

euGENE

Biomedical Knowledge Graph & Agentic AI Platform

PROJECT VALIDATION DOCUMENT

Version 1.0 | March 2026 Prepared for CSL Behring Confidential

Table of Contents

1. Executive Summary
2. Background & Context
3. Solution Overview
4. Functional Requirements
5. Non-Functional Requirements
6. Architecture
7. Tooling & Skills Design
8. Knowledge & Data Strategy
9. Security & Access Control
10. Prompt Design
11. Workflow Orchestration
12. Deployment Model
13. Monitoring & Observability
14. Testing Strategy
15. Risks & Mitigations
16. Open Questions

1. Executive Summary

euGENE is a biomedical knowledge graph and agentic AI platform developed for CSL Behring to enable competitive intelligence across drugs, diseases, genes, proteins, patents, clinical trials, and organizations. The system integrates data from USPTO, PubMed, ClinicalTrials.gov, DrugBank, and internal CSL sources into a Neo4j graph database containing 129,375 nodes and 4,050,249 relationships.

The platform provides both a structured REST API for programmatic access and a conversational AI interface powered by LLM agents (Claude and GPT models) that can query the knowledge graph using natural language. The system follows a five-tier microservices architecture: Neo4j database, Core API (FastAPI), MCP

Tool Server (FastMCP), Agent Backend (Strands framework), and Chat UI (Streamlit).

Authentication is handled through Microsoft Entra ID (enterprise SSO) with a dual-layer JWT token system. The platform is deployed on AWS using Terraform infrastructure-as-code, with environments in both us-east-1 (difflabs/dev) and eu-central-1 (AIA/production). CI/CD is managed through GitLab pipelines.

2. Background & Context

2.1 Business Need

CSL Behring requires competitive intelligence capabilities to assess drug pipelines, disease research, patent landscapes, and organizational partnerships across the pharmaceutical industry. Existing tools lack the ability to traverse complex relationships between biomedical entities and provide actionable insights through natural language queries.

2.2 Project Scope

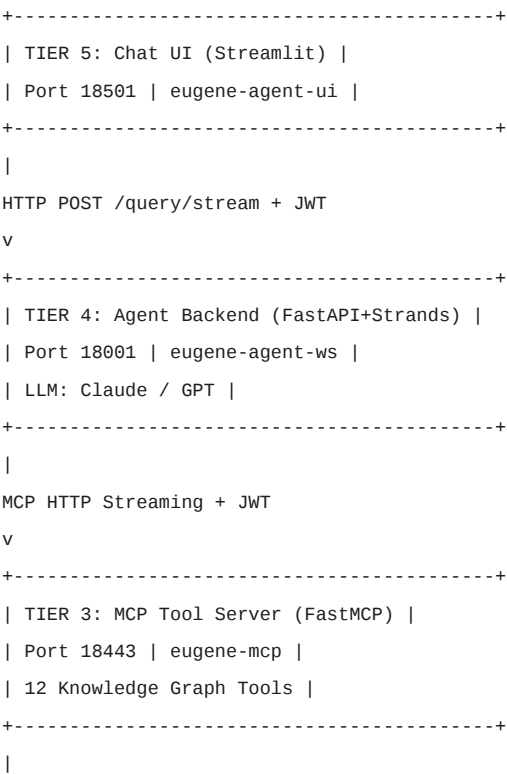
- Build a knowledge graph from public biomedical data sources (DrugBank, ClinicalTrials.gov, USPTO, PubMed)
- Expose graph data via REST API with authentication and role-based access
- Provide an AI-powered conversational interface for natural language querying
- Deploy on AWS with infrastructure-as-code and CI/CD automation
- Support competitive analysis: drug pipelines, organizational assets, patent landscapes

2.3 Stakeholders

2.4 Technology Decisions

3. Solution Overview

3.1 System Architecture (Five-Tier)



```

HTTP REST + JWT
v
+-----+
| TIER 2: Core API (FastAPI) |
| Port 18000 | eugene-ws |
| 38 REST Endpoints |
+-----+
|
| Bolt Protocol (TLS)
v
+-----+
| TIER 1: Neo4j Graph Database |
| Port 17687 | neo4j:5.26.9 |
| 129K nodes, 4M relationships |
+-----+

```

3.2 Service Inventory

3.3 Code Architecture Pattern

The codebase follows Domain-Driven Design (DDD) with Hexagonal Architecture. Each domain module (foundation, organization, patent, pubmed, tpp, graph) has a consistent internal structure:

```

<domain>/
+-- model/ # Pydantic data models & enums
+-- provider/ # Business logic & orchestration
+-- mapper/ # Data transformation (DataFrame -> Model -> Response)
+-- router/ # FastAPI endpoint definitions
+-- conf/ # Dependency injection (factory functions)
+-- infra/ # Database adapters (Neo4j)
+-- writer/ # Persistence & checkpointing
+-- load/ # Load from disk/checkpoints

```

Dependency injection is manual via conf.py factory functions. There is no DI framework; each domain's conf.py instantiates adapters, providers, mappers, and routers with explicit wiring.

