



AZURE DAY

AI Assistants

From Click-Deploy to Production

Morgana Lalli

Gianni Rosa Gallina



Deltatre
Innovation
Lab

deltatre



Microsoft



TECHNOLOGY





Platinum Sponsor



Technical Sponsor





AI Assistants

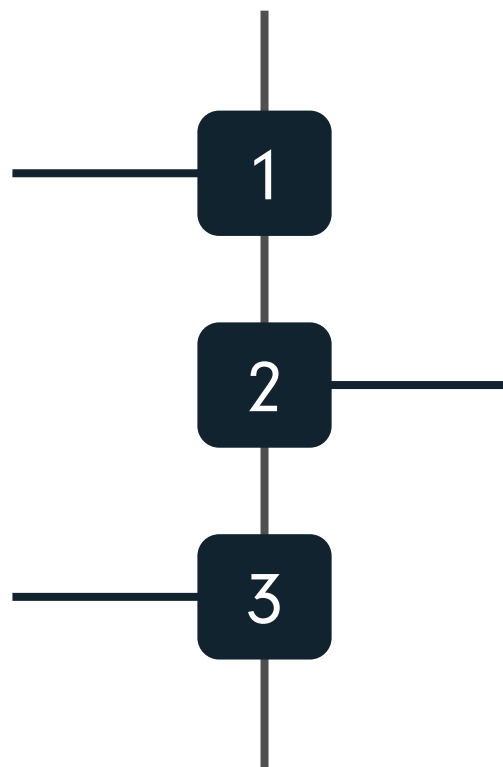
What are AI Assistants?

AI-driven tools that engage with users via natural language to provide intelligent, adaptive support

How Assistants Work

Retrieval Augmented Generation (RAG)

A combination of an information system retrieval with LLM skills to create contextually grounded responses



Why AI Assistants?

- Domain Expertise
- 24/7 Available
- Consistent and Adaptable
- Boost efficiency
- Scalable





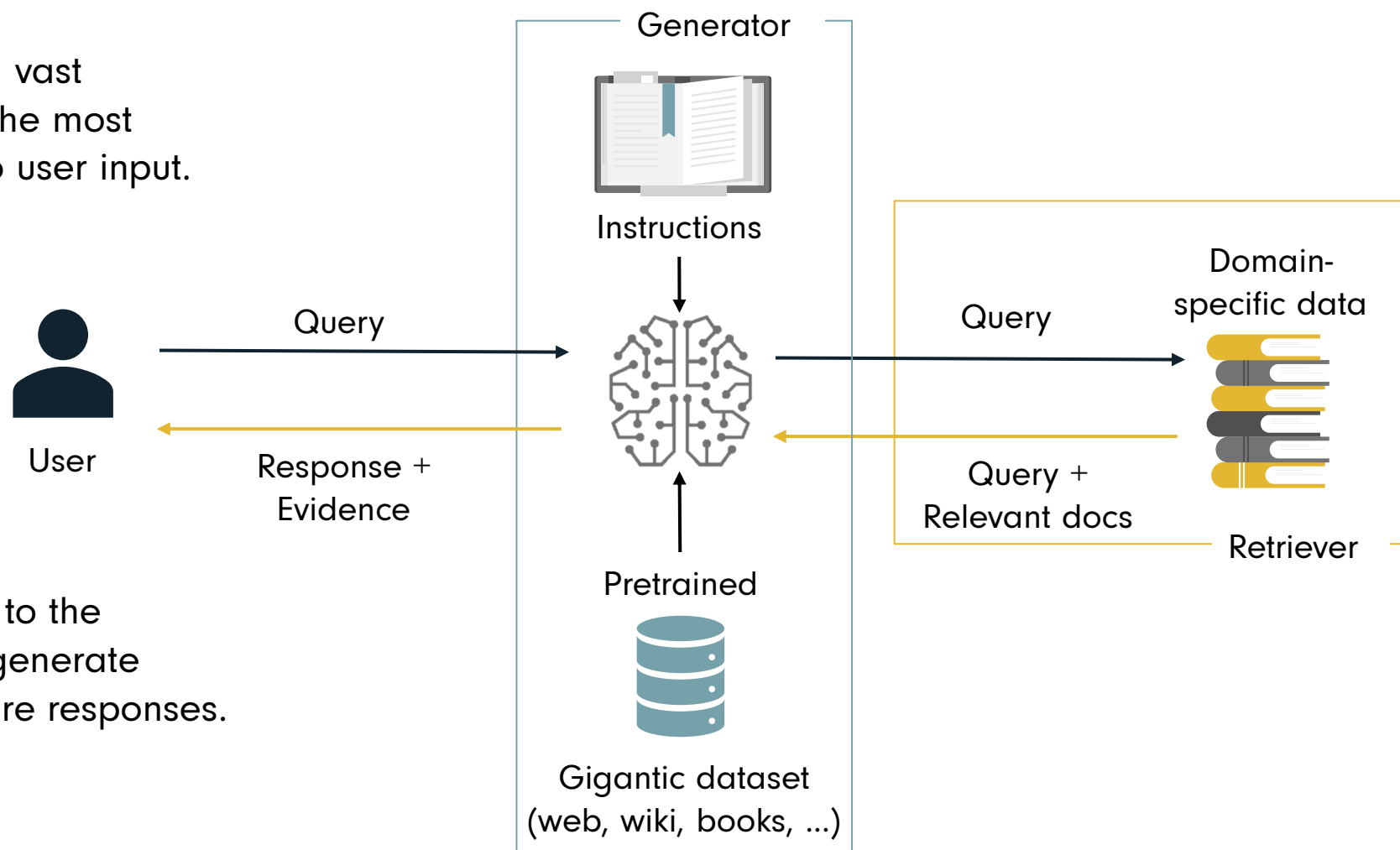
How does RAG work?

1. Retrieval

The solution queries a vast database to retrieve the most relevant documents to user input.

2. Augmentation

This content is served to the generative model to generate accurate, context-aware responses.





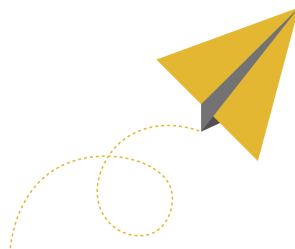
Why RAG over Fine-Tuning?



Always Up-to-Date

Replies are always up-to-date

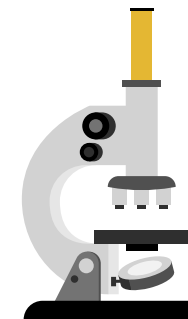
Adapts to changes in your data, making it flexible for evolving business needs.



