How TrustMyApps improved data quality and AI-powered insights for safer digital parenting

TrustMyApps empowers parents to make informed decisions about their children's digital experiences by ranking educational and child-friendly apps. However, their assessment data was unstructured, inconsistent, and difficult for large language models (LLMs) to process reliably.

Calibo partnered with TrustMyApps on a two-week pro bono proof of concept (PoC) to address these challenges. The engagement focused on two key initiatives:

LLM-Optimized Data Transformation

Converting raw assessment data into structured, queryable markdown summaries grouped by severity.

Data Quality and Verification

Implementing automated integrity checks, inconsistency detection, and QA visualization.

The Result

A Retrieval-Augmented Generation (RAG) powered chatbot and dashboard system that enables trustworthy, evidence-based responses— enhancing parental trust and platform credibility.

1. About TrustMyApps

TrustMyApps evaluates digital apps for children across parameters like **safety, educational value, cost-effectiveness, and language friendliness**. Its mission is to ensure every parent has access to transparent, data-backed app assessments before making decisions for their kids.

With increasing reliance on AI and LLMs, TrustMyApps needed to modernize its data pipeline for **accuracy, explainability, and scalability**—a goal achieved through Calibo's expertise.

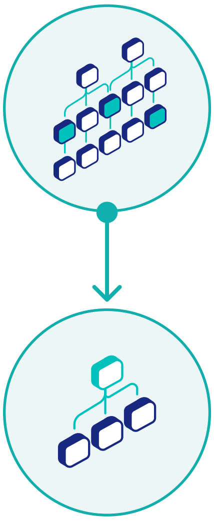
2. The Challenge

TrustMyApps had developed a promising ranking system, but the underlying data presented challenges:

1. Raw app assessment data was highly structured**, deeply nested, and complex**.
2. Existing summaries and groupings lacked automated verification or consistency checks.
3. **Duplicate or conflicting information** reduced overall reliability.
4. Without verified, queryable data, parents risked **inconsistent or incomplete insights**.

3. The Result

Through systematic parsing, normalization, and validation, the dataset was transformed into a clean, dependable format suitable for both human review and automated queries.



Before

Raw, nested JSON with duplicates and no validation, therefore hard to query, inconsistent, and unreliable.

After

Parsed, normalized, and validated JSON. Clean Markdown summaries for readable, queryable data.

The snippet below shows one extracted assessment block (**age_analysis_assessment**) from the model's structured evaluation of an educational app. It covers a single analytic dimension — **minimum_independent_age** —indicating how independently children can use the app. The data is shortened, with only a few evidence quotes shown. In the full dataset, each app includes multiple sections evaluating other aspects such as safety, pedagogy, and usability.

1. **Serialized Form** - the raw single-cell DynamoDB JSON blob from the CSV export, using DynamoDB's native type markers.

```
1. "age_analysis_assessment","{""dimensions"":{""M"":
   {""minimum_independent_age"":{""M"":{""summary"":
   {""S"":""Reviews suggest that children around 6-7 years old
   can use the app more independently...""},""rating"":
   {""N"":""6.0""},""confidence"":{""N"":""0.8""}}}}}}"
```

2. **Deserialized JSON** - the same data parsed into deserialized structure.


```
{
  "age_analysis_assessment": {
    "dimensions": {
      "minimum_independent_age": {
        "summary": "Reviews suggest that children around 6-7
        years old can use the app more independently...",
        "rating": 6.0,
        "confidence": 0.8,
        "evidence": [
          "My 8 year old daughter loved it so much...",
          "My 7 year old lovesss ittt!!!...",
          "My son uses this app independently and loves it..."
        ]
      }
    }
  }
}
```


3. **Markdown Section** - the cleaned, human-readable summary generated for indexing or documentation.

```
# Splash Math: K-5 Learning
**Developer:** StudyPad, Inc.
**Category:** Education

### Age Analysis
- **Minimum independent age (rating 6.0, confidence 0.8):**
Reviews suggest that children around 6-7 years old can use the
app without adult support.
- **Evidence highlights:**
“My 8 year old daughter loved it...”, “My 7 year old lovesss
ittt!!!...”, “My son uses this app independently...”
```

4. The Solution - Calibo's Approach

4.1 LLM-Optimized Data Transformation

Calibo's first objective was to transform app assessment data into a format optimized for AI models.

