ACCURA BIO SOLUTIONS TECHNICAL NOTE
Mar 2025, version 1.0

How a CGT Biotech Implemented a Governed Scientific Knowledge Platform Using Dual-Model LLM Architecture

Alberta C. Kapoor and Marina Colakovic

Introduction

A mid-sized Cell and Gene Therapy (CGT) biotechnology company headquartered in the Boston area, with approximately 15 years of operating history, one commercial therapy, and multiple active clinical-stage programs, initiated a multi-phase program to enable improved access to scientific, quality, and regulatory knowledge and data across the organization.

With time, documentation and data accumulated across several validated systems of record, including the company's Electronic Lab Notebook (ELN), Laboratory Information Management Systems (LIMS), Quality Management Systems (QMS), Electronic Document Management Systems (EDMS), and analytical data platforms. While each system was well controlled individually, there was no integrated knowledge access across these platforms, resulting in operational inefficiencies. Scientists, Chemistry, Manufacturing, and Controls (CMC), QA, and regulatory teams had to spend considerable time locating approved information across these systems of record, manually summarizing data, and preparing inspection-ready responses under considerable time pressure. Although the company conducted pilot studies on commercially available AI solutions, those studies failed due to hallucinations, the use of unapproved documents, and insufficient traceability and auditability.

Therefore, the company decided to build a custom secure, read-only, regulatory-compliant knowledge platform that enables natural-language querying of approved scientific and operational information. Although this platform uses Artificial Intelligence (AI) for natural-language querying, it was designed to operate under a strict governance framework that enforced existing access controls, document lifecycle rules, and traceability requirements, while supporting cross-document synthesis and scientifically worded explanations.

In the platform described in this case study, Large Language Models (LLMs) were used strictly as an interaction and reasoning layer, operating only over curated, policy-filtered content. The system was designed so that LLMs were not granted access to raw source systems, do not perform any autonomous analysis or decision-making, and do not write back to any system of record, to ensure alignment with GxP expectations and inspection-readiness requirements.

Disclaimer: *The architecture and development approach described in this case study represents one possible implementation of a governed AI platform within a regulated biotechnology environment. Organizations differ in their infrastructure maturity, data systems, compliance requirements, and technical constraints, and therefore should evaluate their own infrastructure, governance frameworks, and risk tolerances when determining the appropriate architecture. The tools, cloud services, and design patterns referenced here are illustrative examples. Each company must assess and validate its solution to ensure alignment with internal policies, regulatory requirements, and operational needs.*

Objectives

The purpose of the program was to build a secure, read-only, regulatory-aware knowledge platform that would enable natural-language querying of the company's proprietary, approved scientific and operational information.

The goal of the system was to answer scientific and operational questions, reason across experiments, samples, and regulatory documents, reduce manual search and interpretation effort, and maintain the origin of data sources and compliance. All while enforcing existing access controls, lifecycle rules, and data ownership, and producing citation-backed outputs that are inspection-ready. Furthermore, it was essential that such a system clearly distinguish between known information and gaps in the available knowledge to prevent hallucinations or unapproved information from being provided.

Furthermore, the goal was to allow the system to evolve and expand in a controlled way, so that each change would not necessitate an expensive and lengthy full-system revalidation. This meant that the platform had to be built from the ground up under strict governance to support compliance with quality, security, and regulatory expectations and prevent system overhauls, significant changes, or restructuring in later stages of development.