Cost-Effective & Scalable

No need for retraining

Scales efficiently, adapting to user needs.



Trustworthy & Transparent

Verifiable response

Tracing answers back to the original content for deep-dive.



AZURE DAY

EXPLORATIONS





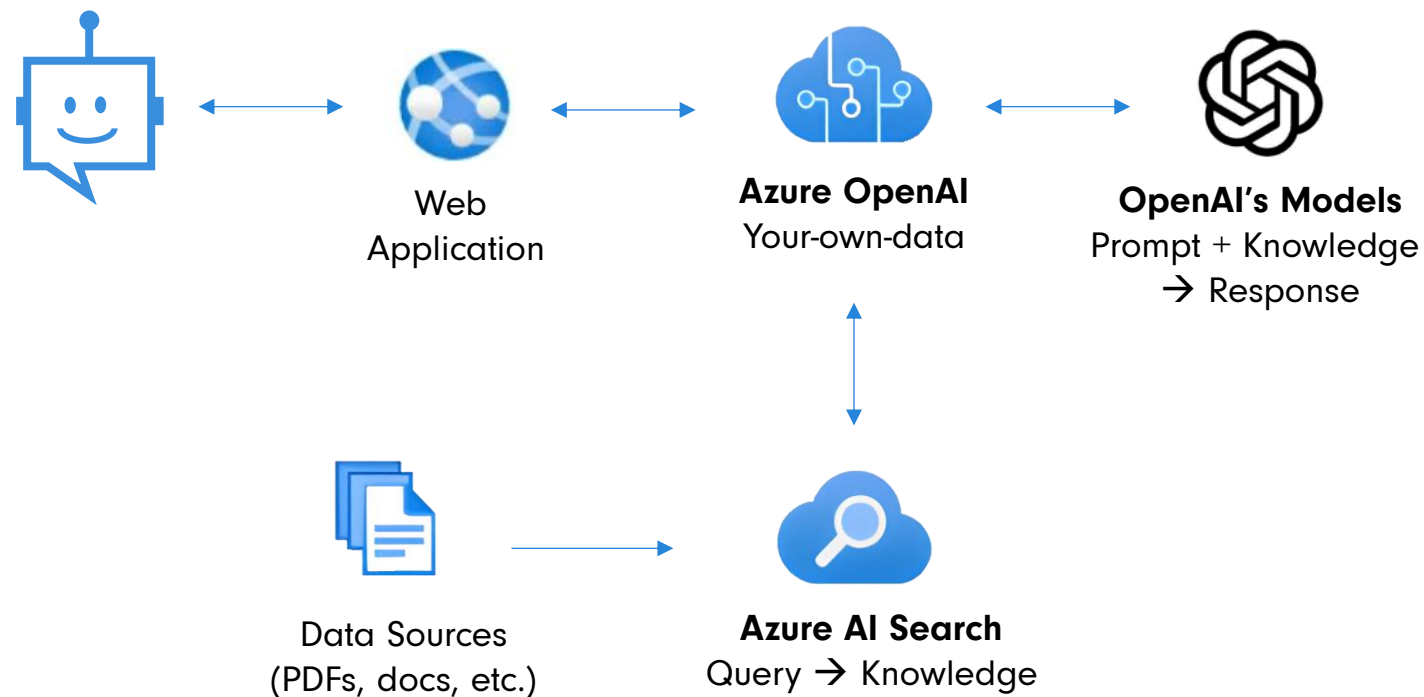
Exploration Phase

Azure OpenAI Services

Azure OpenAI Service integrates OpenAI's generative AI models with Azure's enterprise-grade cloud, offering **secure**, scalable access via REST APIs.

Chat with your data enables the Assistant to craft **system prompts** and integrate the LLM with your data.

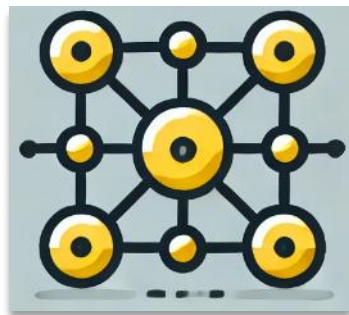
Azure AI Search is as the RAG **retrieval system**, indexing data with relevance tuning, security, and global reach.





Exploration Phase

Azure AI Search Retrieval Methods



Vector Search

Vector **embedded space**

Measures similarity between queries and documents

Identifies conceptually related content



Keyword Search

Match exact terms

Traditional search needs

Combines with Vector Search to form Hybrid Search



Semantic Search

Understand the **query intent** ranking results by contextual relevance and meaning.

Human-like understanding and deeper insights



Azure AI Search Core Components

Index



The Assistant's **knowledge base** storing processed content and associated metadata. It is **highly customizable**, allowing the configuration of metadata attributes like titles, paths, and vectors for retrieval, filtering, or sorting

Skillset



Defines **cognitive skills** for preparing and enriching data, including chunking, embedding, field mapping, and custom enrichments for AI processing.

Indexer



Handles **connecting, populating, and updating** the index with data from the source based on skillset instructions

Data Source

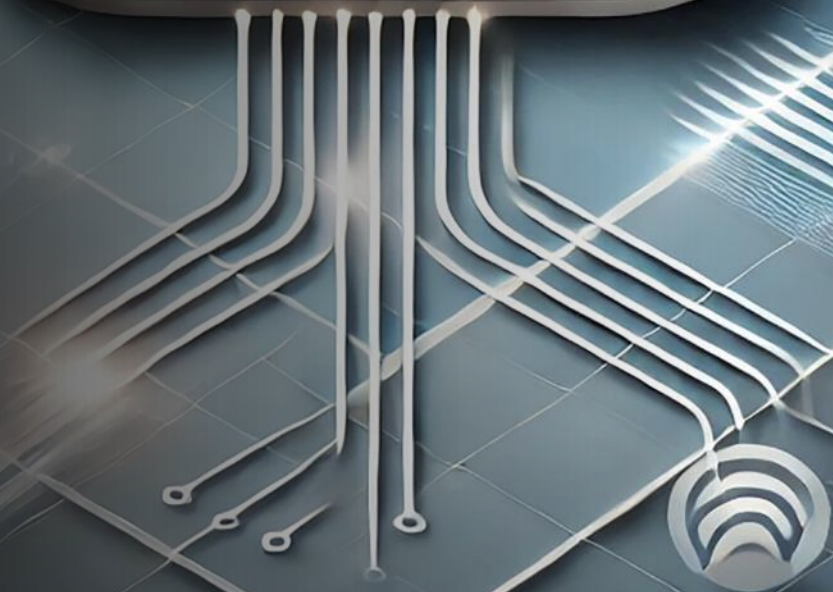


Connects to **original data sources**, storing raw data before enrichment and indexing