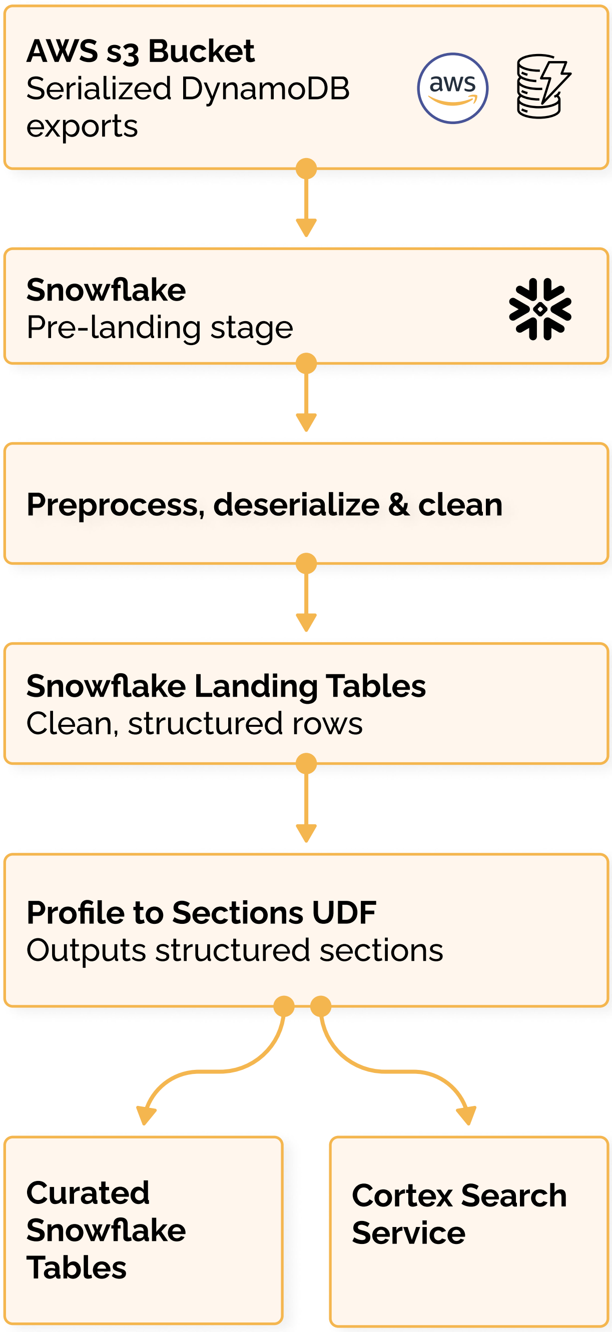
Key Actions

1. Load the raw CSV file stored in S3 into a **Snowflake pre-landing table**.

2. Run a **Python script to** deserialize DynamoDB data types, **clean invalid values**, infer the appropriate Snowflake schema, and write the cleaned output to a new table.

3. Apply a Snowflake UDF to process each profile row, **extract nested fields, and transform them into Markdown text** sections (e.g., header, overall_summary, safety_assessment, etc.) for more granular search.

4. Deploy a Cortex search service to **enable semantic search and RAG capabilities**, supported by a final Snowflake table for data exploration.

