CloudGeometry INSIGHTS

Building an AI Agent for Unified Advertising Intelligence With Databricks and Mosaic AI

Technical Case Study: Integrating End-to-End Ad Performance Data and Narrative Context

A CloudGeometry Insights Research Note

EXECUTIVE SUMMARY

Advertising and marketing teams often rely on dashboards that show what happened – metrics like clicks, spend, return on ad spend (ROAS), and customer acquisition cost (CAC). But they still struggle to understand *why* performance changed. Insights into strategic intent are usually buried in unstructured materials like campaign briefs, creative presentations, and post-mortems. These traditional tools don't connect structured data with narrative content.

Business questions such as "ROAS dropped 15% – was it due to underperforming creative in the Summer Sale campaign?" or "What's the Q4 impact if we reallocate budget to better-performing channels?" require analysis that crosses these data types. Standard BI tools can't answer them.

This paper presents a practical solution: an "Advertising Intelligence Agent" built on Databricks using Mosaic AI. It combines structured performance data with unstructured documents using a multi-tool Retrieval-Augmented Generation (RAG) approach. The agent can access dashboards, analyze documents, and run predictive models, returning answers in natural language.

This represents a shift for business users – from static dashboards to interactive conversations with data. The unified Databricks platform allows teams to securely manage data, build intelligent tools, and deploy production-ready AI agents. The result is faster answers, better media investment decisions, and improved strategic alignment.

The document outlines how to build this system, including its data architecture, governance model, AI tooling, and deployment pipeline. For agency leaders, the business value lies in gaining actionable insight – not just metrics – across every stage of a campaign lifecycle. This paper describes four elements of a successful technical design for building such a system:

- **The Lakehouse Foundation:** Establishing a robust data architecture.
- **Unified Governance:** Securing and managing all data and AI assets.
- **AI Agent Architecture:** Constructing the intelligent agent and its tools.
- **Production and Evaluation:** Deploying and maintaining a trustworthy AI system.

The Lakehouse Foundation: A Medallion Architecture for Advertising Analytics

The foundation of any trustworthy AI application is a reliable, high-quality data source. An AI agent that provides confident but incorrect answers due to poor data quality is more detrimental than no agent at all. To mitigate this risk, our architecture employs the Medallion Architecture, a data design pattern that progressively refines data through a series of quality-gated layers: Bronze (raw), Silver (both cleansed and conformed), and Gold (business-ready).

> The medallion approach tames messy feeds; legacy ETL can only chase them.

This structured approach prevents the formation of a "data swamp" and ensures that the data feeding the AI agent is curated, consistent, and reliable. The entire architecture is built upon Delta Lake, an open-source storage layer that brings ACID transactions, schema enforcement, and time travel capabilities to the data lake, forming the core of the Databricks Lakehouse.

The Bronze Layer: The Historical Archive

The Bronze layer serves as the initial ingestion point and historical archive for all source data. Its primary principle is to capture raw, immutable data exactly as it arrives from source systems, preserving a complete and untampered record for auditing, compliance, and potential reprocessing.

Process and Implementation: Data is ingested from various sources, such as advertising platform APIs (Google, Facebook), CRM systems (Salesforce), and file stores containing documents, using automated tools like Lakeflow Connect or Auto Loader. These tools can efficiently handle both batch and streaming data, incrementally loading new files or events into Bronze Delta tables. To protect against pipeline failures caused by unexpected schema changes from source APIs, a best practice is to land semi-structured data, like JSON payloads, into a single *VARIANT* or *STRING* column. This ensures that no data is dropped. Additional metadata columns, such as `_ingest_timestamp` and `_source_filename`, are added to each record to provide essential data lineage from the very beginning.

The Silver Layer: The Single Source of Truth

The majority of data transformation and cleansing occurs in the Silver layer. Data from the Bronze layer is refined, conformed, and enriched to create a set of validated, queryable tables that serve as the enterprise's single source of truth for advertising data. These transformations are orchestrated using Databricks SQL and PySpark within Lakeflow Declarative Pipelines (formerly Delta Live Tables), which declaratively manage data dependencies and quality expectations.

`Implementation for Structured Data:` For structured data, the process involves cleaning (e.g., handling null values, standardizing currency codes), deduplicating records, and joining datasets from disparate sources. A critical step is creating conformed dimensions. For instance, campaign data from Google Ads and Facebook Ads, which have different schemas, are merged into a single `dim_campaigns` table with a unified structure. This creates a central `fact_performance` table containing key metrics like impressions, clicks, spend, and conversions, linked to conformed dimensions for campaigns, channels, and creatives.