4. Functional Requirements

4.1 Core API Capabilities

4.2 Agent Capabilities (MCP Tools)

4.3 Chat Interface Features

- Real-time streaming responses via Server-Sent Events (SSE)
- Conversation history with session persistence
- Tool selection checkboxes (eugene, pubmed, http)
- OAuth token input for authenticated access
- Suggestion pills for common queries (e.g., drug aliases, org assets)
- New conversation management with UUID tracking

5. Non-Functional Requirements

6. Architecture

6.1 Data Flow

The system implements a layered data flow from user query to database and back:

```
User Query (natural language)
-> Streamlit UI (eugene_agent_ui)
-> POST /agent/api/query/stream with JWT
-> Agent Backend (eugene_agent_ws)
-> Strands Agent with ReAct pattern
-> MCP Tool Calls (eugene_mcp)
-> HTTP REST calls to Core API (eugene_ws)
-> Neo4j Cypher queries
<- Graph results (nodes, relationships)
<- Structured JSON responses
<- Tool results aggregated
<- LLM generates natural language answer
<- SSE stream chunks
<- Real-time display with cursor
<- Final answer displayed in chat
```

6.2 Authentication Flow

PRODUCTION FLOW:

```
User -> GET /login
-> Redirect to Microsoft Entra ID
-> User authenticates with corporate SSO
-> Entra returns ID token (RS256, 0IDC)
-> Backend validates via JWKS endpoint (8h cache)
-> Backend issues Eugene JWT (HS256, 24h TTL)
-> Token includes: tid, sub, upn, roles
```

LOCAL DEV FLOW:

```
User -> GET /login (ENVIRONMENT=local)
-> Immediate Eugene JWT issued
-> User: eugene.test@cslhering.com
-> Roles: [user.public.read]
```

TOKEN FORWARDING:

```
UI -> Agent WS (validates JWT + checks allowlist)
-> MCP Server (JWTVerifier middleware)
-> Core API (get_current_user dependency)
```

6.3 Neo4j Graph Schema

6.4 Relationship Types

7. Tooling & Skills Design

7.1 MCP Server Architecture

The MCP (Model Context Protocol) server exposes knowledge graph capabilities as callable tools for LLM agents. Built with FastMCP framework, it provides 12 tools organized into 5 tool classes:

7.2 Tool Implementation Pattern

Each MCP tool follows a consistent pattern:

- Extracts JWT token from MCP request context
- Makes async HTTP call to Core API with token forwarding
- Handles pagination automatically (up to 10-50 pages)
- Returns structured JSON or formatted text
- Timeout: 30 seconds per HTTP request
- Error handling: Returns None on failure with logging

7.3 Local Agent Tools

The agent backend also provides local (non-MCP) tools:

- calculator - Mathematical computations
- current_time - Current timestamp retrieval
- python_repl - Python code execution sandbox
- http_request - General HTTP requests (optional, on-demand)

7.4 Future Tools (Planned)

- PUBMED - Direct PubMed article search and retrieval
- CLINICAL_TRIALS - ClinicalTrials.gov API integration
- ChEMBL - ChEMBL chemical database queries
- BIORXIV - BioRxiv preprint search

8. Knowledge & Data Strategy

8.1 Data Sources

8.2 Data Ingestion Pipeline

Data loading follows a three-phase approach:

- Phase 1: Foundational bulk import using neo4j-admin (nodes.csv + edges_dedup.csv) - 129K nodes, 4M relationships in 7.6 seconds
- Phase 2: Feature enrichment via APOC batch loading - Disease features (44K, ~53 min), Drug features (8K, ~9.5 min)
- Phase 3: Supplementary data via Cypher scripts and API downloads (clinical trials, patents, PubMed)

8.3 Embedding Strategy

- Model: minishlab/potion-base-8M (lightweight, fast inference)
- Purpose: Semantic search across drugs, diseases, and documents
- Storage: Neo4j node properties + Milvus vector database (lite mode)
- Pre-caching: Models pre-downloaded from S3 in production containers

8.4 Graph Analytics

- Neo4j Graph Data Science (GDS) plugin for in-memory graph projections
- Node Similarity: Cosine distance between node embeddings
- Community Detection: Leiden algorithm for identifying clusters
- GraphSAGE: Node embedding pipeline (project -> train -> predict)
- Link Prediction: GDS beta pipeline for relationship prediction

9. Security & Access Control

9.1 Authentication Architecture

9.2 Role-Based Access Control

Role mapping is currently static in code (roles.py). Named users from CSL are assigned both public and confidential roles. All other authenticated users receive the default public read role.

