



EUGENE PLATFORM

Technical Recommendations & Next-Generation Architecture

Audit Findings, Multi-Agent Target State and a User-Experience Enhancement Programme

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

FRONT MATTER	2
Foreword	2
Document Map	2
Executive Summary	2
PART I — AUDIT FINDINGS AND REMEDIATION	3
Audit Method and Scoring	3
Critical Findings	4
High-Severity Findings	4
Medium-Severity Findings	6
Positive Findings	7
Quick Wins versus Strategic Items	8
PART II — NEXT-GENERATION ARCHITECTURE	8
Target State Overview	8
Specialist Agent Topology	10
AgentCore Layer	10
Authoritative Knowledge & Data Sources	11
Security, Identity, and Governance	11
PART III — UI / UX ENHANCEMENT PROGRAMME	12
Current UI Assessment	12
Fifteen Concrete Enhancements	13
Accessibility and Inclusivity	14
PART IV — PROGRAMME ROADMAP	14
Four-Phase Plan	14
Non-Functional Targets	15
Investment, Risk and ROI	15
Appendix A — Findings Reference (Full)	16
Appendix B — AI Risk Register	18
Appendix C — Glossary	18

FRONT MATTER

Foreword

This volume is the companion to *Eugene Platform — Project Evaluation*. Where the evaluation establishes what has been built, this document establishes what should be built next. Three concerns are addressed in sequence. First, the twenty-one findings raised by the Stage 1 audit, with verbatim descriptions, severity, and concrete remediation. Second, the next-generation architecture proposed in `CSL_Arch_nextGen.pdf` — a CSL-owned orchestration layer running specialist agents on AWS Bedrock AgentCore, with the Eugene knowledge graph as one authoritative source alongside SharePoint, Confluence, deal eRooms, partner platforms, and external public sources. Third, a user-experience enhancement programme of fifteen concrete improvements, derived from a feature-by-feature reading of the current Streamlit and Next.js implementations.

The intent is that this document, read together with the evaluation, supplies stakeholder leadership with everything required to commission Stage 2: the audit findings to be remediated, the architectural target to be built toward, the user-experience commitments to be made, and the phasing and investment envelope under which all of the above can be delivered.

Document Map

The volume is organised in five parts:

- Part I — Audit Findings and Remediation — the twenty-one findings of Stage 1 in full, with severity, description, recommended remediation and effort estimate, plus the six positive findings that the audit asks to preserve.
- Part II — Next-Generation Architecture — the target topology, specialist agent layout, AgentCore service decomposition, knowledge and data sources, and security / governance posture.
- Part III — UI / UX Enhancement Programme — current-state assessment of both UIs, fifteen concrete enhancements, accessibility, and a delivery plan for the user-experience workstream.
- Part IV — Programme Roadmap — four-phase plan with weeks, deliverables, and entry / exit criteria; investment outline; non-functional targets.
- Part V — Risk, Investment and Appendices — programme risks, expected ROI, and reference appendices.

Executive Summary

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. Estimated remediation effort is approximately eight weeks of focused engineering work to complete every CRITICAL and HIGH item, with the MEDIUM items folded into the broader programme.

Critical Items at a Glance

(1) SSL certificate verification is disabled in MCP-client connections (SEC-01), exposing service-to-service traffic to MITM risk; the fix is a half-day. (2) The Strands SlidingWindowConversationManager is shared

across all agent instances (AGT-01), causing concurrent-session corruption under load; the fix is a one-day instancing change. Both should land in the first week of Phase 1.

Strategic Direction

Beyond the fix list, this document recommends adopting the next-generation architecture set out in `CSL_Arch_nextGen.pdf`: a CSL-owned Agent Orchestration & Policy Layer that hosts a Supervisor and five specialist agents (Eugene, Research, PubMed, SEC, ClinicalTrials) on AWS Bedrock AgentCore, with AgentCore Runtime, Memory, Gateway, and Observability as the underlying primitives. The authoritative knowledge tier brings together the Eugene Neo4j graph, a Bedrock Knowledge Base for unstructured CSL content, the existing Milvus vector index, SharePoint / Confluence corpora, and controlled-access deal eRooms. Cross-cutting governance is unified under Entra ID SSO with RBAC, Bedrock Guardrails, AWS PrivateLink, KMS, MFA, and 21 CFR Part 11-grade audit logging.