`Implementation for Unstructured Data:` For unstructured narratives, a separate pipeline is built to process documents like campaign briefs and post-mortem reports. This pipeline extracts raw text from various file formats (PDF, DOCX, PPTX), chunks the text into manageable paragraphs or sections, and stores this content alongside its metadata (e.g., source document name, campaign ID) in a structured Delta table, such as `docs_campaign_briefs`. This table is now ready for downstream embedding and vectorization processes.

The Gold Layer: The Consumption Layer

The Gold layer contains highly refined, aggregated data products designed for specific business use cases and optimized for consumption by analysts and AI applications. These tables are often denormalized to reduce query latency and are modeled to directly answer key business questions.

`Process and Implementation:` The Gold layer is built from the validated data in the Silver layer. The architecture uses materialized views to accelerate performance for common queries.

For example, `mv_weekly_performance_summary` can be created to pre-calculate weekly KPIs, ensuring that dashboard queries or agent requests for this data are served instantly without expensive re-computation. Other Gold tables might calculate high-level business metrics like `agg_roas_by_campaign` or `agg_cac_by_channel`. These tables provide the clean, reliable, and performant data that the AI agent will query to answer quantitative questions. While the Medallion architecture provides a robust framework, it is a design principle, not a rigid dogma.

Some critics note this introduces latency. That said, in such a complex, multi-source use case, the separation of concerns and enforced quality gates are essential for building a trustworthy system. The key is to apply the layers pragmatically where they add value. *Table 1* below provides a concrete, field-level blueprint of the data transformation process, illustrating how data evolves through the Medallion layers.

> Medallion architecture turns raw clicks into decision-ready gold-quality data via a three-phase process.

Layer	Table Name	Field Name	Data Type	Description / Transformation Logic	Source Field(s)
Bronze	raw_platform_spend	payload	VARIANT	Raw JSON object from ad platform API.	API Endpoint
Bronze	raw_campaign_narratives	file_content	BINARY	Raw binary content of an uploaded document (PDF, DOCX).	File Upload
Silver	fact_performance	spend_usd	DECIMAL(18,2)	Cleaned, standardized to USD, cast from string.	raw_platform_spend.payload:spend
Silver	fact_performance	campaign_key	BIGINT	Foreign key to dim_campaigns.	raw_platform_spend.payload:campaign_id
Silver	docs_campaign_briefs	chunked_text	STRING	Text extracted from file_content and split into paragraphs.	raw_campaign_narratives.file_content
Gold	agg_roas_by_campaign	total_roas	DECIMAL(10,2)	SUM(revenue) / SUM(spend_usd).	fact_performance.revenue, fact_performance.spend_usd
Gold	mv_weekly_performance_summary	avg_cpc	DECIMAL(10,2)	SUM(spend_usd) / SUM(clicks) aggregated by week.	fact_performance.spend_usd, fact_performance.clicks

Table 1

Unified Governance: Securing and Managing the AI-Ready Data Estate

Governance is a critical enabler of secure, scalable, and trustworthy AI in a modern data platform. It cannot be limited to passive data hygiene.

> Brilliant agents still fail without audit-grade governance baked in from day one.

The Open Source [Unity Catalog (UC)]() provides the central governance solution for the Databricks platform, providing a unified framework to manage all data and AI assets. For the Advertising Intelligence Agent, UC is the connective tissue that links the Medallion data foundation to the Mosaic AI agent. It ensures every action is secure, auditable, and compliant.

This unified approach is a significant departure from traditional governance models, which often struggle to manage AI assets alongside data.

