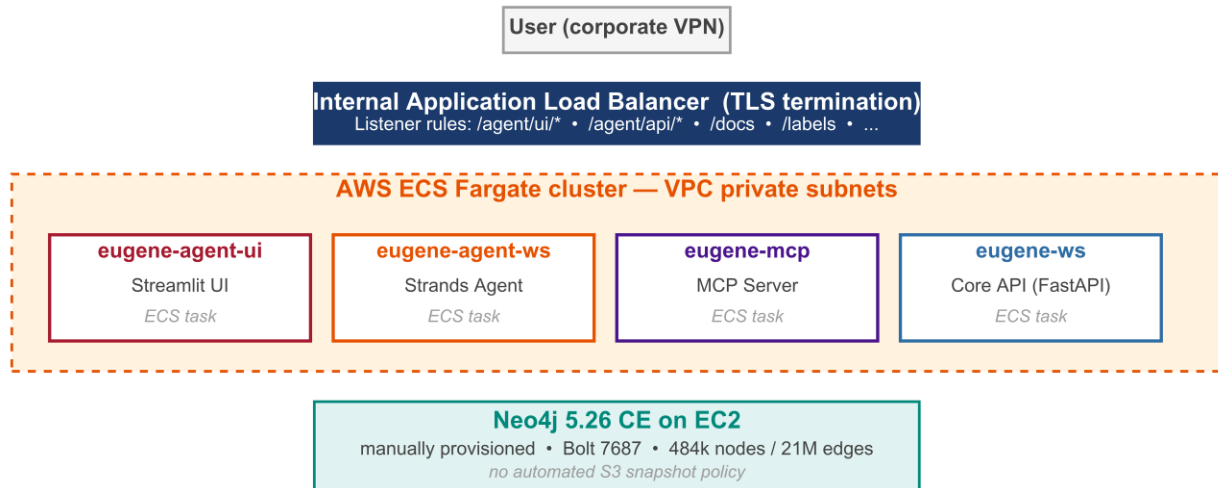
same JWT travels with every request from browser to agent, MCP, and Core API; every tier validates signature, audience, issuer, and expiry.

## 7.2 RBAC Status

Row-level RBAC is not implemented today. The roles claim is present in the JWT but is not enforced inside Cypher. Until Q3 2026, the platform exposes only public data; introduction of patient-derived or commercially sensitive data is gated on RBAC implementation.

## 8. AWS Deployment Topology

### AWS Production Deployment Topology



### 8.1 Environment Matrix

| Attribute | Local            | Difflabs (Dev)  | AIA (Production) |
|-----------|------------------|-----------------|------------------|
| Platform  | Docker Compose   | AWS ECS Fargate | AWS ECS Fargate  |
| Region    | —                | us-east-1       | eu-central-1     |
| Account   | —                | 087084717211    | 010928221940     |
| Auth      | Local JWT bypass | Entra ID        | Entra ID         |
| Neo4j     | Container        | ECS task        | EC2 (manual)     |
| Secrets   | docker.env       | Secrets Manager | Secrets Manager  |
| TLS       | Disabled         | ALB-terminated  | ALB-terminated   |
| IaC root  | —                | difflabs/iac/   | infrastructure/  |

### 8.2 CI / CD

Code pushed to the repository triggers pre-commit hooks (Black, Ruff, isort), then GitHub Actions runs unit tests, integration tests against a Neo4j testcontainer, builds the Docker image, pushes to ECR, and runs Terraform plan. Production deployments require a manual approval gate before Terraform apply triggers a rolling ECS update. The developer guide records the same pipeline as a Mermaid flowchart:



Figure (Mermaid). CI / CD pipeline, from DEVELOPER\_GUIDE.md §13.

## SECTION 4 — GAPS IDENTIFIED

### 9. Audit Outcome

The Stage 1 assessment identified twenty-one findings across security, agent framework, database, and observability domains. Two are CRITICAL, ten are HIGH, and nine are MEDIUM severity. Six positive findings recognise the platform's architectural strengths and are preserved through the recommended remediation programme. Estimated remediation effort to close every CRITICAL and HIGH item is approximately eight weeks of focused engineering work, parallelisable across two to three engineers.