User Experience Direction

The current Streamlit UI delivers a serviceable chat surface but does not expose Eugene's structural strengths — the graph, the citations, the agent's reasoning trail. The Next.js v2 implementation introduces the right primitives (three-pane layout, Cytoscape graph canvas, evidence and node-detail panels), but its feature surface is incomplete. We propose fifteen concrete enhancements, ranging from conversation export and citation drilldown to a tool-call timeline, agent-decision lineage, and accessibility overhaul. The full set is sized at six to eight engineering weeks and is sequenced inside the Phase 2 / Phase 3 windows.

Phasing

A four-phase roadmap is proposed. Phase 1 (Weeks 1–4) addresses CRITICAL and HIGH-severity findings. Phase 2 (Weeks 5–8) hardens session management, observability, and the first cohort of UI enhancements. Phase 3 (Weeks 9–12) introduces the supervisor-plus-specialist agent topology, three new specialist tools (PubMed, SEC, ClinicalTrials), and the next-generation graph canvas. Phase 4 (Weeks 13+) deploys Bedrock Knowledge Base for unstructured retrieval, audit lineage, and the final UI accessibility programme.

Investment Targets

Non-functional targets after the full programme: P50 latency under two seconds for simple queries; P95 under ten seconds for complex multi-hop; one hundred concurrent agents sustained; 99.9 percent availability across AZs; full data residency within the AWS partition; and audit traceability for every agent decision.

PART I — AUDIT FINDINGS AND REMEDIATION

Audit Method and Scoring

1.1 Scope and Method

The Stage 1 audit covered architecture, code quality, agent-framework design, security posture, observability maturity, latency, state-synchronisation gaps, and graph-schema completeness. Methods included exhaustive source-code review, live system inspection in the difflabs (Dev) environment, and architectural analysis across all four microservices: `eugene_ws`, `eugene-agent-ws`, `eugene-mcp`, and `eugene-agent-ui`. Findings were classified by severity (CRITICAL, HIGH, MEDIUM) using a likelihood × impact matrix.

1.2 Severity Definitions

Severity	Definition	Programme placement
CRITICAL	Active or imminent data integrity / security risk; fix immediately.	Phase 1 — Week 1
HIGH	Material risk to availability, correctness, or compliance.	Phase 1 — Weeks 1–4
MEDIUM	Operational or maturity gap; reduces robustness over time.	Phases 2 / 3

1.3 Effort Estimates

Effort is reported in engineering days as a single-developer estimate. The full backlog (CRITICAL + HIGH + MEDIUM) sums to approximately thirty-one days of focused work, comfortably accommodated inside the eight-week Phase 1 + Phase 2 envelope when parallelised across two to three engineers.

Critical Findings

Two CRITICAL findings dominate Phase 1. Both are fixable in a day or less; both should land before any further functional work is committed.

2.1 SEC-01 SSL verification disabled in MCP client

The Agent Backend's MCP client is configured with `verify=False` on its `httpx` connection to the MCP server. In the `difflabs` and `AIA` environments, this means service-to-service traffic between the agent and MCP tiers is vulnerable to man-in-the-middle interception. The remediation is to set `verify=True` and, for self-signed development certificates, supply a `cert path`. Estimated effort: 0.5 day.

2.2 AGT-01 Shared `SlidingWindowConversationManager`

A single `SlidingWindowConversationManager` instance is shared across all agent instances in the `agent-ws` process. Under concurrent load, the manager's internal state can interleave between sessions, leading to incorrect conversation history and contaminated reasoning. The remediation is to instantiate the manager per conversation rather than per service. Estimated effort: 1 day.