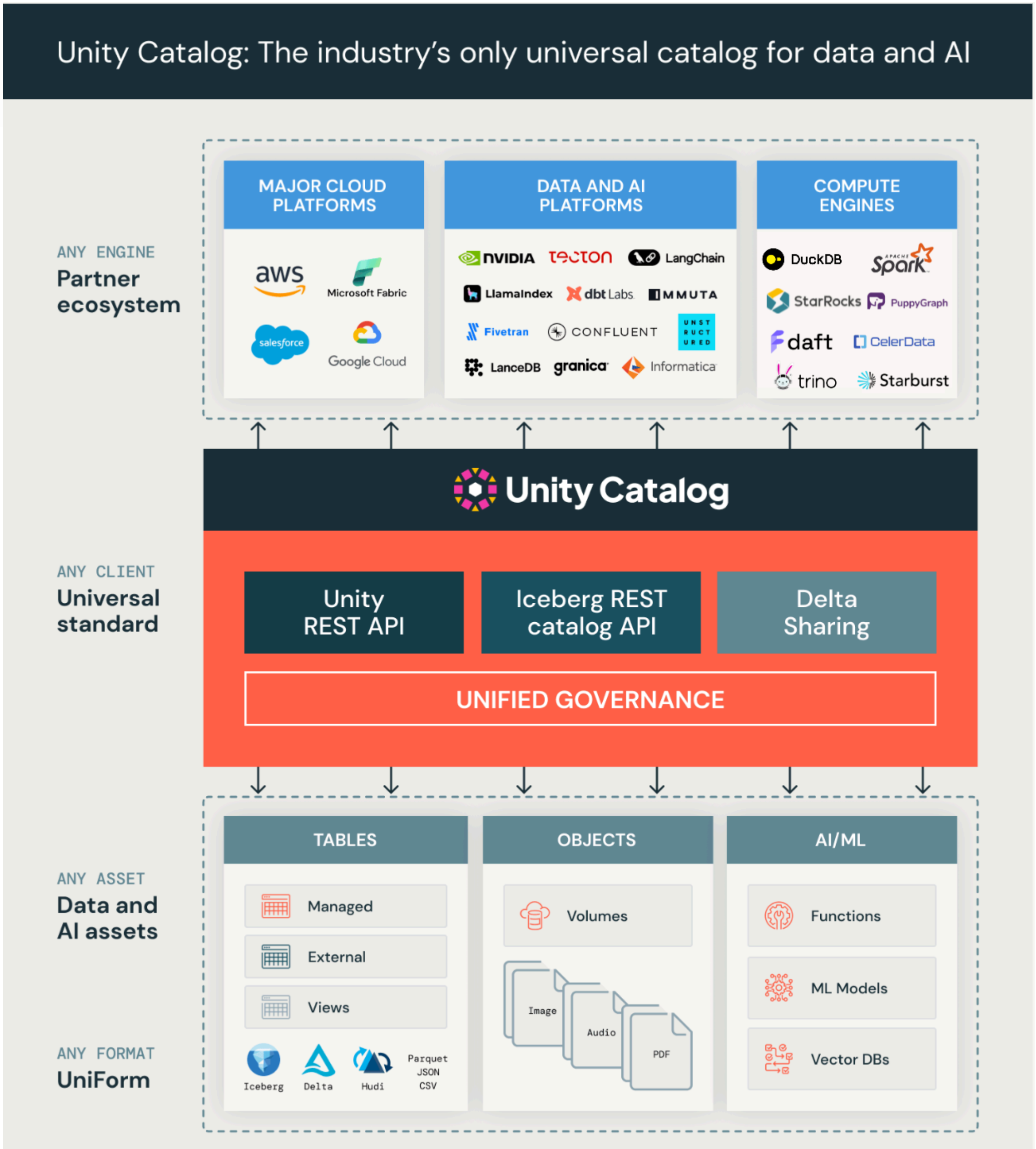
The Three Pillars of AI Governance with Unity Catalog

Unity Catalog is an open-source project (Apache 2.0) introduced as the industry's only universal catalog for both data and AI (Figure 1, below). It provides a standardized API layer for managing structured tables, unstructured files, and machine learning assets across any cloud platform or compute engine.

Unity Catalog cuts across format lines: SQL rows, PDFs, and vectors under one roof.

The utility and versatility it offers are hard to overstate, as UC supports any client, integrates with modern ecosystems, and handles all data formats to enable unified governance and discoverability for all enterprise data and AI assets.