9.3 Token Flow Through Services

```
Client -> Authorization: Bearer <EUGENE_JWT>
-> Agent WS: validate JWT + check allowlist
-> MCP Server: JWTVerifier middleware validates
-> Core API: get_current_user() dependency validates
-> Neo4j: Cypher query execution
```

Each service independently validates the JWT.

Token is forwarded (not re-issued) through the chain.

9.4 Security Configuration

9.5 Known Security Items

10. Prompt Design

10.1 Agent System Prompt

The agent backend uses the following system prompt for LLM interactions:

```
"You are an agent tasked with helping investigate biomedical
companies to assess competitive threat and collaboration
opportunities. In our case assets here mean any drug, disease,
patents, clinical trials, intellectual property, or financial
deals the company may have involvement. As an agent follow the
Reason, Act, Observe (ReAct) pattern. You are an agent that
may call tools to retrieve data."
```

10.2 ReAct Pattern

The agent follows the Reason-Act-Observe (ReAct) pattern:

- Reason: Reason: LLM analyzes user query and determines what information is needed
- Act: Act: LLM selects and calls appropriate MCP tools with correct parameters
- Observe: Observe: LLM reviews tool results and determines if more tools are needed
- Repeat until sufficient information gathered, then generate final answer

10.3 Conversation Management

- Sliding window: 10-message history retained per conversation
- Result truncation: Max 2 items per tool response per turn
- Session persistence: FileSessionManager stores conversation state
- Conversation ID: UUID-based tracking across multiple turns

10.4 LLM Configuration

Provider auto-detection: If LLM_PROVIDER is not set, the system prefers Anthropic if ANTHROPIC_API_KEY is available, otherwise falls back to OpenAI.

11. Workflow Orchestration

11.1 Query Execution Flow (Streaming)

1. User submits prompt via Streamlit UI
2. UI sends POST /agent/api/query/stream with JWT + prompt
3. Agent WS validates JWT and checks user allowlist
4. Agent WS initializes EugeneDataAgent:
 - a. Creates LLM model (OpenAI or Anthropic)
 - b. Connects to MCP server with JWT
 - c. Lists available MCP tools
 - d. Initializes local tools (calculator, current_time, etc.)
 - e. Creates Strands Agent with system prompt
5. Agent executes user prompt with ReAct loop:
 - a. LLM reasons about query
 - b. LLM calls MCP tools as needed
 - c. MCP tools call Core API with JWT forwarding
 - d. Core API executes Neo4j Cypher queries
 - e. Results bubble back through chain
 - f. LLM generates next step or final answer
6. Streaming events sent to UI:
 - {"type": "session", "session_id": "..."}
 - {"type": "content", "content": "chunk..."}
 - {"type": "done"}
7. UI displays tokens in real-time with cursor
8. Metrics logged: tokens, execution time, tools used

11.2 Startup Orchestration

Docker Compose manages service startup order via health check dependencies:

- ```
neo4j (healthcheck: wget localhost:7474, 30s startup)
-> eugene_ws (depends: neo4j healthy; healthcheck: urllib localhost:8000/health)
-> eugene_mcp (depends: eugene_ws healthy)
-> eugene_agent_ws (depends: eugene_mcp)
-> eugene_agent_ui (depends: eugene_agent_ws)
```

## 11.3 Data Ingestion Workflow

Phase 1: Bulk Import

- Stop Neo4j service
- Run neo4j-admin database import full
- Input: nodes.csv + edges\_dedup.csv from S3
- Result: 129K nodes, 4M relationships (7.6 seconds)
- Restart Neo4j service

Phase 2: Feature Enrichment

- Run batch\_load\_disease\_features.py (44K features, ~53 min)
- Run batch\_load\_drug\_features.py (8K features, ~9.5 min)
- Uses APOC batch transactions (1000 rows/batch)

Phase 3: Supplementary Data

- Clinical trials: Cypher scripts (schema exists, data cleared)
- Patents: USPTO API download (not yet executed)
- PubMed: API download + APOC loading (not yet executed)

## 12. Deployment Model

### 12.1 Environment Overview

### 12.2 Docker Images

### 12.3 CI/CD Pipeline (GitLab)

Stages:

1. echo - Debug output (commit info)
2. scan - Checkov security scanning (HIGH/CRITICAL fail)
3. docker - Build & push images to ECR
4. terraform-ec2 - Plan EC2 infrastructure
5. apply-ec2 - Apply EC2 (manual approval)
6. terraform-services - Plan ECS services
7. apply-services - Apply ECS (manual approval)