### 10. Critical Findings

The two CRITICAL findings dominate the first week of the recommended Phase 1 hardening. Both are fixable in a day or less; both should land before any further functional work is committed.

#### [CRITICAL] SEC-01 — SSL verification disabled in MCP client

**Description.** verify=False on httpx; service-to-service traffic vulnerable to MITM.

**Remediation.** Set verify=True; pass cert path for self-signed dev certs.

**Effort.** 0.5 day

#### [CRITICAL] AGT-01 — Shared SlidingWindowConversationManager — race condition

**Description.** Single manager shared across agent instances corrupts concurrent sessions.

**Remediation.** Instantiate the manager per conversation, not per service.

**Effort.** 1 day

### 11. High-Severity Findings

Ten HIGH-severity findings span the agent, security, database, and observability domains. Eight are quick wins (half a day to one day each); two require multi-day effort (rate limiting, structured logging). Together they remove the platform's most acute production risks.

#### [HIGH] AGT-02 — Unbounded ReAct loop

**Description.** No max\_iterations on agent; can cycle indefinitely on tool calls.

**Remediation.** Set max\_iterations=25 in Agent constructor.

**Effort.** 0.5 day

#### [HIGH] AGT-03 — No agent execution timeout

**Description.** Streaming connections held until client times out (~120s).

**Remediation.** Wrap stream\_async() in asyncio.wait\_for(timeout=300).

**Effort.** 0.5 day

#### [HIGH] AGT-04 — Anthropic LLM client timeout not configured

**Description.** OpenAI client has 60s timeout; Anthropic has none.  
**Remediation.** Pass timeout=60.0 to the Anthropic model configuration.  
**Effort.** 0.5 day

#### [HIGH] AGT-05 — python\_repl tool always loaded

**Description.** LLM has access to arbitrary code execution by default.  
**Remediation.** Make python\_repl opt-in; disable in production.  
**Effort.** 0.5 day

#### [HIGH] SEC-02 — Allowlist raises generic Exception

**Description.** Causes HTTP 500 instead of 403; leaks stack to clients.  
**Remediation.** Raise HTTPException(status\_code=403, detail='Access denied').  
**Effort.** 0.5 day

#### [HIGH] SEC-03 — No rate limiting on any endpoint

**Description.** Unlimited concurrent requests; DoS / cost exposure.  
**Remediation.** Add slowapi: 100 req/min per IP; 10 sessions per user.  
**Effort.** 2 days

#### [HIGH] DB-01 — No Neo4j query timeout

**Description.** Driver initialised without socket\_keepalive or transaction timeout.  
**Remediation.** Set connection\_timeout=30 and max\_transaction\_retry\_time=30.  
**Effort.** 1 day

#### [HIGH] OBS-01 — No structured JSON logging

**Description.** Plain-text logging incompatible with CloudWatch Logs Insights.  
**Remediation.** Adopt python-json-logger across all services.  
**Effort.** 2 days

#### [HIGH] OBS-02 — No metrics export

**Description.** No Prometheus or CloudWatch metrics for application signals.  
**Remediation.** Add Prometheus client + scrape config; or CloudWatch EMF.  
**Effort.** 3 days

#### [HIGH] OBS-03 — No distributed tracing

**Description.** No correlation across services; cross-service debugging is forensic.  
**Remediation.** Integrate OpenTelemetry SDK + AWS X-Ray exporter.  
**Effort.** 5 days

#### [HIGH] OBS-04 — Shallow /health endpoints

**Description.** Returns OK without checking dependencies.  
**Remediation.** Add deep checks for Neo4j, MCP, LLM availability.  
**Effort.** 1 day

## 12. Medium-Severity Findings

Nine MEDIUM-severity findings cover token revocation, PII / PHI filtering, connection-pool sizing, deep-pagination caps, file-session robustness, graph statistics, and graph-data freshness. They do not block limited rollout, but they are required for a 100-concurrent-user posture and for the regulatory expectations that follow any introduction of patient-derived or commercially sensitive data.