[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

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

3.1 Agent-Tier HIGH Findings — Recommended Pattern

AGT-02 through AGT-05 share a common root cause: the agent's surface area was assembled from sensible defaults rather than from explicit production constraints. The recommended remediation pattern is a single

AgentExecutionPolicy module that centralises `max_iterations` (set to twenty-five), wall-clock timeout (300 seconds), per-LLM-call timeout (60 seconds for both providers), and the tool allowlist (excluding `python_repl` outside developer environments). Centralising these knobs makes future tuning a configuration change, not a code change.

3.2 Database HIGH Findings

DB-01 introduces explicit transaction and connection timeouts on the Neo4j driver. The recommended values — `connection_timeout=30` and `max_transaction_retry_time=30` — are conservative and balance fail-fast behaviour against transient network blips.

3.3 Security HIGH Findings

SEC-02 corrects the allowlist exception type so that unauthorised users receive HTTP 403 with a clean error body rather than an HTTP 500 with a stack trace. SEC-03 introduces application-level rate limiting via `slowapi`: 100 requests per minute per IP and ten concurrent agent sessions per user. These guardrails are the minimum required for any broader rollout; the audit recommends lifting them to API Gateway in Phase 3 once the next-generation architecture is in place.

3.4 Observability HIGH Findings

The four observability findings — OBS-01 through OBS-04 — together deliver the telemetry backbone the platform currently lacks: structured JSON logging via `python-json-logger`, application metrics via Prometheus or CloudWatch EMF, distributed tracing via OpenTelemetry plus AWS X-Ray, and deep `/health` endpoints that confirm Neo4j, MCP, and LLM availability. These findings are the costliest of the HIGH items in absolute days (combined approximately eleven days), but they unlock everything that comes after.

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 the 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

4.1 PII / PHI as a Strategic Item

SEC-05 is filed as MEDIUM but should be treated as a strategic dependency for any expansion of the platform's data footprint. The recommended approach is an AWS Comprehend PII-detection middleware that redacts candidate PII spans both inbound (in user prompts before they reach the LLM) and outbound (in agent responses before they reach the user). Comprehend's biomedical entity types are an additional benefit because they align well with Eugene's domain.

Positive Findings

The audit also surfaced six structural strengths that should be preserved through the next-generation migration. 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. —

5.1 Why These Strengths Matter

Each positive finding maps directly to a Stage 2 design choice. The DDD / hexagonal pattern is preserved by keeping the Core API as the single graph data interface even after the supervisor-plus-specialist topology is introduced. The forty-four-node ontology becomes the schema contract for the new specialist agents. Dual-LLM provider auto-detection is extended to AWS Bedrock as the primary provider, with Anthropic as fallback. Stateless MCP design is the precondition for AgentCore Gateway adoption. Streaming SSE is preserved end-to-end. Parameterised Cypher and the two-hop cap remain inviolable.

Quick Wins versus Strategic Items

6.1 Programme Sequencing

Half of the findings — eleven of twenty-one — can be resolved in 0.5 to 1.0 day each and should be batched into a single hardening sprint at the start of Phase 1. The remaining items require multi-day investment and are sequenced into Phases 1 (heavy lifting on observability), 2 (session and PII), and 3 (rate limiting, token revocation).