Branch Strategy:

- feature/\* -> Docker build only
- qa -> Full deploy to staging
- main -> Full deploy to production

Authentication:

- GitLab OIDC -> AWS STS assume-role-with-web-identity
- Role: arn:aws:iam::010928221940:role/csl-gitlab-ci-cd-scop-all-branches

### 12.4 Terraform Modules

### 12.5 Production Entrypoint

Production containers use entrypoint.sh scripts that:

- Download .env from S3 (s3://aia-scop-experiment-eugene-data/runtime/.env)
- Export environment variables
- Download pre-cached HuggingFace models from S3
- Start the application server (uvicorn or streamlit)

## 13. Monitoring & Observability

### 13.1 Health Checks

### 13.2 API Canaries

The api-canaries service provides automated monitoring of Core API endpoints:

### 13.3 Database Statistics

The /stats endpoint provides real-time database health metrics:

```
{
 "node_count": "484.3k",
 "relationship_count": "21.4m",
 "drug_count": "4.6k",
 "disease_count": "17.1k",
 "gene_protein_count": "27.7k",
```

```
"clinical_trial_count": "49.3k",
"organization_count": "26.1k"
}
```

## 13.4 Agent Metrics

- Total tokens used per LLM call
- Execution time per agent invocation (seconds, 4 decimal precision)
- Tools used per query (unique tool names)
- Cycle durations (time per ReAct loop iteration)
- Performance decorator (@log\_time) on critical functions

## 13.5 Logging

- Application-level logging at INFO level (configurable)
- Neo4j container logs persisted to Docker volume (eugene\_neo4j\_logs)
- CloudWatch integration for ECS-deployed services
- Structured log format with function names and timing

# 14. Testing Strategy

## 14.1 Test Framework

## 14.2 Test Coverage Areas

## 14.3 Test Execution

```
Run all tests with coverage
pytest

Run specific domain tests
pytest src/organization/

Run with more parallelism
pytest -n 4

Run integration tests only
pytest -m integration

Run serial tests (database-dependent)
pytest -m serial
```

## 14.4 Pre-Commit Hooks

- Terraform validation (difflabs/iac/ modules)
- Black code formatter on src/
- Full pytest suite execution
- Exit code: 0 (all pass) or 1 (any failure blocks commit)

## 14.5 CI/CD Testing

- Checkov security scanning on Terraform (fail on HIGH/CRITICAL)
- Docker image build validation (all 5 services)
- SAST/DAST scanning defined but currently disabled in pipeline

## 14.6 Recommended Additional Testing

- End-to-end: UI -> Agent WS -> MCP -> Core API -> Neo4j flow
- JWT validation edge cases (expired, invalid signature, missing claims)

- MCP tool pagination with large result sets
- Concurrent streaming requests under load
- LLM provider failover (Anthropic -> OpenAI)
- Network timeout and MCP server unavailability handling

## 15. Risks & Mitigations

## 16. Open Questions

--- End of Document ---

Generated from codebase analysis of euGENE Knowledge Graph Project March 2026

| Metric              | Value                                    |
|---------------------|------------------------------------------|
| Total Graph Nodes   | 129,375                                  |
| Total Relationships | 4,050,249                                |
| Core API Endpoints  | 38                                       |
| MCP Agent Tools     | 12                                       |
| Microservices       | 5 (Neo4j, API, MCP, Agent, UI)           |
| LLM Providers       | 2 (Anthropic Claude, OpenAI GPT)         |
| AWS Environments    | 2 (difflabs us-east-1, AIA eu-central-1) |
| Authentication      | Microsoft Entra ID + Eugene JWT (HS256)  |

| Role                      | Responsibility                                |
|---------------------------|-----------------------------------------------|
| CSL Behring R&D           | Primary consumers of competitive intelligence |
| AI Accelerator Team (AIA) | Platform hosting and infrastructure           |
| Diffusion Labs (difflabs) | Development and DevOps environment            |
| Engineering Team          | Platform development and maintenance          |

| Decision        | Choice                       | Rationale                                                                       |
|-----------------|------------------------------|---------------------------------------------------------------------------------|
| Graph Database  | Neo4j 5.26 Community         | Industry standard for knowledge graphs, Cypher query language, APOC/GDS plugins |
| API Framework   | FastAPI (Python)             | Async support, auto-generated OpenAPI docs, Pydantic validation                 |
| Agent Framework | Strands (AWS)                | MCP integration, streaming support, conversation management                     |
| LLM Protocol    | Model Context Protocol (MCP) | Standard tool interface for LLMs, vendor-agnostic                               |
| UI Framework    | Streamlit                    | Rapid prototyping, built-in chat components, SSE streaming                      |
| IaC             | Terraform                    | Multi-environment deployment, state management, GitLab integration              |
| Auth            | Microsoft Entra ID           | Enterprise SSO requirement from CSL                                             |