#### [MEDIUM] SEC-04 — No token revocation

**Description.** No /logout; compromised JWT remains valid until exp.  
**Remediation.** Redis-backed denylist; expose /auth/logout endpoint.  
**Effort.** 3 days

#### [MEDIUM] SEC-05 — PII / PHI not filtered

**Description.** No redaction in prompts or responses; regulatory risk.  
**Remediation.** AWS Comprehend PII detection middleware on prompts and responses.  
**Effort.** 5 days

#### [MEDIUM] DB-02 — No explicit Neo4j connection-pool sizing

**Description.** Pool defaults limit concurrent throughput.  
**Remediation.** Tune pool\_size against load tests.  
**Effort.** 0.5 day

#### [MEDIUM] DB-03 — No deep-pagination offset cap

**Description.** Unbounded OFFSET enables slow scans.  
**Remediation.** Reject OFFSET > 100,000 in pagination utility.  
**Effort.** 0.5 day

#### [MEDIUM] AGT-06 — FileSessionManager — no recovery, no cleanup

**Description.** Disk-based sessions; concurrent writes contend; no purge.  
**Remediation.** fcntl locking now; migrate to DynamoDB / Redis.  
**Effort.** 2 days

#### [MEDIUM] GRAPH-01 — No graph statistics dashboard

**Description.** Node / edge counts and growth rate not surfaced.  
**Remediation.** Add /stats/graph endpoint and a small dashboard.  
**Effort.** 1 day

#### [MEDIUM] GRAPH-02 — Graph data freshness not tracked

**Description.** No last\_updated on nodes; staleness not detectable.  
**Remediation.** Add last\_updated property; expose in /stats.  
**Effort.** 2 days

## 13. Operational, Data, and Experience Gaps

### 13.1 Data Loading Gaps

■ *USPTO patents count is currently zero in production (data deleted, never reloaded). PubMed count is currently zero (partial ingestion abandoned). Therapeutic-area subgroup counts are currently zero. These gaps are documented in the maintenance runbook and addressed in the data-readiness backlog.*

### 13.2 Operational Gaps

- JWT signing-secret rotation is manual; no scheduled rotation.
- Neo4j SSL certificate rotation is manual; expires near year-end.
- Production Neo4j is provisioned manually on EC2; not yet Terraform-managed.
- No automated S3 snapshot policy for Neo4j data.
- No LLM cost instrumentation; per-query token consumption is not tracked.
- API canaries cover only five of twenty Core API endpoints.

### 13.3 Experience Gaps in the Streamlit UI

- No conversation export (JSON, Markdown, PDF).
- No citation panel or evidence drilldown.
- No graph visualisation; the agent's reasoning trail is opaque.
- No tool-call timeline or agent-decision lineage.
- Limited accessibility primitives; no keyboard shortcuts.
- Poor mobile responsiveness.
- No light / dark theme; no system-prompt templates.

## 14. Architectural Strengths to Preserve

The audit also surfaced six structural strengths. They are listed here so that the re-platforming work explicitly retains and amplifies them rather than refactoring them away.