- `A Unified Catalog for All Assets:` Unity Catalog provides a single, three-level namespace (catalog.schema.object) to organize and govern not only Delta tables but also ML models, functions, and other AI assets. This is foundational for our agent. We create a dedicated schema `advertising_gold.agent_ tools`, to house the custom functions, or "tools," that the agent will use. This act of registration transforms a piece of code in a notebook into a governed, discoverable, and reusable enterprise asset. A Python function for forecasting is no longer a one-off script; it becomes `advertising_gold.agent_tools. forecast_budget`, an asset with an owner, permissions, and a clear lineage that can be leveraged by other teams and applications, fostering productivity and breaking down organizational silos.
- `Fine-Grained Access Control:` Security is managed through standard `ANSI SQL GRANT` and `REVOKE` statements, providing a familiar interface for administrators. For this architecture, a `marketing_analysts` group might be granted `SELECT` access only on the Gold layer tables, while the `data_engineers` group has `MODIFY` privileges on the Silver layer. Crucially, the AI agent operates using a dedicated service principal. This principal is granted a minimal set of permissions: `EXECUTE` on the functions within the `agent_tools` schema and `SELECT` on the specific Gold tables it needs to query. This principle of least privilege is enforced end-to-end. When the agent calls a tool, the entire execution chain – from the agent's identity to the function call to the data access – is governed and audited by Unity Catalog.
- `AI-Powered Discovery and Lineage:` Unity Catalog is infused with intelligence to simplify data management. Its automated, column-level lineage capability is indispensable for building trust and enabling debuggability. If the agent produces an unexpected result, its data sources can be traced back to the raw Bronze layer in seconds. Furthermore, AI-powered documentation can automatically generate descriptive comments for tables and columns. This not only reduces the manual burden of cataloging but also enriches the semantic context available to both human users and the AI agent itself, helping it better understand the data it is querying.



Source: https://github.com/unitycatalog/unitycatalog/blob/main/README.md

Figure 1: *Unity Catalog is an open standard designed to simplify how organizations manage data and AI assets across teams and platforms. With a unified structure for organizing information, it enforces clear access controls and tracks data usage for auditability. For business leaders, this means faster access to reliable insights, improved compliance with data policies, and reduced operational risk.*

Architecting a Generative AI Agent with Mosaic AI

With a governed, high-quality data foundation in place, the next step is to construct the AI agent itself. This isn't a simple chatbot; it is a compound pattern to achieve contextual intelligence. The agent must intelligently reason about a user's query and decide which of its specialized tools—or a combination of tools—is required to formulate an answer. This entire process is orchestrated using the Mosaic AI Agent Framework, with the powerful DBRX Instruct model serving as the central reasoning engine.

The Hybrid, Multi-Tool RAG Pattern

A standard RAG architecture narrows the data set from a Large Language Model (LLM) in greater specificity, contrasting it with relevant factual data, to reduce hallucinations and improve the trustworthiness of its responses. Our application advances that pattern by equipping the agent with a versatile, multi-component toolkit so it can readily contend with different data formats and still perform complex computations. The agent's p[rome directive: to decompose a user's query and select the appropriate tool:

- For **quantitative** questions about performance ("What was our cost per click last week?"), it uses the *SQL Data Analyst* tool.
- For **qualitative** questions about strategy ("What was the target audience for the 'Summer Sale' campaign?"), it uses the *Document Researcher* tool.
- For **predictive** questions ("What is the forecasted impact of a budget change?"), it uses the *Predictive Forecaster* tool.

Databricks Mosaic AI Agent Framework provides the components to build, evaluate, and deploy production-quality agents – making multi-step reasoning and tool orchestration possible.

Building the Agent Toolkit in Unity Catalog

Each tool is implemented as a Python function and registered in Unity Catalog, turning it into a governed, callable asset. The docstrings for these functions are critical, as they serve as the instructions that the LLM uses to understand each tool's capability and its parameters.

Tool 1: The SQL Data Analyst

This *SQL Data Analyst* enables the agent to query structured data in the Gold layer tables. For maximum security and control, it is implemented as a pre-defined SQL User-Defined Function (UDF) in Unity Catalog that accepts specific parameters, which prevents the LLM from generating/executing arbitrary SQL.

Implementation (SQL):

```sql
CREATE OR REPLACE FUNCTION
advertising_gold.agent_tools.get_campaign_kpis(

  campaign_name_filter STRING COMMENT 'The
name of the campaign to filter by, e.g.,
''Summer Sale 2024''.'

)

RETURNS STRING

COMMENT 'Returns key performance indicators
(KPIs) for a specific campaign, including
total spend, clicks, conversions, and
ROAS.'

RETURN

  SELECT

    TO_JSON(

      NAMED_STRUCT(

        'campaign_name', campaign_name,

        'total_spend_usd', SUM(spend_usd),

        'total_clicks', SUM(clicks),

        'total_conversions',
SUM(conversions),

        'roas', SUM(revenue) /
SUM(spend_usd)

      )

    )

  FROM
advertising_gold.agg_roas_by_campaign

  WHERE campaign_name =
campaign_name_filter

  GROUP BY campaign_name;
```

Tool 2: The Document Researcher

The *Document Researcher* tool performs a semantic search over the unstructured text across campaign documents. It is built using Mosaic AI Vector Search, which provides a highly scalable and performant vector database that automatically syncs with a source Delta table.