Platform Architecture

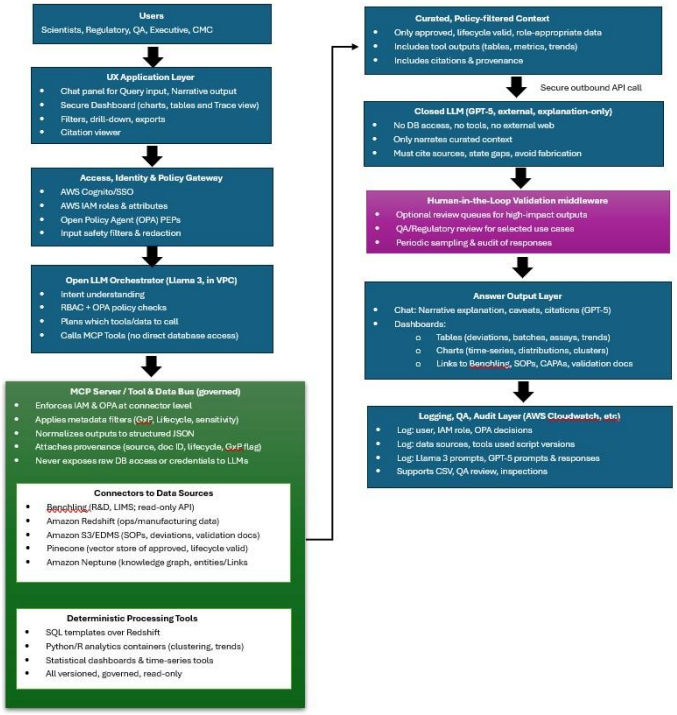

Figure 1: Architecture of the Governed Scientific Knowledge Platform.

Platform Architecture

The company implemented a hybrid architecture using an internally hosted open-source LLM (Meta's Llama 3) and a cloud-hosted commercial LLM (OpenAI's GPT-4) as shown in **Figure 1**. A hybrid open/closed LLM architecture can help organizations balance privacy, cost control, and capability diversity.

The open model preceded the closed model because retrieval, Role-Based Access Control (RBAC) enforcement, lifecycle filtering, and policy evaluation needed to occur inside the company's governance boundary. A closed, vendor-controlled model cannot perform these functions, cannot be validated, and cannot be trusted with raw or sensitive queries.

The open LLM (Llama 3) operated entirely within the company's private Amazon Web Services (AWS) Virtual Public Cloud (VPC). It had no internet access, no direct database access, and no ability to reach any system of record.

Requests were routed through multiple Policy Enforcement Points (PEPs), including AWS Identity and Access Management (IAM) role checks, Open Policy Agent (OPA) policy, and metadata-level filters based on GxP classification, lifecycle state, and sensitivity. As the model operated within the governance boundary, the company-maintained control over it and could implement RBAC, lifecycle rules, GxP classification, sensitivity filters, and policy logic. This setup also enabled deterministic orchestration of retrieval processes.

Llama 3 was not fine-tuned or trained on company data. Instead, it was tightly controlled through system prompts, policy logic, and a restricted set of pre-approved Model Context Protocol (MCP) tools. It could not execute arbitrary code, modify any system of record, or perform retrieval operations.

Version-controlled data processing workflows managed all parsing, chunking, metadata tagging, normalization, and analytics to ensure reproducibility outside of the LLM. The workflows were created and tested during the platform's initial development to guarantee that documents were preprocessed prior to being made accessible for retrieval. This supported processing regulated content in a traceable, reproducible, and inspection-ready manner.

Within these constraints, the open model handled intent classification, enforced RBAC and policy rules, identified which data sources were relevant, and orchestrated retrieval from validated sources such as GMP records, approved regulatory documents, and structured operating data. It assembled this information into a curated, policy-filtered context that included provenance, lifecycle metadata, and citations.

The selected context was provided to the closed LLM (GPT4), which functioned solely as an explanatory layer. GPT-4 could not access databases, invoke tools, or introduce external knowledge. The sole purpose was to transform the confirmed context into explanations supported by citations and scientific accuracy, while GPT-5 was instructed to point out any missing information whenever possible. The commercial LLM operated in a stateless, session-isolated configuration as administered by the cloud provider's service architecture. The Azure OpenAI documentation states that neither prompts nor customer data are stored or used in model training.

This dual model separation supported regulatory expectations around traceability, lifecycle control, and inspection readiness by making retrieval and reasoning independently verifiable.

Workflow Overview

The platform served a diverse group of users, including scientists, members of the regulatory and quality control teams, CMC, and company leadership. They used a dashboard to submit queries related to scientific and operational data.

Before any query reached the Llama 3 orchestration layer, user authentication and advanced authorization checks were enforced to ensure that only users with valid IAM roles could access the system. Once the request entered the private AWS environment, Llama 3 processed the query, interpreted the user's intent, extracted key entities such as cell line details or study identifiers, and applied access controls. These included OPA policy checks, metadata-based filters, and role-specific retrieval rules, designed so that only approved content could be retrieved.