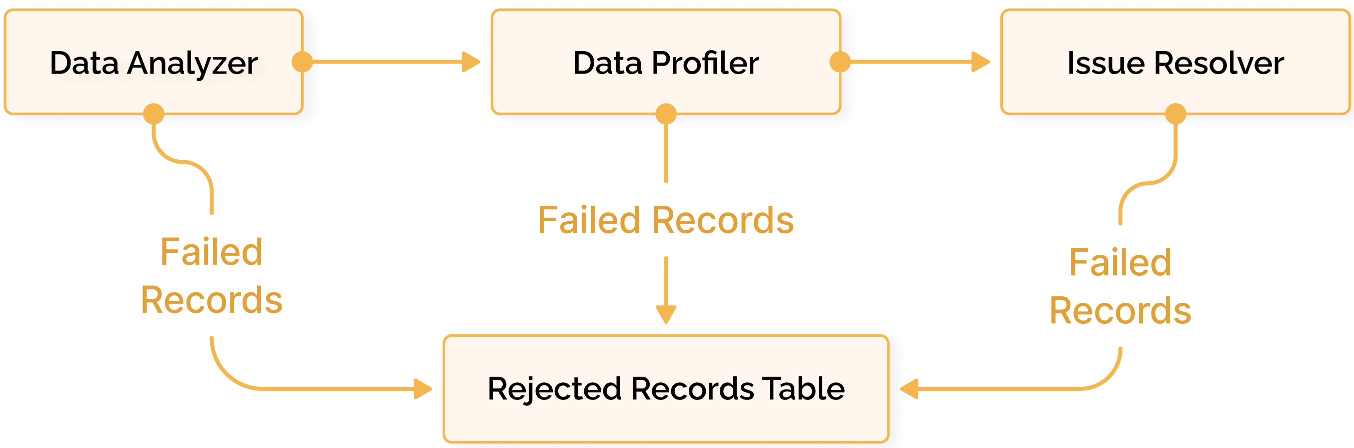


4.2 Data Quality & Verification

Calibo built automated verification mechanisms to ensure data trustworthiness.

Key Actions

1. Implemented automated **data profiling** and integrity checks.
2. Built **duplicate and contradiction detection scripts**.
3. Created a **QA dashboard** for real-time issue visualization.
4. Stored QA findings in LLM-ready format for continuous learning and **feedback loops**.



4.3 LLM Service to Query Research Papers

A Snowflake database knowledge base was developed to store and organize the client's research papers. The knowledge base is accessible through a REST API built with Flask, enabling client applications to query it programmatically.

5. Implementation Journey

5.1 Execution Phases

PHASE-1

Data Ingestion
Raw assessments loaded into Snowflake.

PHASE-2

Transformation
Markdown summaries and hierarchical chunking implemented.

PHASE-3

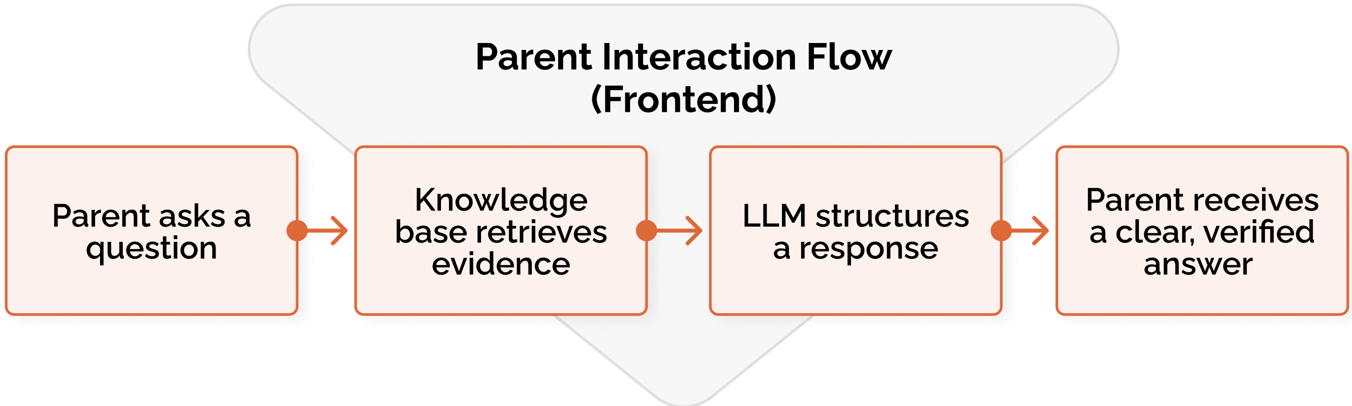
Verification
QA checks, issue detection, and dashboard visualization.