Category	Findings	Total 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

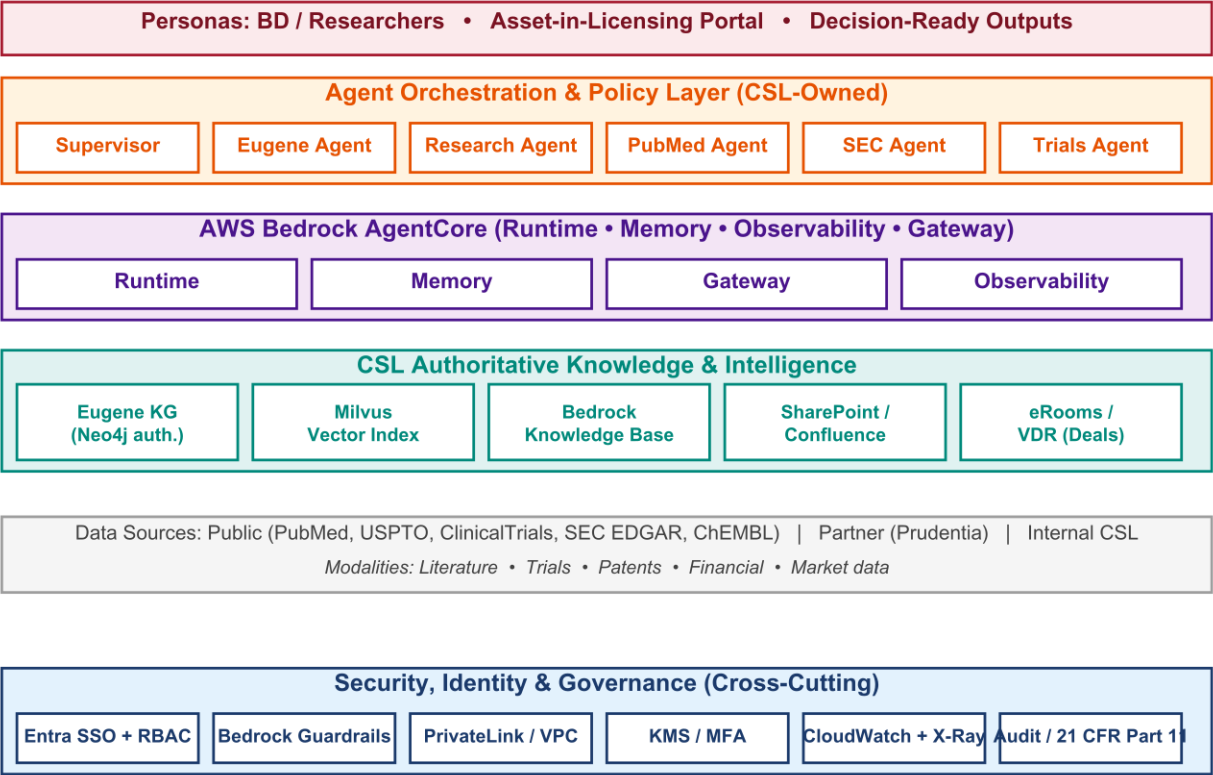
PART II — NEXT-GENERATION ARCHITECTURE

Target State Overview

7.1 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)



7.2 Layer Decomposition

Layer	Responsibility
Users & Experience	Asset-in-licensing portal; chat UI; decision-ready outputs for BD / Researchers
Agent Orchestration & Policy (CSL)	Supervisor + specialist agents; routing; policy enforcement
AWS Bedrock AgentCore	Runtime, Memory, Gateway, Observability
Authoritative Knowledge	Eugene KG, Milvus, Bedrock Knowledge Base, SharePoint / Confluence, eRooms
Data Sources	Public (PubMed, USPTO, Trials, SEC, ChEMBL); Partner (Prudentia)
Security, Identity & Governance	Entra SSO + RBAC; Guardrails; PrivateLink; KMS; X-Ray; Audit

7.3 Key Shifts From the Current State

- Single agent → Supervisor + 5 specialists (Eugene, Research, PubMed, SEC, ClinicalTrials, Synthesis). Domain-optimised models and tool sets.
- External LLM APIs → AWS Bedrock as 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.

Specialist Agent Topology

8.1 Supervisor + Five Specialists

The proposed multi-agent layout is summarised below. The Supervisor performs intent classification and dispatches to one or more specialists, in parallel where the intent permits. Each specialist publishes its own MCP tool surface; the Supervisor never queries data sources directly, preserving a single audit boundary per request.