### [POSITIVE] POS — Clean DDD + Hexagonal Architecture

**Description.** Domain-driven layering with explicit ports and adapters; testable, swappable, well-named.

**Remediation.** —

**Effort.** —

### [POSITIVE] POS — Comprehensive Knowledge Graph Ontology

**Description.** Forty-four node types and thirty-one relationship types unify USPTO, PubMed, trials, drugs, deals.

**Remediation.** —

**Effort.** —

### [POSITIVE] POS — Dual LLM Provider with Auto-Detect

**Description.** Anthropic / OpenAI hot-swap on a single env switch; resilience and cost flexibility.

**Remediation.** —

**Effort.** —

### [POSITIVE] POS — Stateless MCP Server

**Description.** FastMCP with stateless\_http=True; trivially horizontally scalable.

**Remediation.** —

**Effort.** —

### [POSITIVE] POS — Streaming SSE Architecture

**Description.** Token-by-token UX; first-token latency drives perceived speed.

**Remediation.** —

**Effort.** —

### [POSITIVE] POS — Parameterised Cypher; Two-Hop Cap

**Description.** No injection risk; bounded query cost prevents pathological queries.

**Remediation.** —

**Effort.** —

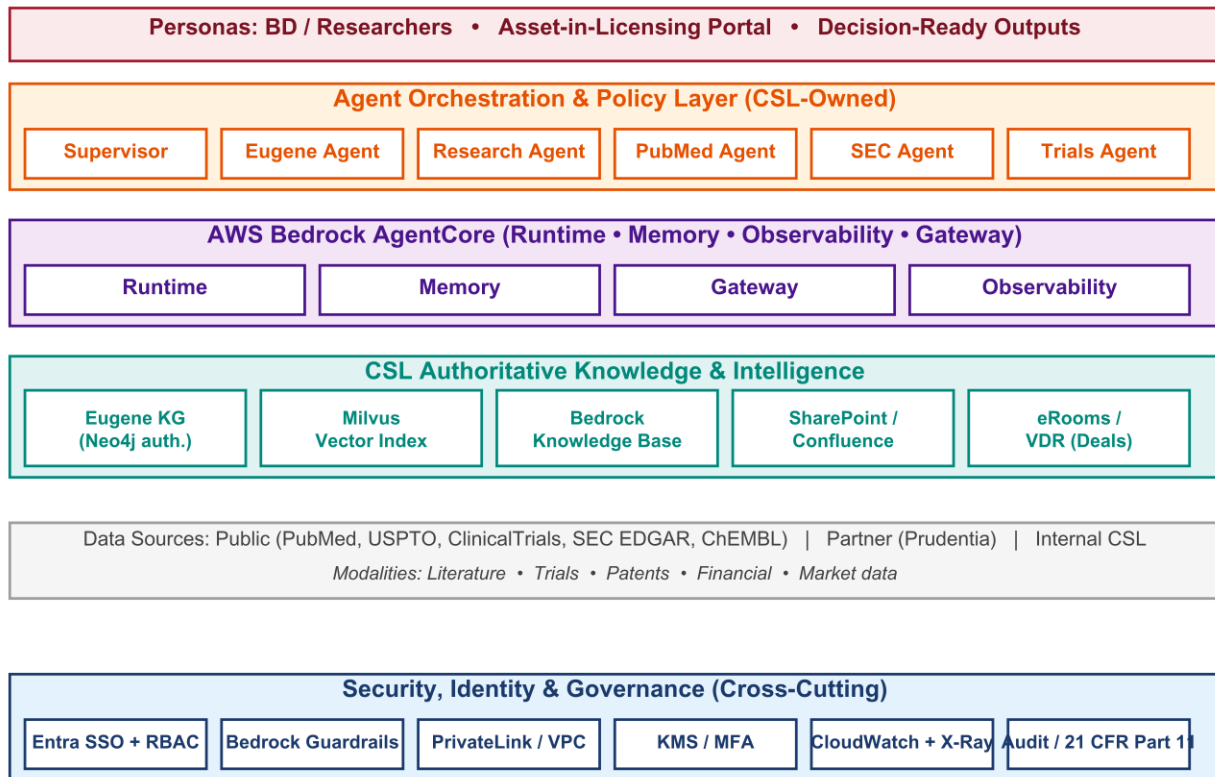
## SECTION 5 — TECHNICAL RECOMMENDATIONS

## 15. Next-Generation Target Architecture

## 15.1 Headline Picture

The next-generation architecture (CSL\_Arch\_nextGen.pdf) re-imagines Eugene as one specialist agent within a CSL-owned orchestration layer running on AWS Bedrock AgentCore. The diagram below summarises the target state.

### CSL Next-Generation Agentic Platform (Target State)



## 15.2 Research Agent — End-to-End Reference Architecture

The diagram below — sourced from docs/CSL\_Arch.drawio.svg — renders the complete CSL-owned Research Agent reference architecture in a single view. It maps every layer of the next-generation platform from the BD / Researcher persona through the CSL-owned orchestration and specialist agents, into AWS Bedrock AgentCore (Runtime, Memory, Observability) and the AgentCore Gateway, across the authoritative knowledge tier (Eugene Knowledge Graph plus dual storage with Milvus indexes and Neo4j embeddings), the CSL data tier (internal tacit knowledge in SharePoint, Confluence, deal memos and IC decisions; controlled-access eRooms / VDRs; external and partner feeds including the Prudentia Sciences platform; and public / licensed sources covering literature, trials, patents, financial and market data), and the cross-cutting Security, Identity and Governance services (Entra SSO, RBAC, security and compliance controls, hybrid cloud / AI runtime, AWS PrivateLink, CloudWatch, X-Ray, KMS, MFA, and Bedrock Guardrails).