Implementation (Python):

`Data Preparation:` The `docs_campaign_briefs` table from the Silver layer, which contains chunked text, is the source.

`Embedding Generation:` A foundation model (e.g., `databricks-bge-large-en`) is used to create vector embeddings for each text chunk.

`Vector Search Index:` A Vector Search endpoint and index are created. The index is configured for `DELTA_SYNC`, meaning it will automatically update as new documents are added to the source table.

`Tool Function:` A Python function is created to query this index. To improve quality, this tool incorporates a re-ranker. After the initial vector search retrieves the top N documents, a more sophisticated cross-encoder model re-ranks these results for relevance to the specific query, ensuring the most pertinent context is passed to the LLM. This function is then registered in Unity Catalog.

Tool 3: The Predictive Forecaster

This *Predictive Forecaster* tool brings classical machine learning to the generative AI workflow. Thus, the agent can generate forecasts.

Implementation (Python):

- `Model Training:` A time-series forecasting model, such as SARIMAX, is trained using the `statsmodels` library on the historical data in the `mv_weekly_performance_summary` Gold table. The model is trained to predict metrics like clicks or conversions based on past trends, seasonality, and exogenous variables like spend.

- `Model Logging:` The trained model is logged to the MLflow Model Registry and registered in Unity Catalog, making it a governed asset.
- `Tool Function:` A Python function is created that loads the trained model, accepts inputs like `budget_change_pct` and `forecast_horizon_days`, generates a prediction, and returns the forecast as a JSON object. This function is then registered as `advertising_gold.agent_tools.forecast_performance` in Unity Catalog.

The core innovation of this architecture is the seamless composition of these diverse tools – SQL queries, vector search, and classical ML. Answering a complex query requires the agent to chain these tools together: first, query SQL to get a campaign's budget, then search documents to understand its strategy, and finally, call the forecast model with the retrieved information.

Orchestration and Debugging with the Agent Framework

The agent's reasoning loop is built using an authoring framework like LangGraph, which defines the agent's state and the logic for transitioning between calling the LLM and executing tools. The DBRX Instruct model serves as the agent's brain, interpreting the user prompt and deciding which tool to call based on the functions' docstrings.

Observability is paramount for debugging and improving the agent. The framework's integration with MLflow Tracing provides complete visibility into the agent's execution flow. A trace view in the MLflow UI visualizes the entire sequence for a complex query, showing the initial prompt, the LLM's thought process, the exact tool calls with their inputs and outputs, and the final synthesized response. This level of detail is essential for senior practitioners to diagnose issues and optimize performance.

Table 2 below specifies the API contract for each tool, which is critical for both development and guiding the LLM's behavior. It provides the agent skillset, codified – no guesswork, just callable tools.

Productionization, Evaluation, and Continuous Improvement

A proof-of-concept in a notebook is far from a production-ready system. The final stage of this architecture addresses the critical "last-mile" challenges of deploying, evaluating, and continuously improving the agent using a robust LLMOps lifecycle. Databricks provides a suite of integrated tools that demonstrate a mature understanding of the operational realities of managing AI systems.

Deployment with Mosaic AI Model Serving

The first step in productionization is to deploy the agent as a scalable, secure REST API. This is accomplished using Mosaic AI Model Serving. The entire agent, including its code and dependencies, is logged as a single object using mlflow.pyfunc.log_model. This MLflow model is then deployed to a serverless model serving endpoint with a few clicks. This endpoint provides a production-ready API that can be integrated into front-end applications or other internal services, and it automatically handles scaling based on load.