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

8.2 Routing Policy

The Supervisor's routing policy is intent-driven. Intents are classified into a small number of buckets — Graph Query, Literature Search, Financial / M&A, Trial Search, Cross-Source Research, Out-of-Scope — with Claude Sonnet as the classifier. Each intent maps to a default specialist plus optional secondaries. Cross-Source Research dispatches to Research, Eugene, PubMed, and SEC concurrently and reconciles answers via the Synthesis Agent.

8.3 Why Specialists, Not One Agent

The single-agent design has three pathologies that worsen with scale. First, prompt and tool bloat: the system prompt currently lists twelve tool schemas; with PubMed, SEC, Trials, and ChEMBL added, the schema bill approaches the model's context budget at every turn. Second, domain-inappropriate model selection: financial filings and biomedical literature do not benefit from the same model size or temperature. Third, auditability: a single conversation that touches three domains is harder to attribute, govern, and reuse. Specialisation addresses all three.

AgentCore Layer

9.1 AgentCore Primitives

- AgentCore Runtime — managed agent execution; replaces the bespoke FastAPI agent service.
- AgentCore Memory — managed conversation state; replaces the FileSessionManager and the AGT-06 finding.

- AgentCore Gateway — managed tool surface and authentication; replaces the bespoke FastMCP front-door, with Entra ID and JWT validation lifted to the gateway.
- AgentCore Observability — managed telemetry pipeline into CloudWatch and X-Ray; addresses OBS-01 through OBS-03.

9.2 Migration Considerations

AgentCore is most valuable where the existing service has a thin framework boundary. The Agent Backend is a strong candidate because the Strands ReAct loop is the substantive logic; the FastAPI surface is thin. The MCP server is a moderate candidate; the twelve tools port cleanly to AgentCore Gateway, but the gateway's tool definitions diverge slightly from FastMCP's conventions and require a translation step. The Core API is intentionally not migrated — it remains the canonical graph data interface.

Authoritative Knowledge & Data Sources

10.1 CSL Authoritative Knowledge Tier

- Eugene Knowledge Graph — Neo4j as the structured biomedical authority.
- Milvus Vector Index — semantic similarity for entity and snippet retrieval.
- Bedrock Knowledge Base — managed retrieval over unstructured CSL content (S3 vectors).
- SharePoint / Confluence — internal tacit knowledge: deal memos, prior diligence notes, IC decisions.
- CSL eRooms / VDR — controlled-access deal-room artefacts; highly sensitive data with separate governance.

10.2 External Data Sources

- PubMed — literature, abstracts, citations.
- USPTO — patents, filings, assignees.
- ClinicalTrials.gov — trial records, phases, sponsors.
- SEC EDGAR — financial filings, M&A, regulatory disclosures.
- ChEMBL — bioactivity, target, and assay data.

10.3 Partner Platforms

The Prudentia Sciences platform contributes four modules — Pattern, Prospect, Prism, Pinnacle — accessed through agreed connectors. The next-generation Eugene architecture treats these as a federated knowledge source: queryable via specialist tools but governed by partnership contracts rather than direct CSL data plane controls.

Security, Identity, and Governance

11.1 Identity and Access

Entra ID SSO remains the user-facing front door, augmented by RBAC policies enforced at the AgentCore Gateway. The roles claim that today exists in the JWT but is not enforced becomes an enforced policy hook in the gateway. Multi-factor authentication is required for all non-read-only roles.

11.2 Governance Controls

- Bedrock Guardrails — content filtering, PII detection, denied-topic policies applied to every LLM call.

- AWS PrivateLink / VPC — all inter-service traffic stays inside the AWS partition; no public-internet egress for LLM calls.
- KMS — keys for at-rest encryption, JWT signing, secret rotation.
- CloudWatch + X-Ray — unified telemetry across the orchestration layer and AgentCore primitives.
- 21 CFR Part 11-grade audit logging — every agent decision, tool invocation, and data access is recorded immutably; supports regulatory inspection.