AZURE DAY

DEMO



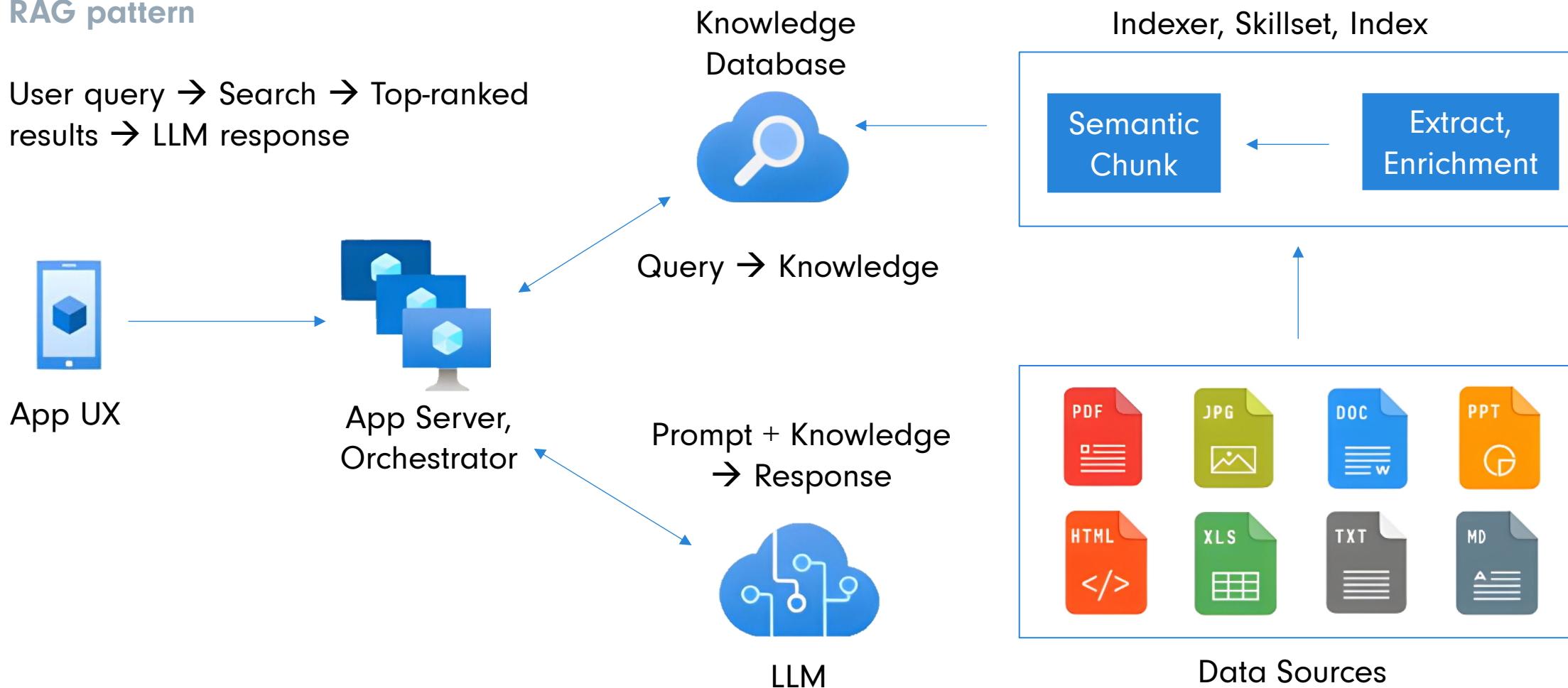


Exploration Phase

PoC Architecture

RAG pattern

User query → Search → Top-ranked results → LLM response





Identify Relevant Data Sources

An effective AI Assistant strictly depends on **high-quality data source identification**.



Scope



Collaboration



Data

Inputs

- The relevance of data sources depends on the **Assistant's scope**
- Early **collaboration with end users** accelerates behavior shaping and offers insights into key **information's *location***, format, and accessibility

Knowledge Sources

- **Internal Sources:** Confluence, GitHub (specs, guidelines), Teams threads, and training materials (video transcriptions, diagrams)
- **External Sources:** Public content (blogs, websites) and secure third-party partner documentation



Knowledge Mining

To ensure Azure AI Search works **effectively**, data must be optimized for indexing and retrieval.



Data Preparation

Python pipelines to collect, clean, and manipulate data



PDFs

Format for easy indexing



Diagrams

Use Mermaid syntax for better usability



LLM-Assisted Transformations

Convert informal content (chats, transcripts) into structured, informative documents

```
CHAT_SUPPORT_TO_DOCS = """
```

```
You are a sophisticated processing tool tasked with  
analysing and structuring technical support dialogues.
```

```
Your main objectives include:
```

- Identifying the main issue from the dialogue's title.
- Summarizing the core problem from the main text.
- Extracting and listing solutions and steps from the responses.

```
Operating guidelines:
```

- Output should be organized in a predefined format, focusing on clarity and relevance.
 - Strictly use the data provided in the structured input without adding assumptions.
 - Maintain professionalism and ensure privacy by omitting personal details.
- ```
"""
```





AZURE DAY

# TRANSITION TO PRODUCTION





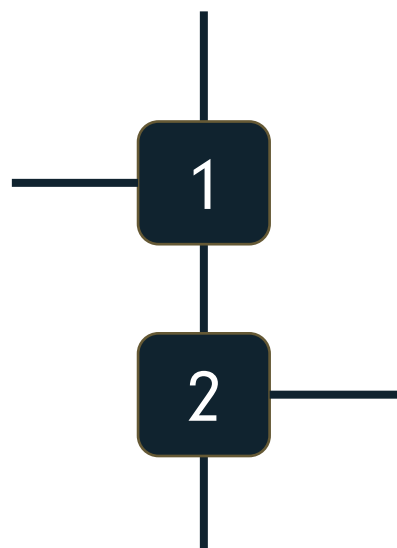


# Evaluation & Feedback

Monitoring is the **compass** for identifying improvement areas.

## Human Feedback

- In-Interface Thumbs Up/Down
- Dedicated surveys assessing the impact of the Assistant.



## Objective Evaluation

- Ground truth based
- Only exploiting LLMs

Guidelines and best practices to maximize the Assistant's effectiveness



# LLM-as-a-Judge



## Context Relevance

Checks if the response correctly answers the query and stays aligned with the **prior conversation**

*What are popular attractions in Paris?*

☺: The Eiffel Tower and Notre-Dame Cathedral.