Before any information could be retrieved, all company documents and datasets had already been processed through validated, rule-based ingestion workflows. These controlled workflows automatically parsed documents, applied lifecycle and GxP metadata, normalized identifiers, and generated embeddings, ensuring that only approved, traceable, and inspection-ready content was available for retrieval. The LLMs were designed not to parse or transform raw documents themselves; they operated over this pre-curated, policy-filtered knowledge layer.

Once permissions were validated, Llama 3 orchestrated retrieval by invoking pre-approved MCP tools, which functioned as governed connectors to vector stores, structured databases, Benchling APIs, document repositories, and deterministic analytics tools. The MCP layer was designed so that all retrieval was read-only, policy-enforced, and traceable.

Retrieved information was curated and organized into a structured, policy-filtered context that included provenance and lifecycle metadata before being sent to the closed LLM for synthesis. GPT-4 then generated a scientific explanation based solely on this approved context, citing all relevant data and identifying any gaps. It could not access databases, invoke tools, or introduce external knowledge.

The final answer was delivered through the dashboard, combining GPT-4's narrative with structured visualizations such as tables, charts, and links to underlying documents. Throughout the process, a comprehensive audit trail was maintained, logging the original query, data sources, policy decisions, model prompts, and system responses. This

supports quality assurance, regulatory inspections, and reproducibility.

Development Roadmap Overview

To manage this complex, companywide governance system and maintain compliance, the company adopted a five-phase, yearlong plan as depicted in **Figure 2**. Each phase of the program incrementally implemented and hardened the architecture layers.

Phase 0 aligned stakeholders and intended to design a platform that met QA, legal, and regulatory needs before any development work started. Phase 1 established a secure architecture using approved knowledge sources and implemented controlled information retrieval with audit systems. Phase 2 enhanced access control for inspection readiness. Phase 3 focused on selecting tools for deterministic data analysis and calculations. In Phase 4, the platform was advanced with knowledge graphs for institutional memory and time-awareness reasoning. Phase 5 validated the system, developed SOPs, and introduced ongoing activities like periodic reviews and computer system validation. The following section details each phase further.

Figure 2: Multiphase development process for the governed scientific knowledge platform.

Phase 0: Alignment on Key Decisions and Establishing Guardrails

Phase 0 focused on establishing technical and governance guardrails before any buildout of the system was started. Phase 0 involved no software development; it focused entirely on governance, architectural decisions, and alignment across QA, Regulatory, IT Security, and Data Engineering teams.
The objective was to define system boundaries, data classifications, and architectural constraints using clear infrastructure decisions that would support inspection readiness.

The team selected AWS as the secure cloud foundation and established a dedicated, isolated VPC to host all internal AI components. Early architectural decisions defined that the open model (Llama 3) would run inside this private AWS environment, while the closed reasoning model (GPT-4) would be accessed externally via a secure API with strict context controls. No model was permitted to access the database directly.

The team conducted a structured inventory of enterprise data systems, including Benchling (R&D and LIMS data), Amazon Redshift (structured operational data), and regulated document storage in Amazon S3. For each system, they documented data ownership, access permissions, compliance classification (GxP vs. non-GxP), sensitivity level, and lifecycle state. This inventory formed the foundation for metadata-driven retrieval controls and policy enforcement. Role definitions were aligned to AWS IAM groups to prepare for later enforcement of RBAC. Standards for data classification were established at the metadata level. Documents and datasets received tags based on the following criteria:

- GxP vs. non-GxP
- Sensitivity level (public, confidential, restricted)
- Lifecycle state (approved, draft, obsolete)

These classifications were designed to integrate later with policy enforcement logic (via OPA in Phase 2) and metadata-aware retrieval filters.