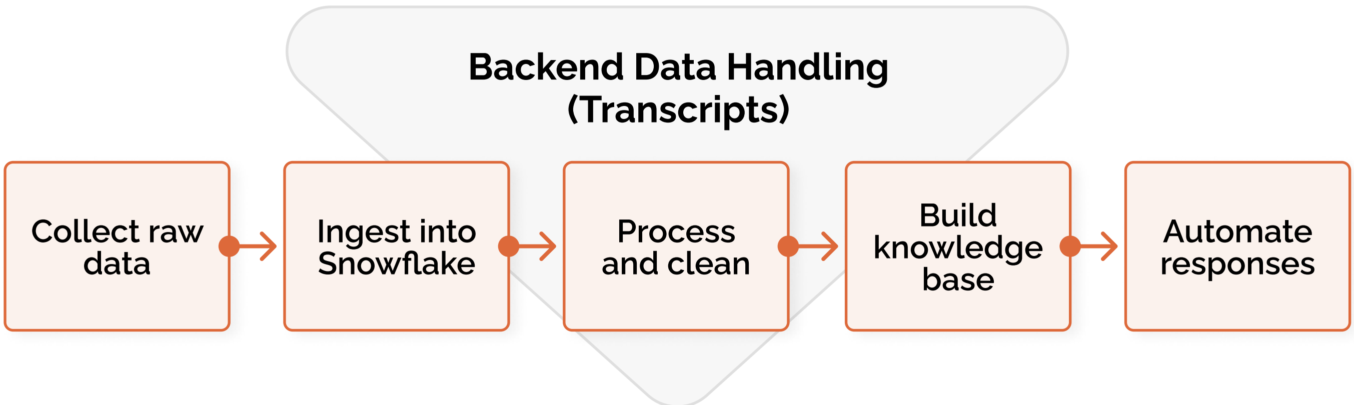
PHASE-4

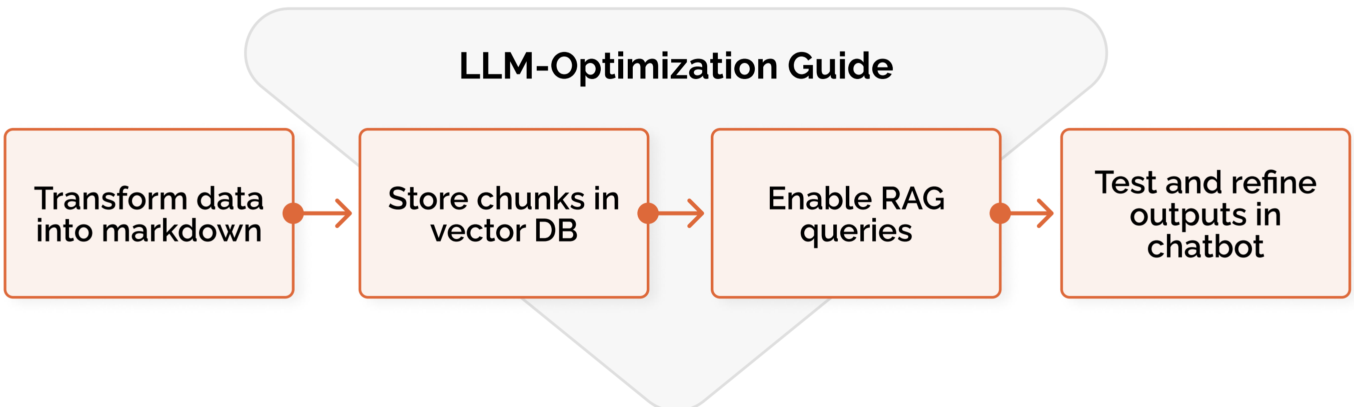
Testing
A Streamlit-based RAG chatbot was developed for internal validation and demo purposes, while the client used their own UI to access the backend services.

5.2 End-To-End Flows

This section outlines the complete system flow—from user interaction on the frontend to backend data processing, model optimization, and retrieval-augmented generation.