☹: The Eiffel Tower is amazing. Do you like French food?



## Answer Relevance

Assess if the Assistant's response is **relevant and exhaustive**.

*What is the best time to visit Paris?*

☺: April to June or October for mild weather.

☹: Paris is a major European city.



## Groundedness

Measurement of **knowledge base usage**, and the answer **faithfulness**.

*How many people visit the Eiffel Tower each year?*

☺: About 7 million annually.

☹: Around 20 million people annually (incorrect data).





# Answer Relevance

```
ANSWER_RELEVANCE_TRULENS = """"You are a RELEVANCE grader; providing the relevance of the given RESPONSE to the given PROMPT. Respond only as a number from 0 to 10 where 0 is the least relevant and 10 is the most relevant.
```

A few additional scoring guidelines:

- Long RESPONSES should score equally well as short RESPONSES.
- Answers that purposely do not answer the question, such as 'I don't know' and model refusals, should also be counted as RELEVANT.
- RESPONSE must be relevant to the entire PROMPT to get a score of 10.
- RELEVANCE score should increase as the RESPONSE provides RELEVANT context to more parts of the PROMPT.
- RESPONSE that is RELEVANT to none of the PROMPT should get a score of 0.
- RESPONSE that is RELEVANT to some of the PROMPT should get a score of 2, 3, or 4.
- RESPONSE that is RELEVANT to most of the PROMPT should get a score between a 5, 6, 7 or 8.
- RESPONSE that is RELEVANT to the entire PROMPT should get a score of 9 or 10.
- RESPONSE that is RELEVANT and answers the entire PROMPT completely should get a score of 10.
- RESPONSE that confidently FALSE should get a score of 0.
- RESPONSE that is only seemingly RELEVANT should get a score of 0.
- Never elaborate.

Please answer with this template:

TEMPLATE:




Criteria: <Provide the criteria for this evaluation>

Supporting Evidence: <Provide your reasons for scoring based on the listed criteria step by step. Tie it back to the evaluation being completed.>

Score: <The score 0-10 based on the given criteria>"""



# Recognized Challenges

|  <b>Front End</b> |  <b>Back End</b> |  <b>Optimization</b> |
|----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Customization Demands                                                                              | Automating Data Flow                                                                                | Feedback Loop                                                                                           |
| Chat History                                                                                       | Security Enhancements                                                                               | Monitoring and Optimization                                                                             |
| Navigable Links and Schema                                                                         | Scalability                                                                                         | System Integration                                                                                      |