11.3 Customer-Level Data Isolation

The next-generation architecture explicitly commits to two principles. First, customer-level data isolation — CSL data does not commingle with any other tenant's data, even within the LLM provider's infrastructure. Second, contractual no-training — the LLM provider must not use CSL data for model training. Both commitments are unlocked by AWS Bedrock's data-governance posture.

PART III — UI / UX ENHANCEMENT PROGRAMME

Current UI Assessment

12.1 Streamlit (eugene-agent-ui)

The Streamlit UI delivers a working chat surface but does not expose the platform's structural strengths. The graph is invisible. Citations are embedded in chat text rather than surfaced as inspectable evidence. Conversation history cannot be exported. The agent's reasoning trail is opaque. Mobile experience is poor; accessibility primitives are minimal. These gaps are framework artefacts, not data-layer constraints.

12.2 Next.js v2 (eugene-agent-ui-next)

The Next.js v2 implementation introduces the right primitives. A three-pane AppShell separates Sidebar, Chat, and Knowledge Graph. A Cytoscape-driven graph canvas provides node selection, expansion, pinning, path-finding, source / sink marking, edge-type filtering, and PNG export. A floating glass-card overlay system supports a Toolbar, Legend, Stats, Node Details panel, and Evidence panel. The streaming client uses an event-driven `parseStream()` function that updates UI state per token. The implementation is closer to a strategic frontend than the Streamlit prototype, but the feature surface is incomplete.

12.3 Gap Inventory

- No conversation export (JSON, Markdown, PDF).
- No citation panel or evidence drilldown.
- No graph mini-map; users get lost when zoomed in.
- No tool-call timeline or agent-decision lineage.
- No role-based hints on tools (read-only / admin / PII-risky).
- No system-prompt templates or domain modes.
- No share / export of conversations.
- Limited accessibility primitives.
- No keyboard shortcuts.
- No light / dark theming consistent across canvas.
- Poor mobile responsiveness.
- No search history or autocomplete.
- No node-property hover tooltips.

Fifteen Concrete Enhancements

The fifteen enhancements below are sized for delivery inside Phase 2 and Phase 3. Each is described with intent, observable behaviour, and an implementation note that grounds it in the existing codebase.

13.1 Conversation Snapshot Export

Add an Export menu in the Sidebar footer with three options: JSON (raw), Markdown (clean narrative with citations), and PDF (includes graph snapshot and tool-call timeline). Implementation: `html2pdf` + `markdown-it`; a new `exportConversation(format)` hook on the Chat store.

13.2 Citation and Evidence Drilldown Panel

Every assistant turn surfaces inline citations of the form [Source: Eugene Node ID:drug_001 | Confidence: 0.92 | Data Age: 3d]. Click a citation to open the `EvidencePanel`, populated by a new `/agent/conversation/{id}/lineage` endpoint that returns the full path: user prompt → intent → tool calls → Cypher executed → result → synthesised answer.

13.3 Graph Mini-Map

A floating viewport indicator in the bottom-right of the graph pane shows the full graph in miniature with the current viewport highlighted. Click the mini-map to jump to a region. Implementation: Cytoscape's minimap plugin or a custom canvas overlay.

13.4 Tool-Call Timeline

Render each turn's tool invocations as a Gantt-style bar chart below the assistant message. Bars are coloured by tool family (`eugeneGraph`, `pubmed`, `sec`, `trials`), labelled by tool name, and clickable to show input parameters, output size, latency, and a 'highlight nodes from this call' action that lights up the relevant subgraph in the canvas.

13.5 Role-Based Tool Hints

Hovering a tool name in the timeline or toolbar shows a badge: ■ Read-only, ■ Admin-only, ■ PII-risky, ■ Prod-safe. The badge is derived from the tool's MCP schema annotation; the implementation is an `AccessControl` enum exposed on each tool definition, surfaced through a `Tooltip` component.