| Service    | Technology        | Port (Local) | Source Directory   | Purpose                        |
|------------|-------------------|--------------|--------------------|--------------------------------|
| neo4j      | Neo4j 5.26.9 CE   | 17474/17687  | docker-compose.yml | Graph database with APOC + GDS |
| eugene_ws  | FastAPI + Uvicorn | 18000        | src/               | Core REST API (38 endpoints)   |
| eugene_mcp | FastMCP           | 18443        | agents/eugene-mcp/ | MCP tool server (12 tools)     |

| Service         | Technology        | Port (Local) | Source Directory         | Purpose            |
|-----------------|-------------------|--------------|--------------------------|--------------------|
| eugene_agent_ws | FastAPI + Strands | 18001        | agents/eugene-agent-ws / | LLM agent backend  |
| eugene_agent_ui | Streamlit         | 18501        | agents/eugene-agent-ui/  | Chat web interface |

| Category           | Endpoints                                                               | Description                                 |
|--------------------|-------------------------------------------------------------------------|---------------------------------------------|
| Authentication     | GET /login, GET /auth/whoami, GET /auth-callback                        | Entra ID OAuth + Eugene JWT issuance        |
| Health & Info      | GET /health, GET /, GET /releases                                       | System health, welcome, release notes       |
| Database Stats     | GET /stats                                                              | Node/relationship counts, data distribution |
| Node Lookup        | GET /node/find/{value}, POST /node/details                              | Find node IDs, get detailed node info       |
| Graph Navigation   | GET /graph/relationship/start/{id}, GET /graph/path/start/{id}/end/{id} | N-hop subgraph, shortest path, reachability |
| Graph Facts        | GET /graph/facts/start/{id}                                             | English-language facts from relationships   |
| Labels & Counts    | GET /labels/{label}, GET /count/{label}                                 | List/count nodes by type                    |
| Drug Aliases       | GET /drugs/aliases/{name}, GET /drugs/aliases/id/{id}                   | Drug synonyms and brand names               |
| Patent Search      | GET /patents/drugs, /clinicaltrials, /geneproteins + counts             | Patent lookup by drug/trial/gene            |
| PubMed Search      | GET /pmids/drugs, /clinicaltrials, /geneproteins + counts               | Article lookup by drug/trial/gene           |
| Facet & Similarity | POST /facet/{label}, POST /similarity/{label}                           | Faceted search, vector similarity           |
| Organizations      | GET /organizations/{name}, GET /organizations/assets/{id}               | Org search, asset discovery                 |

| Tool                     | Method                  | Purpose                                      |
|--------------------------|-------------------------|----------------------------------------------|
| fetch_identity           | EugeneIdentityTools     | Retrieve current user identity from JWT      |
| fetch_by_label           | EugeneFetchTools        | List drugs or diseases with pagination       |
| fetch_similar            | EugeneFetchTools        | Find similar nodes by relationship patterns  |
| lookup_node_by_value     | EugeneNodeTools         | Resolve entity name to node ID (fuzzy match) |
| fetch_node_details       | EugeneNodeTools         | Get detailed node attributes by ID           |
| fetch_drug_aliases       | EugeneDrugTools         | Get all drug synonyms and brand names        |
| fetch_facts              | EugeneFactTools         | Paginated facts retrieval (up to 50 pages)   |
| fetch_node_relationships | EugeneGraphTools        | N-hop relationship traversal                 |
| fetch_paths              | EugeneGraphTools        | Find path between two nodes                  |
| has_reachable_path       | EugeneGraphTools        | Check reachability between nodes             |
| find_organization_names  | EugeneOrganizationTools | Wildcard search for organizations            |
| find_organization_assets | EugeneOrganizationTools | Get org assets (drugs, patents, trials)      |

| Requirement     | Target                 | Implementation                                     |
|-----------------|------------------------|----------------------------------------------------|
| Availability    | 99.5% uptime           | ECS with health checks, auto-restart on failure    |
| Scalability     | Horizontal scaling     | Stateless HTTP services, Docker containerization   |
| Response Time   | < 5s for graph queries | Neo4j APOC optimized queries, pagination           |
| Security        | Enterprise-grade auth  | Entra ID SSO + JWT + role-based access             |
| Data Freshness  | Batch updates          | Data ingestion scripts, S3-based loading           |
| Observability   | CloudWatch monitoring  | Health endpoints, API canaries, logging            |
| Portability     | Multi-environment      | Docker Compose (local), ECS (cloud), Terraform IaC |
| Maintainability | DDD architecture       | Hexagonal pattern, domain isolation, test coverage |