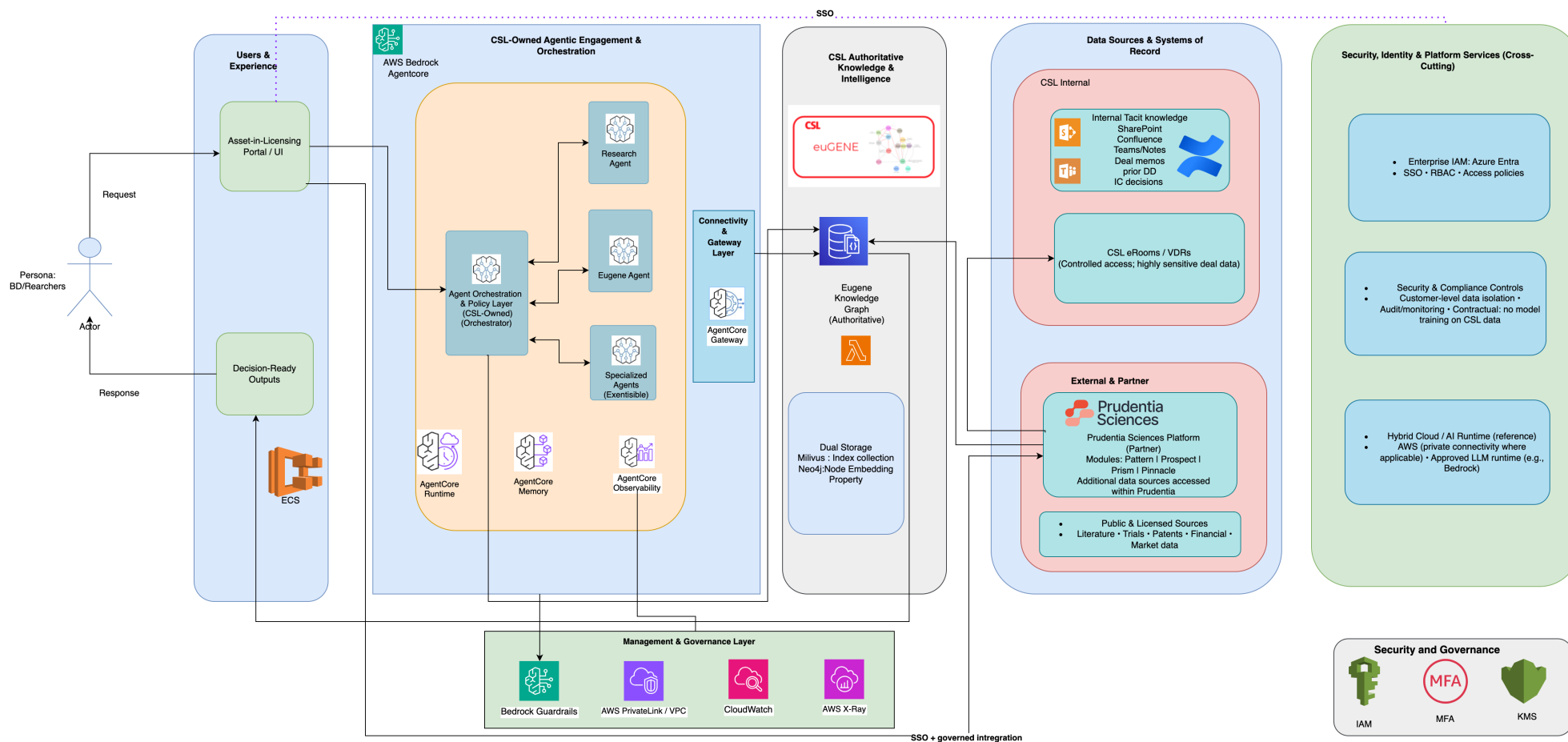


Figure. CSL Research Agent — next-generation reference architecture (rendered from docs/CSL\_Arch.drawio.svg).