13.6 System Prompt Templates

A Sidebar dropdown labelled "Agent Personality" offers four templates: Default Biomedical, Patent Analyst, Clinical Trialist, Competitive Intelligence. Each template selects a curated tool subset and prepends a domain-specific opening to the system prompt. Templates live in `app/templates/` and are loaded via a `/agent/templates` API.

13.7 Evidence Drilldown with Graph Expansion

Clicking a citation auto-expands the graph canvas to display the path from the user prompt to the cited fact. Path nodes are coloured red (start), green (end), and yellow (intermediate); edges are labelled with the relationship type. Implementation: extend the `useGraphExplorer.expand()` hook to accept a path argument.

13.8 Share and Collaborative Export

A Share button generates a unique short link (DynamoDB-backed, seven-day TTL) that returns a read-only view of the conversation, the frozen message history, and the graph snapshot. The shared view is uneditable but supports node inspection and export. Implementation: `POST /api/shares/{conversation_id}`; UUID returned as v2 short link.

13.9 Accessibility Overhaul

ARIA labels on every interactive element — graph nodes, toolbar controls, panel toggles. Live region announcements for tool-call events ("agent is calling `fetch_node_relationships`"). High-contrast theme. Logical tab order through Sidebar → Chat → Graph. Implementation: Chakra UI plus headless-ui ARIA

primitives with `aria-live="polite"` announcements.

13.10 Keyboard Shortcuts

Cmd+K to focus search; Cmd+N for new conversation; Cmd+E to export; Cmd+? to show the help / shortcut cheat-sheet; Escape to close all panels; Ctrl+Shift+L to cycle graph layouts. Implementation: a `useKeyboardShortcuts` hook in the AppShell.

13.11 Light / Dark Theme Toggle

Theme toggle button in the Sidebar header. Uses Chakra's `useColorMode()`; persists to `localStorage`; the graph canvas adapts its node and edge palette accordingly. Implementation: extend the existing `ColorModeSync` component.

13.12 Mobile-Responsive Layout

Tablet (768 px): hide sidebar by default behind a hamburger; graph shrinks to 50 percent. Mobile (< 640 px): stack chat above graph; graph hidden by default; tab bar to switch. Pinch-zoom and swipe gestures on the graph canvas. Implementation: Chakra responsive props are already used; extend to all panels.

13.13 Search History and Autocomplete

Sidebar section "Past Queries" shows the last ten prompts; click to rerun. Autocomplete in the chat input does fuzzy matching against past prompts and suggestion-pill text. Filter by date range, tool family, or node label. Implementation: `useLocalStorage` + `Fuse.js` for fuzzy search.

13.14 Agent-Decision Lineage UI

A "Why?" button on each assistant turn opens a modal showing the decision path: intent classification → agent routing → tool selection rationale. Each step shows a confidence score. Backed by the same `/agent/conversation/{id}/lineage` endpoint introduced in 13.2.

13.15 Node-Property Hover Tooltips

Hovering a graph node shows a floating tooltip: `[Label] node_name | Properties: { ... } | Confidence: 0.95 | Updated: 2 d ago`. Implementation: Cytoscape's `nodeHover` event positions the tooltip near the cursor, populated from `node.data()`.

Accessibility and Inclusivity

14.1 Standards Targeted

- WCAG 2.2 AA — keyboard, contrast, alt text, focus visibility.
- Section 508 — government-procurement-grade accessibility.
- Internationalisation primitives — message catalogue scaffolding for downstream localisation.

14.2 Inclusive Design Principles

The UI programme adopts three inclusive-design principles. First, no exclusive reliance on colour to communicate meaning — every colour code is paired with an icon or text label. Second, every interactive element exposes a name, role, and value to assistive technology. Third, animation respects `prefers-reduced-motion`. These are non-negotiable defaults baked into the component library.