| Node Label      | Count   | Source              | Key Properties                               |
|-----------------|---------|---------------------|----------------------------------------------|
| drug            | 4,600   | DrugBank            | node_id, node_name, description, drugbank_id |
| disease         | 17,100  | Disease Ontology    | node_id, node_name, MONDO_ID, MONDO_NAME     |
| gene_protein    | 27,700  | Protein DBs         | node_id, node_name, annotation               |
| clinical_trial  | 49,300  | ClinicalTrials.gov  | node_id, node_name, nct_id                   |
| organization    | 26,100  | Company/Sponsor     | node_id, node_name, disambiguated            |
| pathway         | Various | Pathway DBs         | node_id, node_name                           |
| anatomy         | Various | Anatomical Ontology | node_id, node_name                           |
| patent          | Pending | USPTO               | node_id, node_name, patent_number            |
| pubmed_document | Pending | PubMed/PMC          | pmid, title, doi                             |

| Category       | Relationships                                                   |
|----------------|-----------------------------------------------------------------|
| Therapeutic    | indication, contraindication, off_label_use, drug_effect        |
| Biological     | drug_protein, disease_protein, pathway_protein, protein_protein |
| Phenotypic     | disease_phenotype_positive, disease_phenotype_negative          |
| Anatomical     | anatomy_protein_present, anatomy_protein_absent                 |
| Disease        | disease_disease (comorbidities)                                 |
| Drug           | drug_drug, has_drug_alias                                       |
| Publication    | disclosed_in, featured_in, analyzed_in, has_publication         |
| Organizational | sponsor, patent_holder, partnership                             |

| Tool Class              | Tools                                                     | Purpose                              |
|-------------------------|-----------------------------------------------------------|--------------------------------------|
| EugeneIdentityTools     | fetch_identity                                            | User authentication validation       |
| EugeneFetchTools        | fetch_by_label, fetch_similar                             | Browse and find similar entities     |
| EugeneNodeTools         | lookup_node_by_value, fetch_node_details                  | Entity resolution and details        |
| EugeneDrugTools         | fetch_drug_aliases                                        | Drug name/synonym lookup             |
| EugeneFactTools         | fetch_facts                                               | Natural language fact extraction     |
| EugeneGraphTools        | fetch_node_relationships, fetch_paths, has_reachable_path | Graph traversal and pathfinding      |
| EugeneOrganizationTools | find_organization_names, find_organization_assets         | Company research and asset discovery |

| Source             | Data Type                                       | Ingestion Method              | Status                    |
|--------------------|-------------------------------------------------|-------------------------------|---------------------------|
| DrugBank           | Drugs, drug-disease, drug-protein relationships | CSV bulk import (neo4j-admin) | Loaded (4.6K drugs)       |
| Disease Ontology   | Diseases, MONDO mappings                        | CSV bulk import               | Loaded (17.1K diseases)   |
| Protein Databases  | Genes, proteins, pathways                       | CSV bulk import               | Loaded (27.7K genes)      |
| ClinicalTrials.gov | Clinical trials, sponsors                       | CSV + Cypher scripts          | Schema only (49.3K nodes) |
| USPTO              | Patents, applications, claims                   | API download + Cypher         | Not yet ingested          |
| PubMed/PMC         | Articles, affiliations, DOIs                    | API download + Cypher         | Not yet ingested          |
| CSL Internal       | TPP, organizational data                        | Manual + checkpoint files     | Partial                   |

| Layer          | Mechanism          | Algorithm    | Details                                       |
|----------------|--------------------|--------------|-----------------------------------------------|
| Enterprise SSO | Microsoft Entra ID | RS256 (OIDC) | Corporate identity, JWKS validation, 8h cache |

| Layer         | Mechanism           | Algorithm         | Details                                             |
|---------------|---------------------|-------------------|-----------------------------------------------------|
| API Tokens    | Eugene JWT          | HS256 (symmetric) | 24h TTL, claims: tid, sub, upn, roles               |
| MCP Server    | FastMCP JWTVerifier | HS256             | Middleware-based, validates every tool call         |
| Agent Backend | Allowlist + JWT     | HS256             | User UPN checked against pipe-separated allowlist   |
| Local Dev     | Bypass mode         | HS256             | ENVIRONMENT=local skips Entra ID, issues test token |

| Role                   | Scope                                 | Status                                   |
|------------------------|---------------------------------------|------------------------------------------|
| user.public.read       | Read access to public biomedical data | Implemented (default for all users)      |
| user.confidential.read | Read access to proprietary CSL data   | Defined, not yet enforced at query level |