The team designed the system in separate, clearly defined modules:

- User query understanding: Llama 3 interprets the user's questions and decides what information is needed.
- Data retrieval**:** structured, rule-based processes locate the relevant data, using metadata and approved sources.
- Reasoning and explanation: GPT-4 generates scientific summaries and explanations based only on the curated data.
- Logging and auditing**:** All actions and queries are recorded for traceability using AWS CloudWatch. The team also established clear operational rules, recorded in formal architecture decisions:
- The system was read-only**,** so no changes could be made to underlying data.
- Retrieval was the only operation; no analysis or model fine-tuning was performed on regulated (GxP) data.
- The system could not write back to Benchling, Redshift, or other databases.
- No autonomous actions were allowed; human oversight was required.
- Responses were required to include citations to the original data.
- Interactions were auditable to support compliance and inspection readiness.

In parallel, use-case prioritization workshops were conducted, and two low-risk use-cases were selected: a validation document retrieval assistant and a deviation/CAPA historical pattern retrieval assistant. These use cases were chosen because they rely only on finalized documents and structured metadata rather than real-time operating systems (e.g., actively running experiments, active manufacturing batches, or live production).

By the end of Phase 0, the company had defined a secure AWS-based infrastructure, model separation strategy (Llama 3 internal, GPT-4 external), identity alignment via AWS IAM, and

metadata standards that supported compliance. This structure was intended to ensure that governance preceded implementation, so that when development began, it would occur within an enforceable, inspection-aligned boundary.

Phase 1: Build and Pilot Implementation

In this phase, the team established a secure, compliant AI platform that provides advisory insights based on approved GxP knowledge. The effort began with a comprehensive inventory of data across experimental, operations, regulatory, and LIMS systems to map ownership, access levels, and compliance requirements. The data domains were defined and classified (GxP vs. non-GxP, approved vs. draft) and aligned to a role-based permission matrix, which embedded governance controls into the platform from the start.

The team focused on ingesting and standardizing key documents such as SOPs, deviations, and validation reports. Units, terminology, timestamps, and identifiers in the scientific data were normalized, but the original records were kept intact to maintain traceability for audits. Unstructured documents were logically parsed into sections (e.g., stability results, validation summaries), enriched with lifecycle and compliance metadata, and indexed into a curated vector store designed for traceable retrieval rather than generative memory.

To support this ingestion effort, the team implemented validated workflows using rule-based Python workflows and AWS-native Extract Transform Load (ETL) components such as Lambda, Step Functions, Glue, and Textract. These workflows automatically extracted text from documents, parsed them into logical sections, applied lifecycle and GxP metadata, normalized units and identifiers, and generated embeddings using a fixed, version-locked model. These workflows were version-controlled, reproducible, and auditable, ensuring consistent processing of regulated content.

To support efficient and compliant access to structured operating data, the team implemented Model Context Protocol (MCP) tools that held the actual read-only database connections (e.g., to Amazon Redshift or Snowflake). The architecture was built so that structured operational data was stored in Amazon Redshift, but it also allowed for Snowflake MCP to be used for analytical tasks in later stages. The LLMs were not granted direct database access; instead, Llama 3 invoked these MCP tools to execute policy-enforced SQL queries inside the private AWS environment. This design preserved performance because the MCP servers queried the databases directly while ensuring that all retrieval remained read-only, auditable, and governed by the same RBAC and lifecycle rules applied to document ingestion. By separating database access from the LLM, the system maintains both efficiency and helps meet compliance objectives.

A basic LLM orchestration layer was implemented to handle user intent classification, permission checks, controlled data retrieval, and constrained scientific summarization with mandatory citations. Strict audit logging captured user queries, retrieved sources, model prompts, and outputs to ensure reproducibility and inspection readiness. Core architecture layers, including query routing, retrieval orchestration, constrained reasoning, and logging, were made operational, while higher-risk features such as analytics, autonomous actions, and system write-back were deliberately excluded.

Teams from Data Engineering, QA, IT Security, Regulatory, and pilot users worked together to make sure governance was properly enforced. As a result, the company's first inspection-defensible AI assistant, supporting SOP Q&A and deviation summaries, was built, and it provided a robust foundation for more advanced capabilities in later phases.

Phase 2: Strengthening Access Control and System Trust

Phase 2 focused on turning the LLM orchestration layer into an inspection-ready system by embedding formal access controls and policy enforcement into the platform itself. While earlier phases supported secure retrieval and constrained summarization, this phase made governance enforceable at the level of the system.