PART IV — PROGRAMME ROADMAP

Four-Phase Plan

15.1 Phase Overview

Phase	Weeks	Outcomes
1. Foundation Fixes	1 – 4	All CRITICAL + HIGH findings resolved; structured logging; deep /health; rate limiting live.
2. Session & Observability	5 – 8	FileSessionManager replaced; OpenTelemetry + X-Ray; PII / PHI middleware; Neo4j S3 backups; first UI cohort (13.1–13.5).
3. Multi-Agent & New Tools	9 – 12	Supervisor + specialist topology; PubMed / SEC / Trials agents; Bedrock as primary LLM; UI cohort 13.6–13.10.
4. Vector Search & Audit	13 +	Bedrock Knowledge Base for unstructured KB; freshness tracking; explainability lineage API; UI cohort 13.11–13.15; accessibility audit.

15.2 Entry / Exit Criteria

- Phase 1 exit — 100 percent of CRITICAL / HIGH findings closed; structured logs in CloudWatch Logs Insights; rate limiting in production.
- Phase 2 exit — DynamoDB-backed sessions; OTel traces visible in X-Ray; PII redaction in production; first five UI enhancements shipped to staging.
- Phase 3 exit — Supervisor routing demonstrably correct on a fixed evaluation set; Bedrock as primary LLM; ten of fifteen UI enhancements live.
- Phase 4 exit — Bedrock KB powering at least one specialist; lineage API live; full UI accessibility audit passed.

Non-Functional Targets

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

Investment, Risk and ROI

17.1 Programme Risks

- Bedrock service availability in eu-central-1 for Phase 3 — fallback via cross-region routing or Anthropic.

- Source-data licensing for SEC EDGAR / ChEMBL beyond fair-use limits — procurement workstream in parallel with Phase 3.
- Change management on persona workflows — internal champions and phased rollout.
- Recruitment for the specialist UI workstream — secured through engagement with the existing Next.js team.

17.2 Expected ROI

The programme's ROI is the redirection of multiple analyst-days per investigation to minutes, scaled across the BD organisation. A conservative estimate, drawn from the validation document's value drivers, is two to four hours saved per significant query, with between fifty and two hundred such queries per week across the active analyst cohort. The eight-week Phase 1 + Phase 2 work pays for itself inside a quarter on time-saving alone; Phases 3 and 4 deliver the strategic positioning of Eugene as a CSL-wide intelligence platform.

Appendix A — Findings Reference (Full)

[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

[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

[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

Appendix B — AI Risk Register

ID	Risk	L	I	Mitigation
AI-R01	Hallucination in biomedical context	High	Critical	Citation enforcement; fact-check tool
AI-R02	PII / PHI in user prompts	Med	High	AWS Comprehend PII detection middleware
AI-R03	Prompt injection via tool results	Med	High	Input / output sanitisation
AI-R04	Unbounded agent cost	High	High	Token budget; cost alerting
AI-R05	python_repl arbitrary code execution	Low	Critical	Remove from default tools (AGT-05)
AI-R06	Model bias in drug recommendations	Med	High	Bias evaluation framework
AI-R07	Tool-result tampering	Low	High	Signed responses; integrity checks
AI-R08	Agent decision non-explainability	High	High	Citation trail; lineage API
AI-R09	Single-LLM-provider dependency	Med	Med	Dual-provider auto-detect (current)
AI-R10	Regulatory non-compliance (21 CFR Part 11)	Med	Critical	Audit log for all agent decisions

Appendix C — Glossary

Term	Meaning
AgentCore	AWS Bedrock managed services for agent runtime, memory, gateway, observability
Bedrock KB	Amazon Bedrock Knowledge Base — managed retrieval over S3 vector content
Cytoscape.js	JavaScript graph visualisation library used in the v2 UI

Term	Meaning
EDGAR	SEC's Electronic Data Gathering, Analysis, and Retrieval system
Guardrails	Bedrock-native content / PII / topic policy filters
MCP	Model Context Protocol — agent tool surface protocol
NVL	Neo4j Visualization Library (used in the v2 graph canvas exploration)
PrivateLink	AWS service exposing endpoints inside a VPC without public egress
RBAC	Role-Based Access Control
X-Ray	AWS distributed tracing service