| Variable                | Purpose                  | Service         |
|-------------------------|--------------------------|-----------------|
| ENTRA_CLIENT_ID         | OAuth application ID     | eugene_ws       |
| ENTRA_CLIENT_SECRET     | OAuth client secret      | eugene_ws       |
| ENTRA_TENANT_ID         | Azure AD tenant          | eugene_ws       |
| EUGENE_CLIENT_ID        | JWT audience claim       | All services    |
| EUGENE_CLIENT_SECRET    | JWT signing key (HS256)  | All services    |
| EUGENE_TENANT_ID        | JWT issuer tenant        | All services    |
| EUGENE_AGENT_ALLOWLIST  | Pipe-separated user UPNs | eugene_agent_ws |
| NEO4J_USERNAME/PASSWORD | Database credentials     | eugene_ws       |

| Item                  | Severity | Status    | Description                                                    |
|-----------------------|----------|-----------|----------------------------------------------------------------|
| Cypher Injection      | HIGH     | Open      | User input not parameterized in one-hop adapter query builder  |
| SSL verify=False      | MEDIUM   | By Design | Internal network; MCP/Agent HTTP clients skip TLS verification |
| Self-signed TLS certs | LOW      | Active    | Neo4j and ALB use self-signed certificates                     |
| No rate limiting      | MEDIUM   | Open      | API endpoints have no rate limiting middleware                 |
| SAST/DAST disabled    | MEDIUM   | Open      | Security scanning commented out in CI pipeline                 |
| Static role mapping   | LOW      | Open      | Roles hard-coded; should migrate to database                   |

| Provider  | Model                    | Max Tokens | Temperature | Top-P |
|-----------|--------------------------|------------|-------------|-------|
| Anthropic | claude-sonnet-4-20250514 | 16,384     | 0.7         | N/A   |
| OpenAI    | gpt-4.1-mini (default)   | 4,096      | 0.7         | 1.0   |

| Environment        | AWS Account  | Region       | Infrastructure  | Deployment Method                |
|--------------------|--------------|--------------|-----------------|----------------------------------|
| Local Dev          | N/A          | N/A          | Docker Compose  | docker compose up --build        |
| difflabs (Dev)     | 087084717211 | us-east-1    | ECS + EC2 + ALB | Manual scripts (bin/docker/ecr/) |
| AlA (Staging/Prod) | 010928221940 | eu-central-1 | ECS + EC2 + ALB | GitLab CI/CD pipeline            |

| Image     | Base             | Size Factor         | Special Dependencies                       |
|-----------|------------------|---------------------|--------------------------------------------|
| eugene_ws | python:3.13-slim | Large (PyTorch CPU) | torch, transformers, sentence-transformers |

| Image           | Base                   | Size Factor | Special Dependencies              |
|-----------------|------------------------|-------------|-----------------------------------|
| eugene_mcp      | python:3.13-slim       | Small       | fastmcp, httpx, mcp               |
| eugene_agent_ws | python:3.13-slim       | Medium      | strands-agents, anthropic, openai |
| eugene_agent_ui | python:3.13-slim       | Small       | streamlit, httpx                  |
| eugene_neo4j_ce | neo4j:5.26.9-community | Medium      | APOC, GDS, AWS CLI, SSM Agent     |

| Module          | Resources                                           | Key Variables                                      |
|-----------------|-----------------------------------------------------|----------------------------------------------------|
| eugene-ec2      | EC2 (r7i.xlarge), EBS, Security Groups, IAM Profile | vpc_id, subnet, instance_type, availability_zone   |
| eugene-services | ECS Cluster, Task Definitions, ALB Target Groups    | ecs_cluster_name, subnets, image_hashes, IAM roles |

| Service         | Endpoint                     | Method             | Interval          |
|-----------------|------------------------------|--------------------|-------------------|
| neo4j           | http://localhost:7474        | wget spider        | 10s (30s startup) |
| eugene_ws       | http://localhost:8000/health | Python urllib      | 10s (15s startup) |
| eugene_mcp      | http://localhost:8000/health | HTTP GET           | On demand         |
| eugene_agent_ws | http://localhost:8000/health | HTTP GET           | On demand         |
| eugene_agent_ui | http://localhost:8501        | Streamlit built-in | On demand         |

| Canary                    | Endpoint Tested        | Validation                     |
|---------------------------|------------------------|--------------------------------|
| HealthEndpointCanary      | GET /health            | Response 200 OK                |
| StatsEndpointCanary       | GET /stats             | Valid node/relationship counts |
| NodeFindEndpointCanary    | GET /node/find/{value} | Node ID returned               |
| NodeDetailsEndpointCanary | POST /node/details     | Node attributes returned       |
| LabelCountsEndpointCanary | GET /count/{label}     | Non-zero counts                |

