**Runwei Azure Services - Internal Technical Documentation**

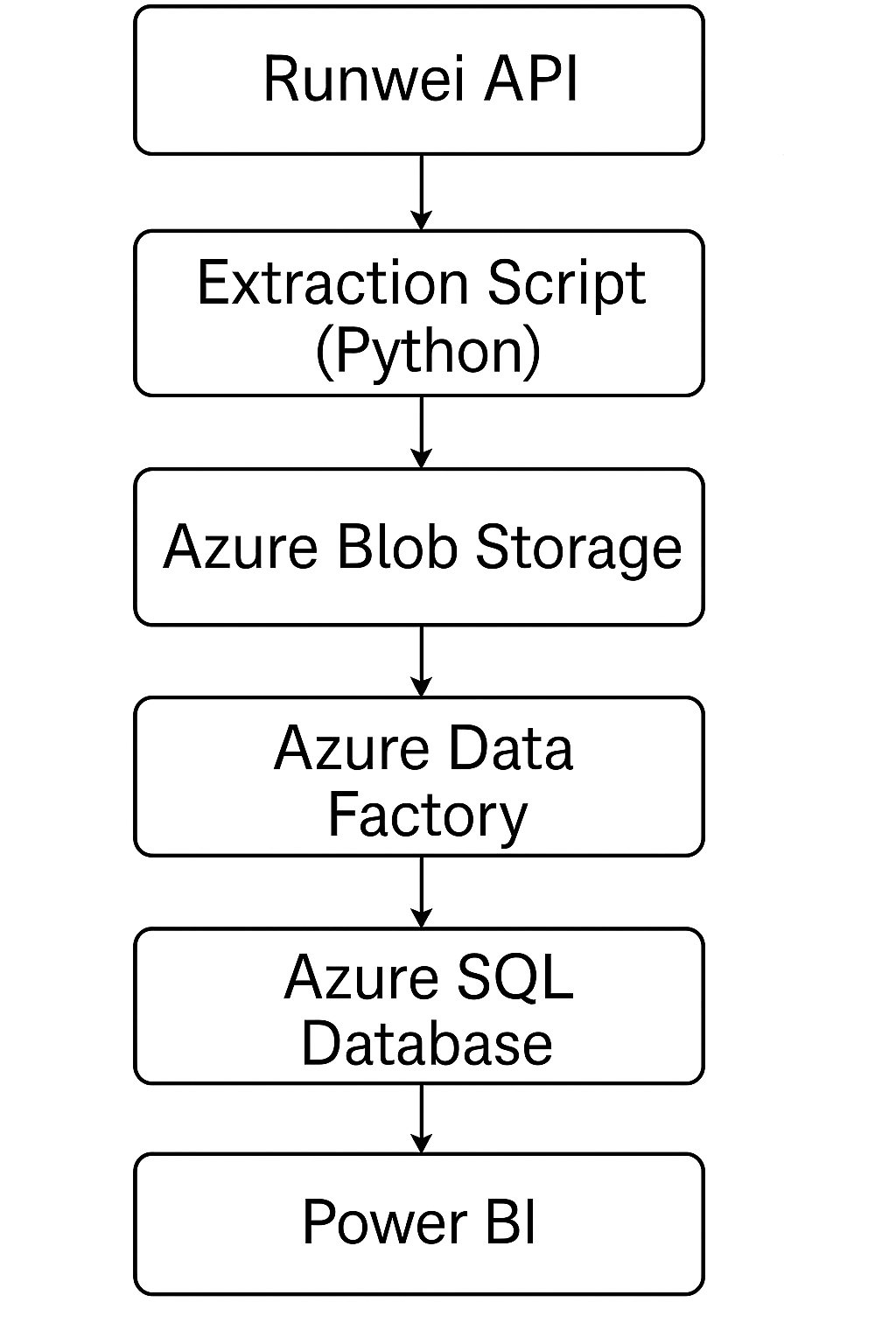
**1. Project Overview**

This document outlines the Azure-based architecture and services deployed at Runwei to support the ETL automation, data storage, analytics, and RAG (Retrieval-Augmented Generation) system for the Opportunity Data Intelligence Platform. The solution is designed for scalability, fault tolerance, and ease of maintenance.