Tool Name	UC Function Name	Description (for LLM)	Input Parameters (Name, Type, Description)	Return Value (Type, Description)
SQL Analyst	advertising_gold.agent_tools.get_campaign_kpis	Executes a query to retrieve key performance indicators for a specific campaign. Use this for questions about spend, clicks, conversions, or ROAS.	campaign_name_filter: STRING, "The exact name of the campaign to query."	STRING, "A JSON formatted string containing the campaign's KPIs."
Document Researcher Azure Key Vault, GCP Secret Manager)	advertising_gold.agent_tools.forecast_performance	Forecasts future performance metrics based on historical data and potential budget scenarios. Use for questions about future outcomes.	metric_to_forecast: STRING, "e.g., 'clicks', 'conversions'"; budget_change_pct: FLOAT, "e.g., 20.0 for a 20% increase"; horizon_days: INT, "e.g., 90 for a quarterly forecast."	STRING, "A JSON string with the forecasted time series data."
Predictive Forecaster	advertising_gold.agent_tools.forecast_performance	Forecasts future performance metrics based on historical data and potential budget scenarios. Use for questions about future outcomes.	metric_to_forecast: STRING, "e.g., 'clicks', 'conversions'"; budget_change_pct: FLOAT, "e.g., 20.0 for a 20% increase"; horizon_days: INT, "e.g., 90 for a quarterly forecast."	STRING, "A JSON string with the forecasted time series data."

Table 2

Quality Assurance with Mosaic AI Agent Evaluation

Ensuring the agent's quality and preventing performance degradation over time is arguably the most critical aspect of LLMOps. Mosaic AI Agent Evaluation provides a comprehensive framework for this purpose. The evaluation strategy involves three key components:

- `Creating a "Golden" Evaluation Set:` A curated dataset of representative business questions and ideal, expert-written answers is created. This set serves as the ground truth for measuring performance.
- `LLM-as-a-Judge:` A powerful foundation model (e.g., GPT-4 or DBRX) is used as an impartial "judge" to automatically score the agent's responses against the golden set. This evaluation measures key quality dimensions such as Faithfulness (is the answer fully supported by the retrieved context?), Answer Relevance, and Context Recall (did the retrieval step find all the necessary information?). This automated process can be integrated into a CI/CD pipeline to catch regressions before they reach production.

- Human-in-the-Loop Feedback: Technology alone cannot define quality. The Agent Evaluation Review App provides a simple UI for business domain experts – the marketing managers and analysts who will actually use the agent – to review, score, and provide qualitative feedback on responses without writing any code. Feedback is captured in a Delta table and provides key insights into where the agent is succeeding or failing in real-world scenarios. This structured feedback loop between business users and the AI team is the primary driver of the agent's long-term improvement.

Monitoring with Lakehouse Monitoring

Once deployed, the agent's performance is continuously tracked using Lakehouse Monitoring. This service automatically creates a monitoring dashboard that tracks data and model quality over time. For the Advertising Intelligence Agent, this includes monitoring input prompts for data drift (e.g., are users asking new types of questions the agent wasn't trained to handle?), tracking token consumption for cost management, and analyzing the statistical properties of the generated responses to detect any degradation in quality. Alerts can be configured to notify the team of any significant deviations, enabling proactive maintenance.

Metric	*Definition*	*How to Measure*	*Target Threshold*
Faithfulness	Does the agent's answer contradict the provided context? Is every part of the answer fully supported by the retrieved information?	LLM-as-a-judge prompt asking the judge to verify the answer against the provided context on a binary scale.	> 95%
Answer Relevance	Is the agent's answer directly relevant to the user's question? Does it fully address what was asked without including extraneous details?	LLM-as-a-judge prompt or human rating on a 1-5 scale.	> 4.5/5
Context Recall	Did the retrieval step (Document Researcher tool) fetch all the necessary documents/chunks required to fully answer the question?	Compare retrieved documents against a pre-annotated "golden" set of relevant documents for each test query.	> 90%
Tool Execution Accuracy	Did the agent call the correct tool with the correct parameters for a given query?	Parse MLflow traces to check tool calls and parameters against an expected sequence for a given test query.	> 98%

Table 3

Table 3 above shows how we operationalized the concept of quality by defining a concrete set of KPIs for the AI application. Trust isn't a feeling; these KPIs quantify it. This framework is essential for setting project goals, creating automated tests, and reporting on performance to leadership.

Conclusion and Strategic Recommendations

In this Technical Case Study, I've detailed an end-to-end architecture for building a sophisticated Advertising Intelligence Agent on the Databricks platform. The journey spans from establishing a reliable data foundation with the Medallion Architecture on Delta Lake, to implementing robust governance with Unity Catalog, to constructing a multi-tool generative AI agent with Mosaic AI, and finally, to deploying and maintaining it with a mature LLMOps framework.

The feasibility of such a compound AI system is a direct result of the Databricks Data Intelligence Platform's unified nature. Attempting to stitch together these capabilities – data engineering, data warehousing, data governance, classical ML, and generative AI – from disparate point solutions would create a system that is brittle, insecure, and operationally unmanageable. The seamless integration of these components within a single platform is what makes this advanced application practical for enterprise deployment.