The team implemented centralized policy logic using Open Policy Agent (OPA). These policies were applied at multiple points in the system known as Policy Enforcement Points (PEP), including query routing and data retrieval. When a user submits a query, Llama 3 first checks the request against OPA policies before retrieving any data. These policies reference user roles and attributes from AWS IAM, ensuring that AI access follows the same role-based access rules as direct access to enterprise systems.
Access control was enforced at two layers: infrastructure-level IAM and network policies restricted unauthorized access to the LLM, while OPA policies and metadata-aware PEPs within the Llama 3 orchestration layer ensured that only role-appropriate, lifecycle-valid, and sensitivity-filtered content could be retrieved.

Lifecycle controls were enforced at the metadata level. Documents in Pinecone and structured records in Amazon Redshift were tagged with GxP classification, lifecycle status, and sensitivity level. Queries automatically filtered out draft or obsolete content before any data reached the reasoning stage. This supported that only compliant content could be accessed.

Within the private AWS environment, Llama 3 continued to handle intent classification, query routing, and retrieval orchestration, but now applied mandatory policy checks via OPA before fetching any data. The system was designed to assemble only policy-compliant content and sent securely to GPT-4, which remained externally hosted and strictly read-only. GPT-4 could not access databases or bypass policy logic; it could only summarize content that had already passed access controls.

Additional compliance safeguards included:

- Mandatory citations for all outputs
- Refusal templates for restricted or out-of-scope queries
- Validation checks before delivering final answers

Interactions were logged to support traceability via AWS CloudWatch and the following data was logged:

- User identity (AWS IAM)
- Policy decisions from OPA
- Documents retrieved from Pinecone and Redshift
- Model versions (Llama 3 and GPT-4)
- Prompts and final responses

This logging enabled replaying interactions and performing QA and compliance testing across roles and lifecycle states.

This phase was led by QA and the Computer System Validation (CSV) team, in collaboration with Security and Platform Engineering, and it embedded governance into the technical core of the system. By combining AWS identity management, OPA policy enforcement, metadata-level lifecycle gating, and strict separation of responsibilities between Llama 3 and GPT-4, the platform became a policy-enforced knowledge system capable of passing regulatory inspections.

Phase 3: Analytics and Tool Delegation

Phase 3 expanded the platform's operational value by moving analytical tasks into deterministic, auditable tools.
Analysis was delegated to controlled computational services running within Amazon Web Services (AWS), which reduced hallucination risk and enabled higher-impact use cases. The LLMs are not permitted to perform any data or statistical analysis because LLMs can generate fluent but factually incorrect outputs, "hallucinations." This makes them unsuitable for analytical tasks that require guaranteed correctness and traceability *(Farquhar et al., 2024, Nature)*. Hence, these tasks were specifically reserved for dedicated computational tools.

An analytics layer was introduced alongside existing retrieval services. Structured operations and manufacturing data continued to reside in Amazon Redshift, but now the retrieval orchestration managed by Llama 3 within the private AWS VPC could selectively invoke SQL queries for trend analysis, deviation frequency counts, and batch comparisons. For more advanced statistical evaluations, a controlled Python execution environment running on AWS (e.g., containerized compute within the VPC) was added. This environment executed validated scripts for clustering, statistical summaries, and time series analysis. Significantly, all tool executions were predefined, version-controlled, and access-governed through the same policy checks enforced by Open Policy Agent (OPA) and AWS IAM. All analytical tools were version-controlled, parameterized, and validated under CSV expectations to ensure deterministic, reproducible outputs.

When a user submitted an analytical request, such as deviation clustering or assay comparison, Llama 3 performed the intent classification and determined whether a tool call was required. If so, it invoked the appropriate SQL query in Redshift or a Python-based analysis job. The outputs of these tools, including structured tables, calculated metrics, and confidence indicators, were returned to the orchestration layer and assembled into the authorized knowledge context. Only these outputs, along with relevant supporting documents retrieved from Pinecone, were forwarded to GPT-4 for narrative synthesis.

In this architecture, GPT-4 acted strictly as a narrator. It translated structured outputs into scientifically appropriate explanations, maintained citation discipline, and included qualifiers or confidence statements derived directly from tool-generated results. This separation ensured that analytical accuracy could be attributed to verified code and database queries rather than to text generated by probabilistic models.