**2. Azure Services in Use**

|  |  |  |
| --- | --- | --- |
| **Service** | **Role** | **Notes** |
| Azure Data Factory | ETL orchestration | Pipelines for extraction, transformation, and load; scheduled daily. |
| Azure Blob Storage | Raw data storage | JSON files from API saved with timestamps; intermediate storage optional for RAG inputs. |
| Azure SQL Database | Data warehouse | Stores transformed opportunity records; primary storage for analytics and AI systems. |
| Power BI | Visualization | Connects to Azure SQL; dashboards updated daily. |
| FAISS (On VM or local server) | Vector database | Local FAISS flat index built from SQL exports. |
| DeepSeek API | Language generation | Used for RAG response generation; API key secured via Azure Key Vault (future plan). |

**3. ETL Pipeline Details**

****

**3.1 Extraction**

* Source: Runwei API.
* Extraction script: Python (extract\_and\_upload.py).
* Storage: Files named runwei\_opportunities\_YYYYMMDD.json saved to Azure Blob Storage.
* Trigger: Task Scheduler on VM; plan to move to Azure Functions.

**3.2 Transformation**

* Tool: Azure Data Factory (Mapping Data Flow).
* Tasks:
  + Blank strings ("", N/A, n/a) converted to NULL.
  + Date formats standardized (YYYY-MM-DD).
  + Schema enforced for 40+ fields.

**3.3 Loading**

* Destination: Azure SQL Database (table: [dbo].[Opportunities]).
* Constraints:
  + Primary Key: ID
  + Insert/Update strategy considered for future (currently truncate-load).

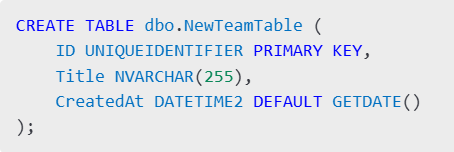
**3.4 Editing and Creating SQL Tables**

To edit and create tables in Azure SQL Database using the web interface:

* **Login to Azure Portal** and navigate to the RunweiOpportunities SQL database.
* In the left menu, click **Query editor**.
* Authenticate using either:
  + SQL Server Authentication (sql-admin + password),

\* Login: sql-admin ; Password: Runwei2025 (Need add addressIP first time)

* After logging in, you'll see the database structure (tables, views, etc.).
* To create a new table:
  + Click New Query.
  + Enter SQL like below:



1. Click **Run** to execute.
2. Refresh the Tables list to confirm creation.

⚠️ Note: Use dbo. prefix to follow convention.

**4. Visualization Layer (Power BI)**

* Connected to Azure SQL Database.(\* Contact me to acquire permission)
* Main databases:
  + [Opportunities]
  + [Opportunities\_cleaned]
* Data refreshed daily automatically.

**5. RAG (Retrieval-Augmented Generation) Integration**

**4.1 Data Preparation**

* Export fields: Title, ShortDescription, Eligibility, Tags, Industry, Description from Azure SQL.
* Text concatenated and embedded using sentence-transformer models (all-MiniLM-L6-v2).

**5.2 Vector Store**

* FAISS Index:
  + Flat index (IndexFlatIP) for maximum retrieval speed.
  + Stored locally; plan to deploy in VM blob storage if scaling is needed.

**5.3 Query + LLM**

* Top-K (e.g., K=5) nearest neighbors retrieved based on user query embedding.
* Context documents passed to DeepSeek API with a structured prompt.
* DeepSeek returns a summarized or direct answer.

**6. Automation and Monitoring**

* ADF Pipeline Monitoring:
  + Email alerts for failures (to be integrated with Azure Monitor).
* Blob Storage Cleanup:
  + Older JSON files archived or deleted periodically.
* Manual FAISS Refresh:
  + Currently triggered after daily pipeline; automate with cron job or Azure Function.

**7. Best Practices and Future Recommendations**

* Move Task Scheduler to Azure Function App for scalable extraction.
* Secure API keys and sensitive parameters using Azure Key Vault.
* Implement delta loading instead of full loads if API supports change tracking.
* Move FAISS deployment onto Azure VM with persistent volume support.
* Explore Azure OpenAI Service as alternative or supplement to DeepSeek for improved cost/performance.

**8. Repository and Access**

* Extraction Script: /runwei-etl/scripts/extract\_and\_upload.py
* ADF Pipelines: Azure Data Factory instance Runwei-ADF
* SQL Server: runwei-sql.database.windows.net - Database: OpportunitiesDB
* FAISS RAG Service: (internal use only, repo pending migration)

\*Access requires appropriate Azure AD permissions.