Three properties of this reference are worth highlighting for stakeholders. First, the Research Agent is one specialist within a CSL-owned orchestration layer — the Eugene Knowledge Graph remains the authoritative structured source, but the architecture admits additional specialists (PubMed, SEC, ClinicalTrials, ChEMBL, and future modalities) without re-platforming. Second, all service-to-service connectivity is mediated through the AgentCore Gateway — a single audit boundary for every agent action. Third, governance is unified: Bedrock Guardrails, customer-level data isolation, and contractual no-training commitments apply uniformly across every specialist, regardless of which underlying LLM is selected.

### 15.3 Key Shifts From the Current State

- Single agent → Supervisor + five specialists (Eugene, Research, PubMed, SEC, ClinicalTrials, Synthesis). Domain-optimised models and tool sets.
- External LLM APIs → AWS Bedrock as the primary LLM runtime, with Anthropic as fallback. Data residency stays inside the AWS partition.
- File-based sessions → AgentCore Memory (managed) or DynamoDB / ElastiCache. Durability and horizontal scale.
- Graph-only retrieval → Graph + Bedrock Knowledge Base + Milvus + SharePoint / Confluence / eRooms. Eugene becomes the authoritative structured source; unstructured CSL knowledge is retrieved via Bedrock KB; deal-room artefacts via controlled-access policies.
- Basic logging → AgentCore Observability + CloudWatch + X-Ray. Distributed tracing, structured logs, and metrics with unified alerting.
- 12 MCP tools → 16+ tools spanning PubMed, SEC EDGAR, ClinicalTrials, ChEMBL, plus the existing Eugene tool set.
- Implicit governance → Bedrock Guardrails, AWS PrivateLink, KMS, MFA, and 21 CFR Part 11-grade audit logging baked into every agent action.

### 15.4 Specialist Agent Topology

| Agent           | Responsibility                     | Tools                           | Model         |
|-----------------|------------------------------------|---------------------------------|---------------|
| Supervisor      | Intent routing, response synthesis | classify_intent, route_to_agent | Claude Sonnet |
| Eugene Agent    | KG queries, relationship traversal | 12 Eugene MCP tools             | Claude Sonnet |
| Research Agent  | Free-text research synthesis       | search_kb, summarise            | Claude Sonnet |
| PubMed Agent    | Literature, citations              | search_pubmed, get_article      | GPT-4.1-mini  |
| SEC Agent       | Filings, M&A, financials           | search_sec, get_filing          | GPT-4.1-mini  |
| Trials Agent    | Trial discovery, protocol          | search_trials, get_protocol     | GPT-4.1-mini  |
| Synthesis Agent | Cross-source aggregation           | All read tools                  | Claude Sonnet |

## 16. Four-Phase Roadmap

### 16.1 Phasing

| Phase               | Weeks | Outcomes                                                                                     |
|---------------------|-------|----------------------------------------------------------------------------------------------|
| 1. Foundation Fixes | 1 – 4 | All CRITICAL + HIGH findings resolved; structured logging; deep /health; rate limiting live. |

| Phase                      | Weeks  | Outcomes                                                                                                                       |
|----------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------|
| 2. Session & Observability | 5 – 8  | FileSessionManager replaced; OpenTelemetry + X-Ray; PII / PHI middleware; Neo4j S3 backups; first UI cohort delivered.         |
| 3. Multi-Agent & New Tools | 9 – 12 | Supervisor + specialist topology; PubMed / SEC / Trials agents; Bedrock as primary LLM; UI cohort 2 delivered.                 |
| 4. Vector Search & Audit   | 13 +   | Bedrock Knowledge Base for unstructured KB; freshness tracking; explainability lineage API; UI cohort 3 + accessibility audit. |