For technical leadership planning to embark on a similar initiative, the following strategic recommendations should be considered:

- `Foster a Cross-Functional Team Structure:` Projects of this complexity demand tight collaboration. The ideal team is a "pod" comprising data engineers (to build the Medallion pipelines), data scientists (to develop the forecasting models and text processing logic), and ML engineers (to architect the agent and manage the LLMOps lifecycle). The unified Databricks platform is designed to support this collaborative model.
- `Proactively Manage Costs:` Databricks is a powerful, consumption-based platform that requires diligent cost management to be effective. Best practices are essential: leverage serverless compute for data engineering and model serving to pay only for what is used, use job clusters instead of all-purpose clusters for production ETL workloads to avoid idle compute costs, optimize data layout with techniques like liquid clustering to reduce query scan costs, and utilize Lakehouse Monitoring for comprehensive cost tracking and budgeting.
- `Adopt a Phased Implementation Roadmap:` A "big bang" approach is risky. A phased implementation delivers value incrementally and allows the team to learn and adapt. A recommended roadmap is:
  - `Phase 1 (Sprints 1-2):` Focus on the data foundation. Build and validate the Bronze, Silver, and Gold Medallion pipelines for the core structured advertising and CRM data.
  - `Phase 2 (Sprints 3-4):` Deliver initial value. Develop and deploy the "SQL Analyst" tool and a basic agent that can answer quantitative questions. This provides a tangible result to business users early in the process.
  - `Phase 3 (Sprints 5-6):` Incorporate qualitative context. Build the pipeline for unstructured documents and implement the "Document Researcher" tool using Mosaic AI Vector Search.

- Phase 4 (Sprints 7+): Add advanced capabilities and mature the process. Integrate the "Predictive Forecaster" tool and establish the full LLMOps evaluation and monitoring loop with human feedback.

Ultimately, the successful implementation of an Advertising Intelligence Agent is as much an organizational and cultural endeavor as it is a technical one. The Databricks platform provides the technological catalyst, but realizing its full potential requires a commitment to data quality, a breakdown of silos between technical and business teams, and an iterative, product-centric mindset toward building and improving AI applications.

Sources

1) Databricks Data Intelligence Platform Reviews 2025 - BARC research
2) 3 Databricks Mosaic AI Use Cases to Supercharge Your Log Analytics Program
3) Databricks IQ: AI-Driven Analytics for Faster Data Insights / Databricks
4) Mosaic AI Agent Framework / Databricks
5) RAG (Retrieval Augmented Generation) on Azure Databricks – Azure
6) Databricks Data Intelligence Platform: Serverless Data Engineering in the Age of AI
7) Databricks Data Intelligence Platform / element61
8) Understanding the Three Layers of Medallion Architecture - ER/Studio
9) What is a Medallion Architecture? – Databricks
10) Databricks Delta Lake 101: A Comprehensive Guide (2025) - Chaos Genius
11) Lakehouse Storage - Databricks
12) What is the medallion lakehouse architecture? - Databricks Documentation
13) What is Azure Databricks? - Azure Databricks / Microsoft Learn
14) Guide to Building ETL Pipelines in Databricks - Airbyte
15) Data engineering with Databricks
16) What is the medallion lakehouse architecture? - Azure Databricks ...
17) Databricks for Data Engineers: The Ultimate Step-by-Step Guide
18) Databricks ETL Framework: Best Practices for DLT Pipelines
19) Medallion Architecture: Key Concepts and Examples - DataForge
20) Data Products: A Case Against Medallion Architecture by Modern Data 101 / Medium /
21) The Case Against Centralized Medallion Architecture / Towards Data Science
22) Data Products: A Case Against Medallion Architecture : r/dataengineering - Reddit
23) Unity Catalog / Databricks
24) What is Unity Catalog? - Azure Databricks / Microsoft Learn
25) What is Unity Catalog? - Databricks Documentation
26) AI-Powered Documentation, Search and Discovery - Databricks
27) DBRX: A New Standard for Open Source LLMs - Databricks
28) RAG (Retrieval Augmented Generation) on Databricks
29) AI Agents on Mosaic AI in 5 Minutes / Databricks
30) Build and deploy quality AI agent systems - Databricks
31) Structured retrieval AI agent tools - Databricks Documentation
32) Improve RAG chain quality / Databricks Documentation
33) Advanced RAG Techniques - DataCamp
34) ARIMA Model - Complete Guide to Time Series Forecasting
35) Complete Guide To SARIMAX in Python - GeeksforGeeks
36) Manage model lifecycle in Unity Catalog - Databricks Documentation
37) User-defined functions (UDFs) in Unity Catalog - Databricks Documentation
38) Tutorial: Build, evaluate, and deploy a retrieval agent - Azure Databricks / Microsoft Learn
39) Tutorial: Build, evaluate, and deploy a retrieval agent / Databricks ...
40) DBRX - Wikipedia
41) databricks/dbrx-base - Hugging Face
42) Get started with AI agents / Databricks Documentation
43) Get started with AI agents - Azure Databricks / Microsoft Learn
44) AI and machine learning on Databricks