| Component   | Configuration                                                |
|-------------|--------------------------------------------------------------|
| Framework   | pytest 7.0+                                                  |
| Parallelism | pytest-xdist (-n 2, 2 workers)                               |
| Coverage    | pytest-cov (src/ directory, LCOV + terminal output)          |
| Markers     | serial (no parallelization), integration (integration tests) |
| Environment | Loads .env file via pytest-dotenv                            |
| Formatter   | Black (line-length 88)                                       |
| Linters     | Ruff, mypy                                                   |

| Domain               | Test Modules                                                          | Type                 |
|----------------------|-----------------------------------------------------------------------|----------------------|
| Organization         | name_verifier_mapper_test, resolution_mapper_test, provider tests (4) | Unit                 |
| Utility              | regex_validation_test                                                 | Unit                 |
| Graph Infrastructure | conftest.py fixtures                                                  | Integration fixtures |
| Foundation           | Various adapter/provider tests                                        | Unit + Integration   |
| Patent               | Patent adapter tests                                                  | Unit                 |
| Document             | Document analysis tests                                               | Unit                 |

| Risk                                | Severity | Likelihood | Mitigation                                        | Status |
|-------------------------------------|----------|------------|---------------------------------------------------|--------|
| Cypher injection in one-hop adapter | HIGH     | MEDIUM     | Parameterize all Cypher queries using \$variables | OPEN   |
| No API rate limiting                | MEDIUM   | MEDIUM     | Implement FastAPI middleware (slowapi or custom)  | OPEN   |

| Risk                                | Severity | Likelihood | Mitigation                                                 | Status      |
|-------------------------------------|----------|------------|------------------------------------------------------------|-------------|
| Self-signed TLS certificates        | LOW      | HIGH       | Migrate to AWS ACM for automatic renewal                   | OPEN        |
| Neo4j Community Edition limits      | MEDIUM   | LOW        | Upgrade to Enterprise for RBAC, clustering                 | DEFERRED    |
| SAST/DAST scanning disabled         | MEDIUM   | HIGH       | Re-enable Checkov and add Snyk/Trivy scanning              | OPEN        |
| Static role mapping in code         | LOW      | LOW        | Migrate to database-backed role management                 | OPEN        |
| Incomplete data (patents, PubMed)   | MEDIUM   | HIGH       | Execute pending ingestion pipelines                        | IN PROGRESS |
| Single MCP instance bottleneck      | MEDIUM   | LOW        | Implement S3SessionManager for horizontal scale            | PLANNED     |
| AIA deployment broken               | HIGH     | HIGH       | Fix ECS IAM permission issue                               | OPEN        |
| API key exposure in docker.env      | HIGH     | LOW        | Ensure .gitignore excludes docker.env; use Secrets Manager | MITIGATED   |
| Python version variance (3.12/3.13) | LOW      | LOW        | Standardize on single Python version across all containers | OPEN        |
| LLM provider outage                 | MEDIUM   | LOW        | Dual provider support (auto-fallback Anthropic <-> OpenAI) | PARTIAL     |

| #  | Question                                                               | Context                                                          | Priority |
|----|------------------------------------------------------------------------|------------------------------------------------------------------|----------|
| 1  | When will patent and PubMed data ingestion be completed?               | Code exists but data not loaded; impacts agent answer quality    | HIGH     |
| 2  | Should role-based data filtering be enforced at API or database level? | Neo4j CE lacks native RBAC; application-level filtering required | MEDIUM   |
| 3  | What is the plan for AIA deployment ECS IAM fix?                       | Production deployment blocked by permission issue                | HIGH     |
| 4  | Should the MCP server scale horizontally with S3SessionManager?        | Current FileSessionManager limits to single instance             | MEDIUM   |
| 5  | Is the qa branch ready to merge into main?                             | Branches are currently desynced; may cause deployment issues     | HIGH     |
| 6  | Should SAST/DAST scanning be mandatory for all merge requests?         | Currently commented out in pipeline; security gap                | MEDIUM   |
| 7  | What is the TLS certificate renewal strategy?                          | Self-signed certs expire; manual process today                   | MEDIUM   |
| 8  | Should clinical trial data be re-ingested or removed from schema?      | 49.3K node shells exist but data was cleared                     | LOW      |
| 9  | Is Neo4j Enterprise Edition justified for production RBAC needs?       | CE limitations affect security posture                           | LOW      |
| 10 | What additional MCP tools (PubMed, ChEMBL, etc.) are needed for MVP?   | Tool stubs exist in code but are disabled                        | MEDIUM   |
| 11 | What is the target SLA for API response times?                         | No documented performance requirements                           | MEDIUM   |
| 12 | Should the Claude Desktop MCP integration be prioritized?              | Referenced in next_steps.md as agentic coworker example          | LOW      |