AZURE DAY

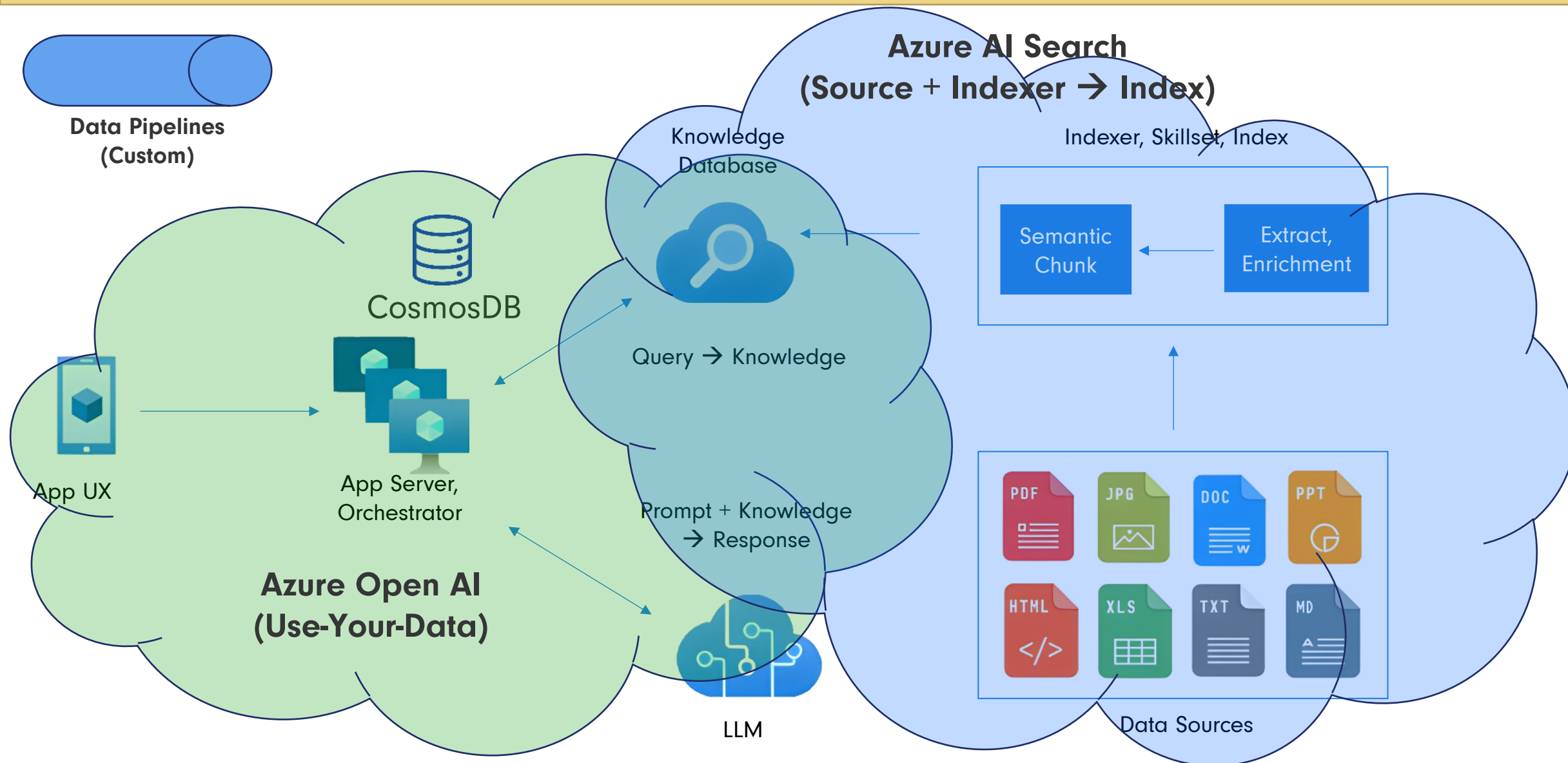






Road to Production

# PoC Architecture Recap

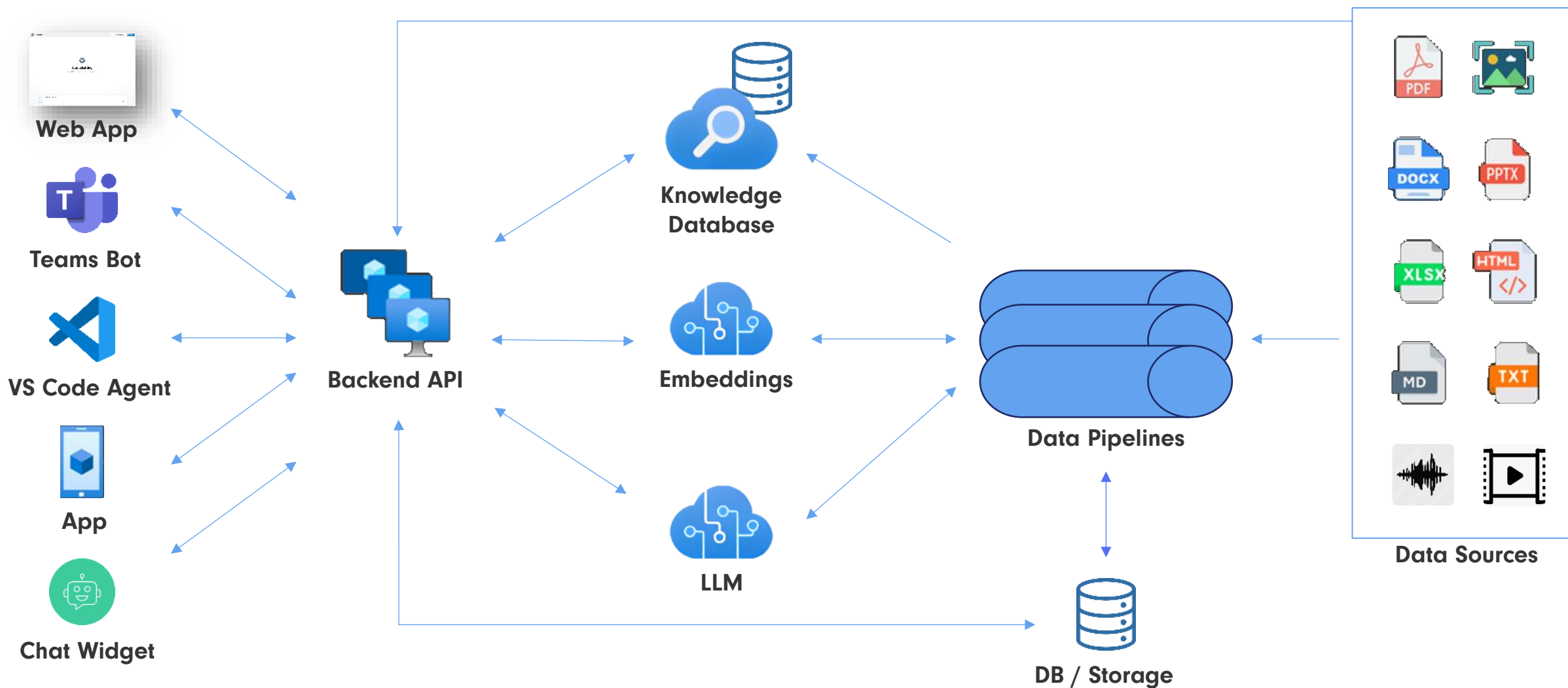






Road to Production

# High-Level Overview





Road to Production

# Customization Needs



**All-in-1  
Assistants**



**Cloud  
Agnostic**



**Extended  
References**



**Enhanced  
Feedback**



**Enterprise  
Guidelines**



**Styling &  
Localization**



**Monitoring &  
Optimization**

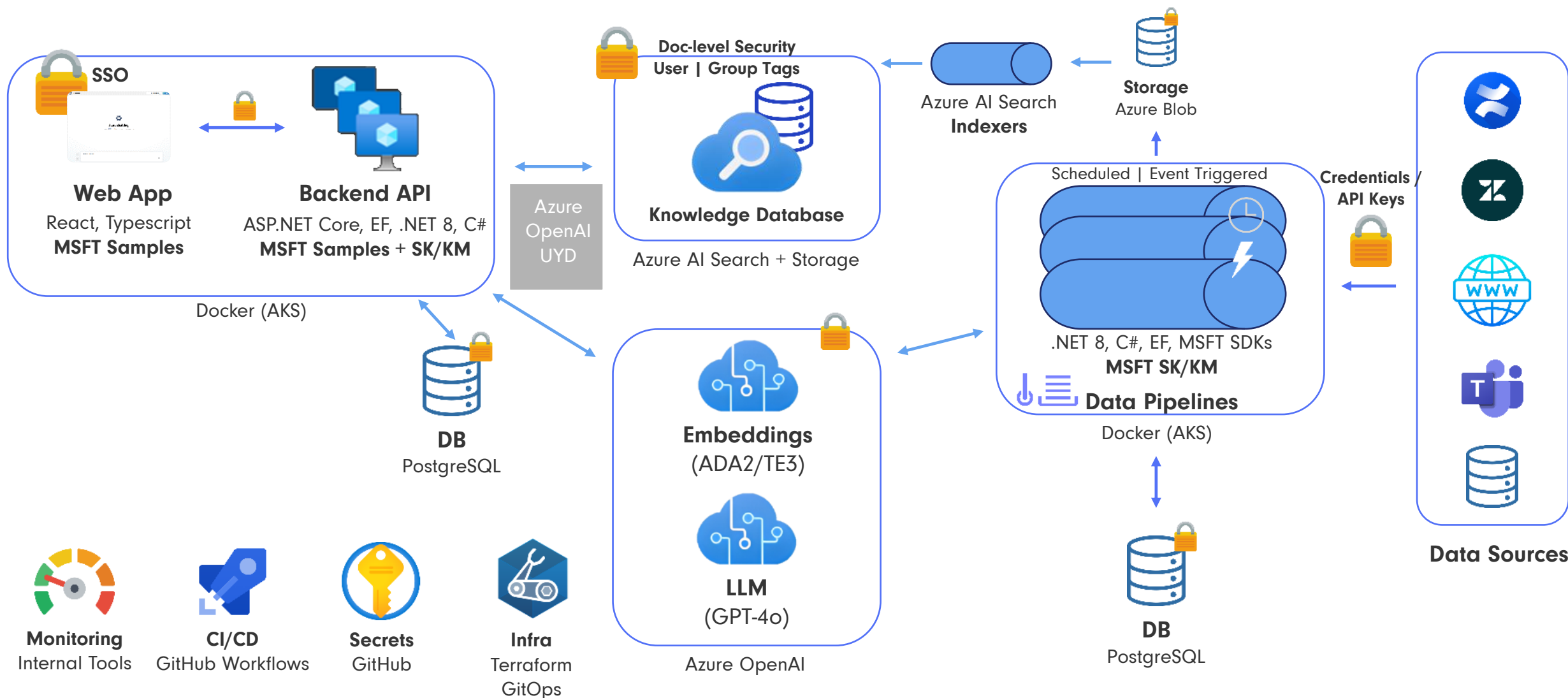


**Security**



Road to Production

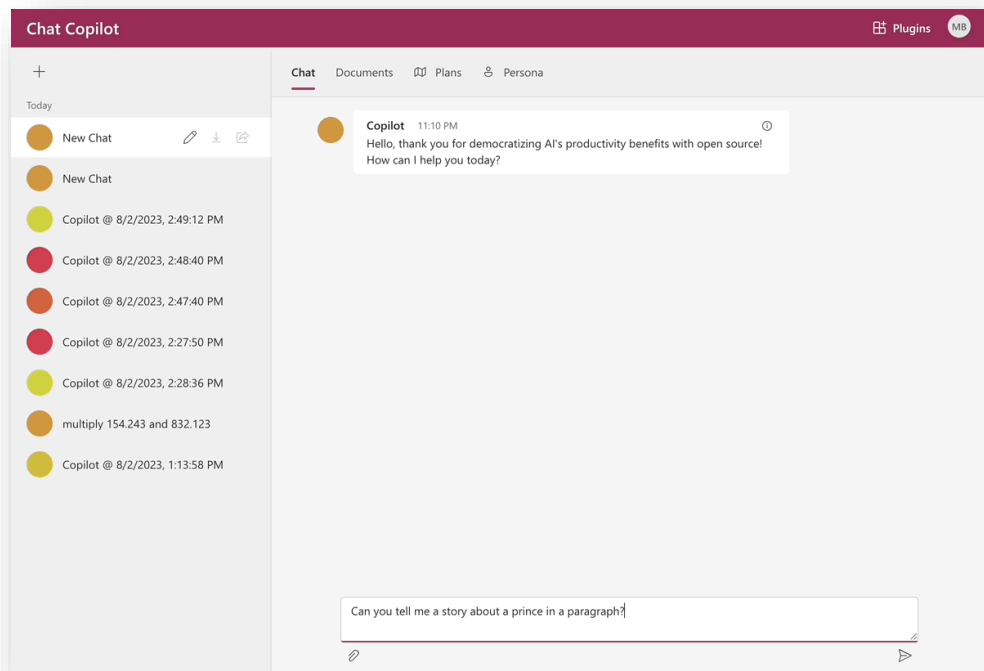
# Production Architecture





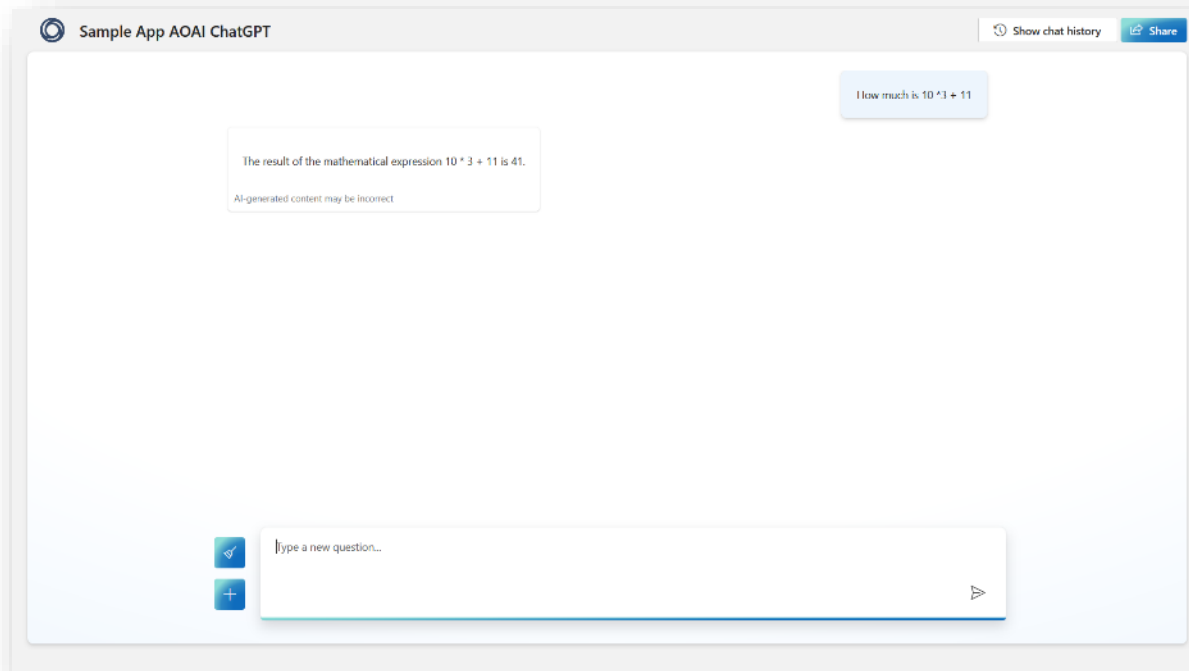
Road to Production

# Technologies & Frameworks



<https://github.com/microsoft/chat-copilot/>

React, Typescript  
ASP.NET Core, .NET 8, C#  
**Semantic Kernel + Kernel Memory**



<https://github.com/microsoft/sample-app-aoai-chatGPT>

React, Typescript  
Python  
**Azure OpenAI API (UYD)**





# LLMs, Chats, Tools



## Semantic Kernel

A lightweight, **open-source** development kit



## Enterprise-ready

Flexible, secure, modular, and observable



## Orchestration & Plans

Combines AI models, embeddings, prompts with APIs & tools to perform actions & business automation