About the Author

Irina Groilova, Cloud Native Backend Team Lead at CloudGeometry, brings over a decade of experience in scalable application development, cloud services, and AI integration. With a strong foundation in Python, Node.js, and AWS, she leads cross-functional teams in designing and delivering reliable backend systems for telecom and cloud service projects. Irina holds AWS certifications in Solutions Architecture and Machine Learning, and has deep technical expertise across Databricks, Kubernetes, and CI/CD pipelines. Known for her hands-on leadership, mentoring, and sharp architectural insight, she plays a key role in driving high-performance, cloud-native solutions aligned with business goals.

CloudGeometry Insights is our applied research initiative where architecture decisions, security tradeoffs, and emerging tech are dissected by our engineers who *live* in production. This isn't editorial. It's strategic intelligence from practitioners shipping real systems at scale – the kind of thinking we bring to client roadmaps: shared openly, challenge-ready, and rooted in field experience. If you're a CTO, CIO or founder trying to separate signal from noise, this is where we publish what's worth your time.

Want to hear from our technical experts on a topic of particular interest? Contact us via insights@cloudgeometry, and let us know what you want to learn more about. If we've already written on the subject, we will send it on; if not, we will let you know and add it to our research agenda for future publication.

ABOUT CLOUDGEOMETRY

CloudGeometry delivers expert technical services, helping our clients unlock the full potential of cloud-native open-source tooling and commercial platform technologies. As an AWS Advanced Consulting partner, our certified solution architects and platform engineers help address the range of challenges facing enterprise innovators and venture-funded startups alike. The Cloud Native Computing Foundation has accredited us as a Kubernetes Certified Service Provider.

- Foundation Services solidify, simplify, and modernize your tech foundations: healthier systems, streamlined migrations, and robust uptime for future-ready growth.
- Advanced Services provides software development teams with proven strategies and tactics for better technical & commercial outcomes for a clear path to cloud-native velocity
- AI/ML & Data Services help you apply the latest in AI and data analytics technology to your crucial business processes: MLOps, Generative AI, AI/ML Development. As a Databricks Systems Integrator, our data engineering experts ensure your data is accurate, accessible, and ready to fuel decision-making.

For the last decade, we've built and deployed hundreds of big, fast apps backed by scale-out infrastructure across a range of industries. Working across a wide variety of applications – Financial Services, Industrial Automation, HIPAA-compliant Healthcare, AdTech, Consumer-grade Mobile, IoT and smart devices, among others – has given us valuable insights into essential full-stack application patterns.

Learning from these similarities lets us develop adaptive, flexible, technology-driven solutions even faster. It's why we created **CGDevX**, our free, open-source hybrid-cloud application delivery platform. By automating many of the routine tasks associated with software development, CGDevX gives you a shorter, more effective path for engineering teams to focus on creating value-added features and innovations.

Count on us to accelerate application modernization, Kubernetes adoption, developer enablement, secure multi-tenancy, DevOps automation and more. With roots in Silicon Valley, we've seen firsthand what works (and what doesn't). We've completed hundreds of migrations and application modernization projects, so we're well-versed in solving for rising costs and evolving technology and helping clients choose solutions that prevent needless vendor lock-in.

We'll help you find the right technology for your business goals and budget, and we'll integrate the right tools with your Software Development and IT processes, all backed by a complete portfolio of ongoing application development and full-stack support services. From enterprise upgrades to cloud-native scale-out to practical applications of AI, CloudGeometry helps plot the shortest path across all dimensions of modern cloud software and data engineering.

aws partner network
ADVANCED CONSULTING PARTNER

databricks

kubernetes CERTIFIED SERVICE PROVIDER

LINUX FOUNDATION
AI & DATA WORKSTREAMS