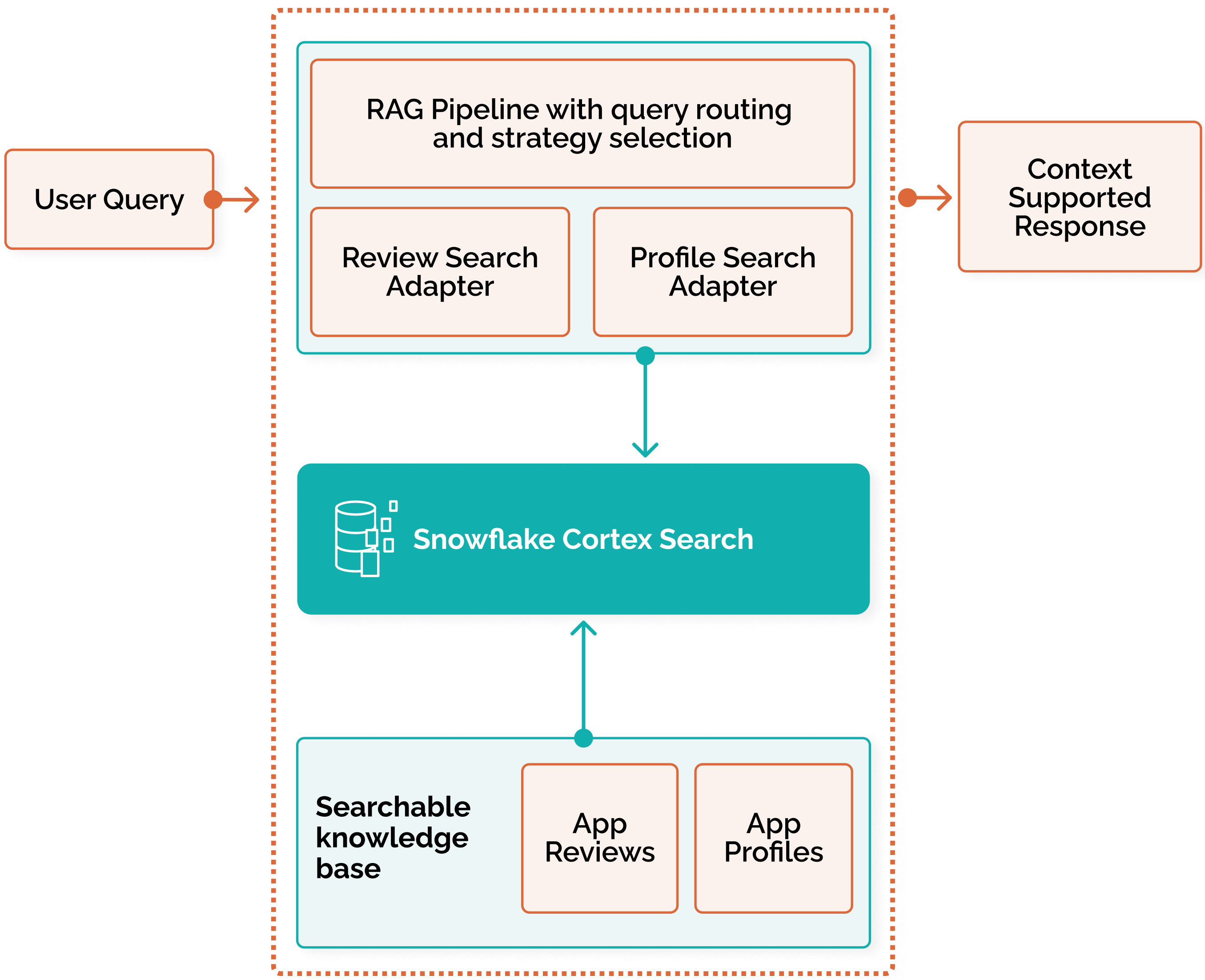






> Calibo supported us in identifying inconsistencies in our database, which we were then able to correct — improving data quality and reliability.
>
> **Business Lead - TrustMyApps**

RAG Pipeline Overview



5.3 REST API Pipeline

- **Snowflake Knowledge Base**
  Created a Snowflake knowledge base to store and organize client research papers.

- **Querying**
  Exposed the knowledge base through a Flask JSON API for querying.

- **Client Access**
  Enabled client access via HTTP requests or REST API clients.

6. Technical Highlights and Key Outcomes

6.1 Technical Highlights

Hierarchical Chunking

Designed a hierarchical chunking strategy to improve LLM query precision.

Automated QA

Automated QA with profiling, verification, and issue tracking.