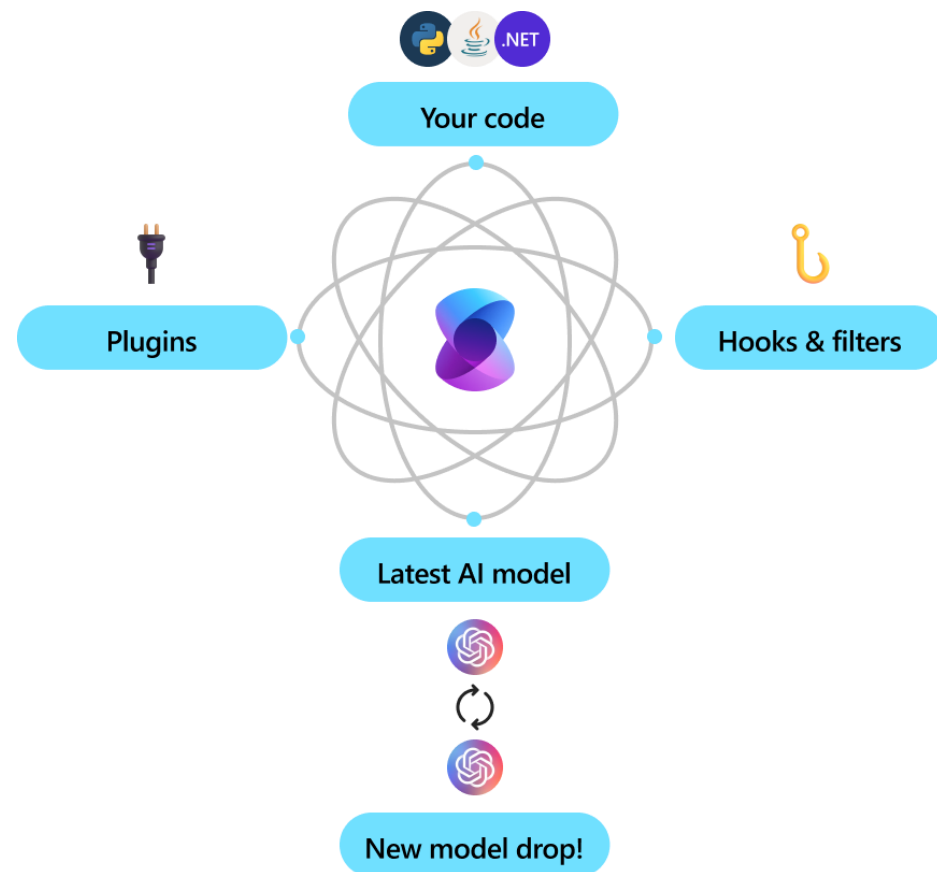
## Modular and Extensible

Many out-of-the-box connectors  
Support for integrating *existing* code as **plugin**



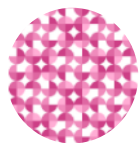
## Cross-Platform

.NET, Python, and Java





# Memory, Data, Workflows



## Kernel Memory

A multi-modal, open-source AI Service for efficient datasets indexing



## Data Pipelines & Workflow Orchestration

Supports Data Retrieval, Custom Pipelines, and Semantic Memory processing



## RAG Support

Advanced querying via embeddings and LLMs, over multiple knowledge bases



## Built-in Data Formats & Transformations + Custom

Supports various data types, including PDFs, documents, web pages, and images



## Out-of-the-Box Connectors + Custom

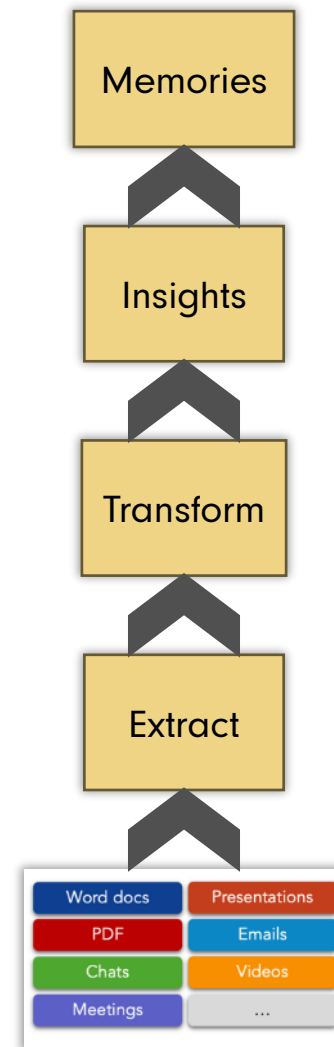
Support a wide range of data stores (Vector DBs, RDBMS, File System, etc.)  
Integrates with external tools like Azure OpenAI, ChatGPT and Microsoft Copilot