## 16.2 Quick Wins versus Strategic Items

| Category                 | Findings                                                                                      | Effort    |
|--------------------------|-----------------------------------------------------------------------------------------------|-----------|
| Quick Wins (0.5 – 1 day) | SEC-01, AGT-01, AGT-02, AGT-03, AGT-04, AGT-05, SEC-02, DB-01, DB-02, DB-03, OBS-04, GRAPH-01 | ≈ 8 days  |
| Strategic (2 – 5 days)   | SEC-03, SEC-04, SEC-05, OBS-01, OBS-02, OBS-03, AGT-06, GRAPH-02                              | ≈ 23 days |

## 17. User-Experience Enhancement Programme

Fifteen concrete enhancements are sized for delivery inside Phases 2 and 3. They retain the strengths of the current Streamlit chat surface and complete the Next.js v2 implementation that is already underway.

- Conversation snapshot export (JSON / Markdown / PDF).
- Citation and evidence drilldown panel with lineage API.
- Graph mini-map and zoom controls on the canvas.
- Tool-call timeline rendered as a per-turn Gantt.
- Role-based tool hints (read-only / admin-only / PII-risky).
- System-prompt templates: Default Biomedical, Patent Analyst, Clinical Trialist, Competitive Intelligence.
- Evidence drilldown with graph expansion to show cited paths.
- Share and collaborative export with read-only short links.
- Accessibility overhaul (WCAG 2.2 AA).
- Keyboard shortcuts (Cmd+K, Cmd+N, Cmd+E, Cmd+?).
- Light / dark theme toggle synchronised across canvas.
- Mobile-responsive layout (tablet and phone).
- Search history with autocomplete.
- Agent-decision lineage UI ("Why?" button per turn).
- Node-property hover tooltips on the graph canvas.

## 18. Non-Functional Targets

| Dimension                  | Target       |
|----------------------------|--------------|
| Latency P50 (simple query) | < 2 seconds  |
| Latency P95 (multi-hop)    | < 10 seconds |

| Dimension              | Target                                      |
|------------------------|---------------------------------------------|
| Concurrent agents      | 100 sustained                               |
| Throughput per service | 1,000 req / minute                          |
| Availability           | 99.9 % (multi-AZ)                           |
| Data residency         | Within AWS partition end-to-end             |
| Audit                  | Per-decision lineage; 21 CFR Part 11-ready  |
| Security posture       | Zero trust; gateway + service authn / authz |
| Accessibility          | WCAG 2.2 AA                                 |
| Mobile experience      | Functional on tablet and phone              |

## 19. Programme Risks and Mitigations

| Risk                                                     | Likelihood | Impact   | Mitigation                                            |
|----------------------------------------------------------|------------|----------|-------------------------------------------------------|
| LLM API rate limit under concurrent load                 | Medium     | High     | Queueing; exponential backoff; cache frequent queries |
| Bedrock service availability in eu-central-1 for Phase 3 | Low        | High     | Cross-region fallback or Anthropic                    |
| SEC EDGAR / ChEMBL licensing beyond fair-use             | Medium     | Medium   | Procurement workstream in parallel with Phase 3       |
| Change management on persona workflows                   | Medium     | Medium   | Internal champions; phased rollout                    |
| JWT secret compromise                                    | Low        | Critical | Rotation policy; future RS256                         |
| Neo4j data loss                                          | Low        | Critical | S3 snapshot policy in Phase 2                         |
| Dependency CVE in 175-package requirements               | Medium     | Medium   | pip-audit in CI; Dependabot                           |

## 20. Recommendation

We recommend proceeding to the four-phase remediation and next-generation roadmap, with the eight-week Phase 1 + Phase 2 security and operational backbone as the first commitment. The platform's architectural strengths — clean DDD layering, comprehensive ontology, dual-LLM resilience, stateless MCP, streaming SSE, and parameterised Cypher — mean that hardening work compounds rather than displaces what already works.

With the recommended investment, Eugene transitions from a credible first-generation internal tool to a strategic CSL asset that materially shortens BD diligence cycles by synthesising structured and unstructured intelligence in a single conversational, governed, auditable surface.

## APPENDICES

### Appendix A — Source Documents

- EUGENE\_PROJECT\_VALIDATION\_DOCUMENT.md (v1.0, March 2026).