During this phase, the system's logging and audit capabilities were significantly strengthened. Using AWS CloudWatch, the platform recorded not only user questions and the documents retrieved, but also details about any tools used, database queries, versions of analysis scripts, input values, output results, timestamps, and the versions of the AI models involved. This level of detail was intended to ensure that every analytical result could be reproduced and independently reviewed during inspections or internal audits.

This phase expanded the types of use cases the system could support, such as identifying patterns in deviations across manufacturing lots, analyzing batch performance trends over time, comparing assay results, and generating structured summaries to support inspection readiness. The Data Science and Analytics teams led it and developed it in close collaboration with QA, Manufacturing, and QC subject matter experts. This enabled the organization to deliver meaningful operational insights while maintaining strict controls, compliance safeguards, and complete traceability.

Phase 4: Knowledge Graph and Scaleup

Phase 4 introduced long-term memory and cross-domain structure into the AI platform by layering a knowledge graph alongside the existing vector-based retrieval system. While earlier phases enabled controlled retrieval and deterministic analytics, this phase focused on connecting information across time, domains, and systems to support institutional learning rather than point-in-time question answering.
A graph database was deployed within Amazon Web Services (AWS), for example, using Amazon Neptune, to model relationships between entities such as compounds, assays, batches, deviations, SOPs, CAPAs, regulatory submissions, and study outcomes. Data that was previously stored independently in Amazon Redshift, Benchling, and indexed in Pinecone was enriched with structured entity identifiers and relationship mappings. The system was able to traverse modeled relationships to support more complex queries, for example, linking a deviation to its associated batch record, related assay performance trends, and subsequent CAPA documentation across time.

The Llama 3 orchestration layer was expanded to manage more complex questions involving relationships between data points. For example, if a user asked about recurring stability issues linked to a specific compound across multiple manufacturing campaigns, the system could search connected records and identify patterns before assembling the approved information into a response. Time-based filtering was also added, allowing users to view information in chronological order, which was especially helpful for regulatory preparation and long-term quality trend analysis.

Information from these connected data sources was combined with relevant documents retrieved from Pinecone and, when needed, structured data or calculations from Redshift. As in earlier phases, only content that met policy and access rules was sent to GPT-4. GPT-4 continued to serve strictly as an explanation layer, converting structured data and identified relationships into clear, citation-supported summaries.

During this phase, improving system performance was also a priority. As user questions began to span multiple departments and data sources, the team optimized how information was stored and searched to keep response times fast, even as usage increased. Improvements were made to the structuring of data relationships, and temporary storage (caching) was introduced within the AWS environment to accelerate repeated queries. Logging through AWS CloudWatch was also expanded to capture these more complex searches, designed to maintain traceability and inspection-readiness as its capabilities increased.

Although this phase was not essential for delivering immediate operational value, it marked a shift from a compliant, retrieval-focused platform to a more scalable, long-term knowledge system. By linking information across R&D, manufacturing, quality, and regulatory domains, the organization enabled a deeper understanding of historical trends and cross-functional relationships. This phase introduced architectural components that support broader, cross-domain knowledge retrieval.

Phase 5: Validation, SOPs, and Steady State

Phase 5 marked the transition from AI development to operating a regulated and validated system within a GxP framework. While the platform's core architecture remained stable, the primary focus shifted to governance, compliance, and operational discipline. AWS CloudWatch was utilized to establish structured logging and audit trails in accordance with Computer System Validation (CSV) standards. This supported comprehensive traceability of user activities, model iterations, and system actions, thereby providing reproducibility and enabling full audits to meet compliance objectives.

Change control processes were applied across all components of the system. Updates to the closed reasoning model (GPT-4) and the open orchestration model (Llama 3) underwent formal risk assessments, regression testing, and approval workflows before deployment. Similarly, prompt templates, data ingestion pipelines, and retrieval mechanisms were version-controlled and tested to help prevent unintended consequences. These controls extended to metadata tagging, document parsing logic, and policy enforcement configurations, ensuring any modifications complied with validated procedures, which maintained system integrity.