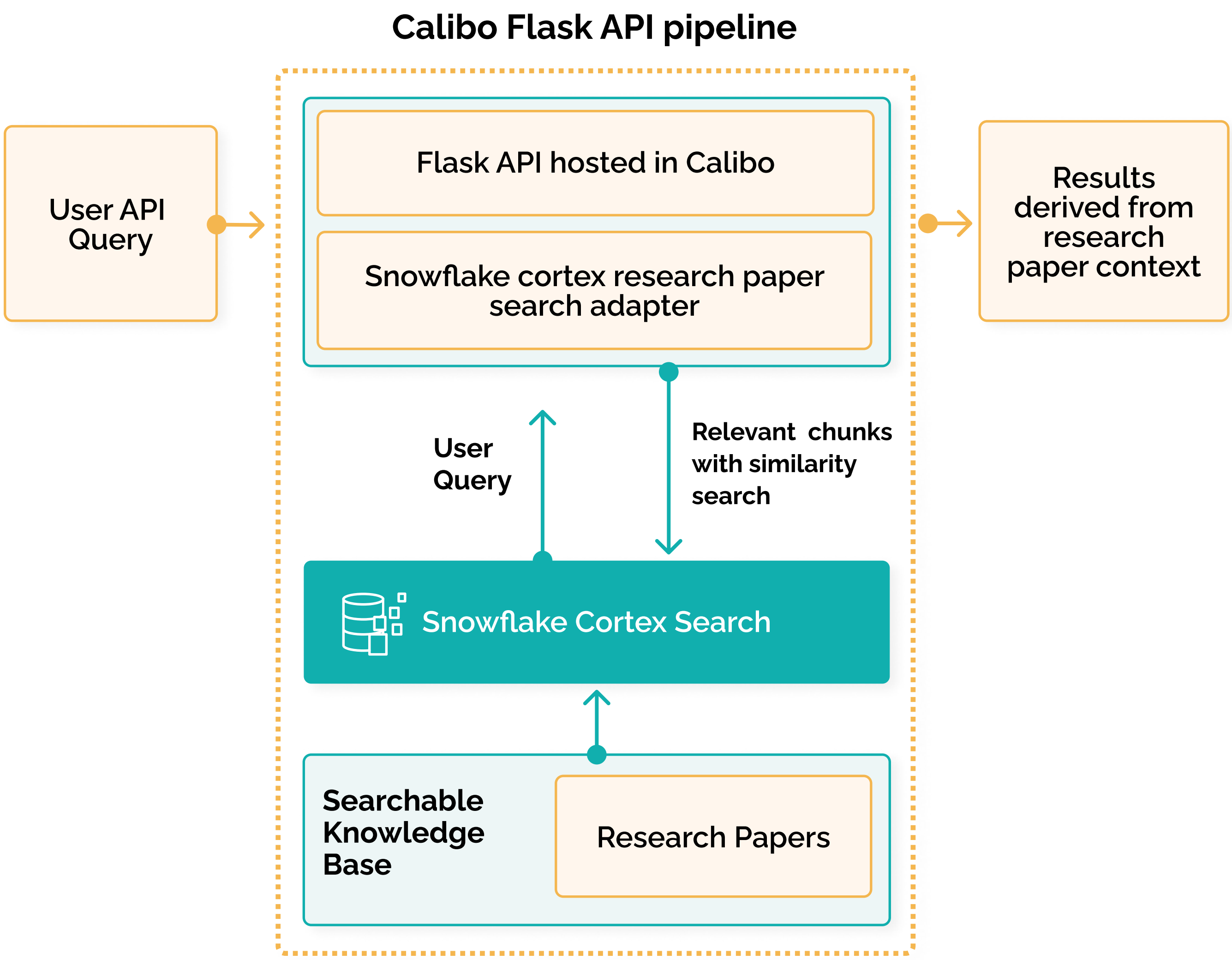
RAG System

Developed a RAG system using Snowflake Cortex with vector search optimization and built a Streamlit UI.

Developed a Flask-based JSON API pipeline to enable external data access.

Created a Snowflake knowledge base to store and organize verified research papers and assessment data.

1. Exposed this knowledge base via a Flask JSON API endpoint.
2. Enabled secure client access through REST API requests for seamless querying and integration.



6.2 Key Outcomes

Aspect	Before	After (Calibo Solution)
Data Structure	Raw JSON, no grouping	Structured markdown summaries by severity
Verification	Manual / limited	Automated QA dashboard & checks
Query Capability	Keyword search only	LLM-optimized retrieval (RAG) + JSON API for client access
Reliability	Inconsistent	Verified, evidence-based results

7. Results and Impact

Enhanced App Ranking Reliability
Verified, explainable results improved user trust.

Operational Efficiency
Streamlined data flow reduced manual QA time.

Impact Summary

1. Improved verification and insights into data inconsistencies.

2. Built duplicate and contradiction detection scripts. Higher precision and relevance in data retrieval for RAG queries.

3. More **consistent and structured data** for LLM-based analysis.

8. Why Calibo?

Calibo combines expertise in AI, LLM pipelines, and data governance to deliver business value with technical rigor.

Proven ability to move fast — delivered PoC in a few weeks.

End-to-end capability from **data ingestion to LLM deployment.**

Demonstrated excellence using **Snowflake Cortex, Streamlit, and vector search** technologies.

> "The Calibo team built a seamless end-to-end data pipeline — from DynamoDB exports to Snowflake and JSON APIs — making our entire LLM workflow faster, verifiable, and reusable."

Technical Lead - TrustMyApps

Calibo

Calibo helps enterprises accelerate use case delivery through a self-service platform that unifies software and data engineering. From cloud modernization to AI-powered apps, Calibo reduces development time by 50% and drives measurable outcomes.

Discover how enterprises accelerate time to value with Calibo's unified platform.

www.calibo.com