Regular internal audits and mock inspections become integral to operations, verifying that the system consistently enforces policy decisions and generates output based solely on authorized data. By embedding this rigorous validation and change management process, the organization intended to ensure the AI platform operated reliably and compliantly as part of regulated workflows. This phase prioritized stability and traceability over rapid innovation, providing confidence to stakeholders that the system could withstand regulatory scrutiny while delivering ongoing value.

Conclusion

This case study demonstrates that deploying AI in a GxP-regulated CGT environment is not a single technology project. Instead, it is a governed, phased development process. By starting with controlled retrieval and auditability, then progressively hardening access controls, delegating analytics to deterministic tools, and expanding toward structured knowledge modeling, the organization was able to reduce risk while steadily increasing value and expanding the use cases the platform could manage.

Rather than pursuing autonomy early, the company prioritized inspection defensibility, traceability, and clear system boundaries from the start. Each phase deliberately constrained the scope so that no write-back, automated decisions, or uncontrolled analytics were done until the proper governance, logging, and validation controls were in place. This staged approach allowed QA and Regulatory to remain co-owners of the platform, and they were invested partners rather than downstream reviewers.

By Phase 3, most of the operational value was realized through deterministic analytics and structured outputs, proving that compliance and innovation are not mutually exclusive when architecture and governance evolve together.

By treating AI as a regulated system from day 1, the organization avoided rework, maintained QA buy-in, and built a foundation capable of evolving and scaling into increasingly sophisticated use cases without triggering major architectural overhauls.

Key Lessons Learned

- Focus on information retrieval and traceability to improve access to knowledge sources, not on automation.
- Define explicit boundaries for the AI system before starting buildout and ensure no write-back or decision-making authority is given to the system.
- Governance must precede everything else; role-based access control, lifecycle gating, citation enforcement, and logging were prerequisites before expanding into analytics.
- Ensure comprehensive logging of queries, tool calls, inputs, outputs, and model versions to create a controlled knowledge platform.
- Delegate reasoning to tools specifically built for that purpose, such as statistical analysis to SQL and other API based tools, to reduce AI model risk. This also increases credibility with QA and CSV.
- Ongoing AI validation, SOP management, and inspection reference are essential in a regulated environment.
- Scaling knowledge graphs to enable cross-domain reasoning should occur only after foundational trust and governance mechanisms are established in the system.

References

1. OpenAI. (n.d.). *OpenAI API documentation.* OpenAI. https://platform.openai.com/docs/
2. Meta AI. (2023, October 11). *Large language model: Llama 3.* Meta AI. https://ai.facebook.com/blog/largelanguage-model-llama-3/
3. Llama Learning Platform. (2025, April 14). *Llama 3: The state-of-the-art open-source large language model. Llama 3: The State-of-the-Art Open-Source Large Language Model*
4. Amazon Web Services. (n.d.). *GxP-compliant systems on AWS.* AWS. https://aws.amazon.com/compliance/gxps/
5. Amazon Web Services. (n.d.). *AWS Identity and Access Management (IAM) User Guide.* https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html
6. U.S. Food and Drug Administration. (2007). *Guidance for industry: Computerized systems used in clinical investigations.* FDA. https://www.fda.gov/media/85183/download
7. U.S. Food and Drug Administration. (2021). *Artificial intelligence and machine learning in software as a medical device (SaMD).* FDA. https://www.fda.gov/media/145022/download
8. Pinecone. (n.d.). *Vector databases for retrieval augmented generation.* https://www.pinecone.io/
9. European Medicines Agency (EMA). (n.d.). *Guiding principles for good AI practice in drug development.* EMA. [PDF]
10. Ai, Y., Huang, D.S., Pan, Y., Chen, W., Li, B. (2025). Distilling closed-source LLM knowledge for locally stable and economic biomedical entity linking. In *Advanced Intelligent Computing Technology and Applications* (Lecture Notes in Computer Science, vol. 15866). Springer, Singapore. https://doi.org/10.1007/978-981-95-0027-7_9
11. Farquhar, S., Kossen, J., Kuhn, L., et al. (2024). Detecting hallucinations in large language models using semantic entropy. *Nature*, 630, 625–630.