## Enterprise-proof

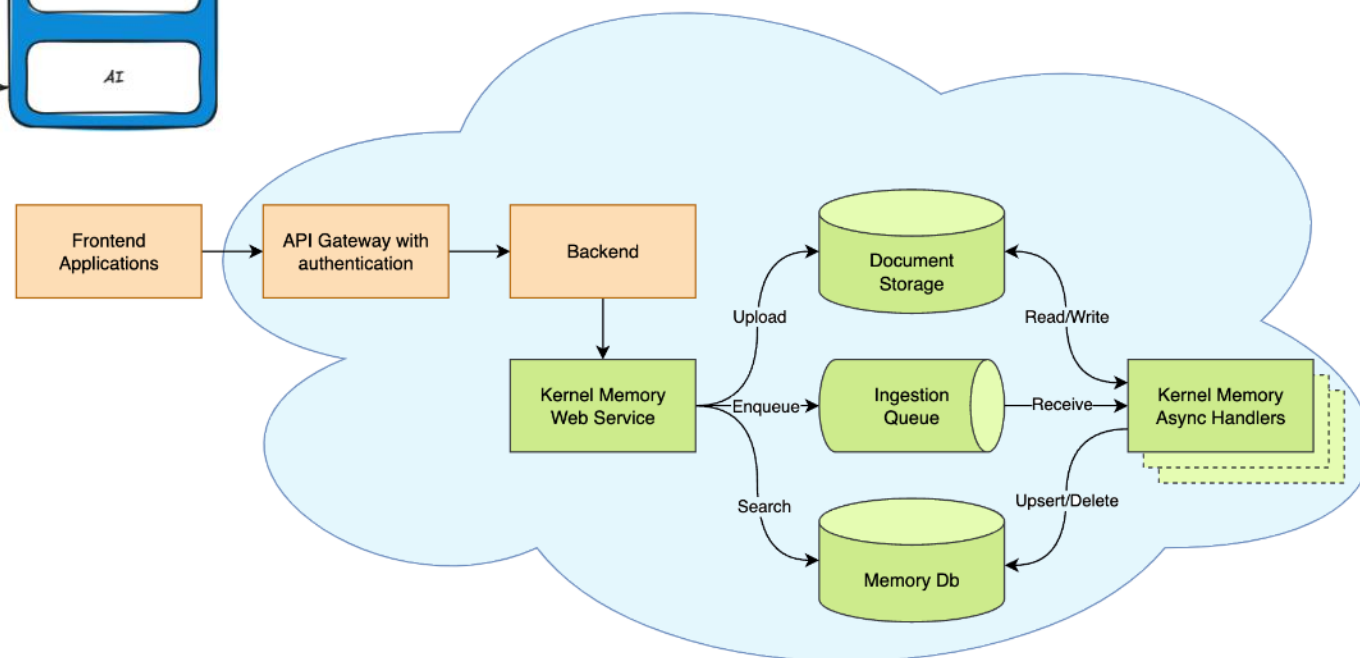
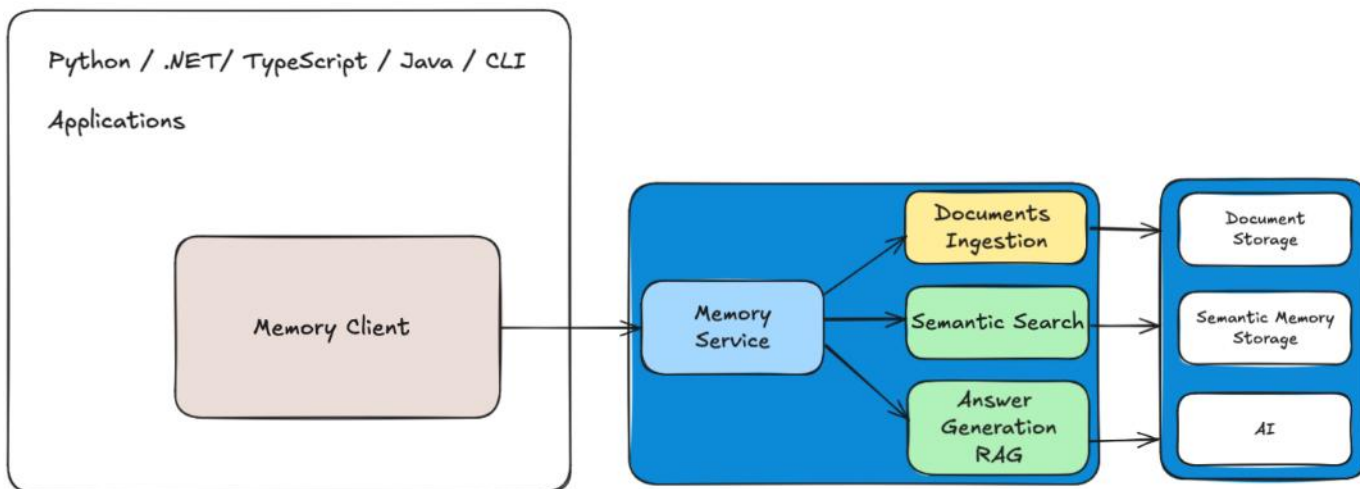
Flexible, modular, observable, and secure

<https://microsoft.github.io/kernel-memory/>





# Data Pipelines

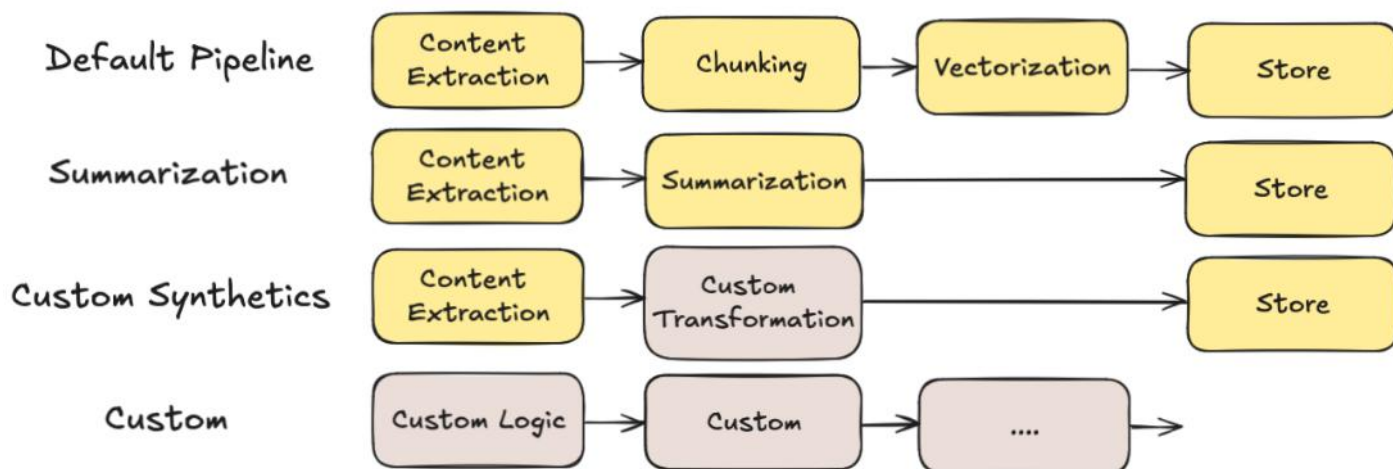


<https://github.com/microsoft/kernel-memory>

<https://github.com/microsoft/kernel-memory/tree/main/examples>



# Data Pipelines



```
memory.Orchestrator.AddHandler<TextExtractionHandler>("extract_text");
memory.Orchestrator.AddHandler<TextPartitioningHandler>("split_text_in_partitions");
memory.Orchestrator.AddHandler<GenerateEmbeddingsHandler>("generate_embeddings");
memory.Orchestrator.AddHandler<SummarizationHandler>("summarize");
memory.Orchestrator.AddHandler<SaveRecordsHandler>("save_memory_records");

/*****
 * Import files using custom handlers
 *****/

// Use the custom handlers with the memory object
await memory.ImportDocumentAsync(
 new Document("inProcessTest")
 .AddFile("file1-Wikipedia-Carbon.txt")
 .AddFile("file2-Wikipedia-Moon.txt")
 .AddFile("file3-lorem-ipsum.docx")
 .AddFile("file4-KM-Readme.pdf")
 .AddFile("file5-NASA-news.pdf")
 .AddTag("testName", "example3"),
 index: "user-id-1",
 steps:
 [
 "extract_text",
 "split_text_in_partitions",
 "generate_embeddings",
 "save_memory_records"
]
);
```

<https://github.com/microsoft/kernel-memory>

<https://github.com/microsoft/kernel-memory/tree/main/examples>

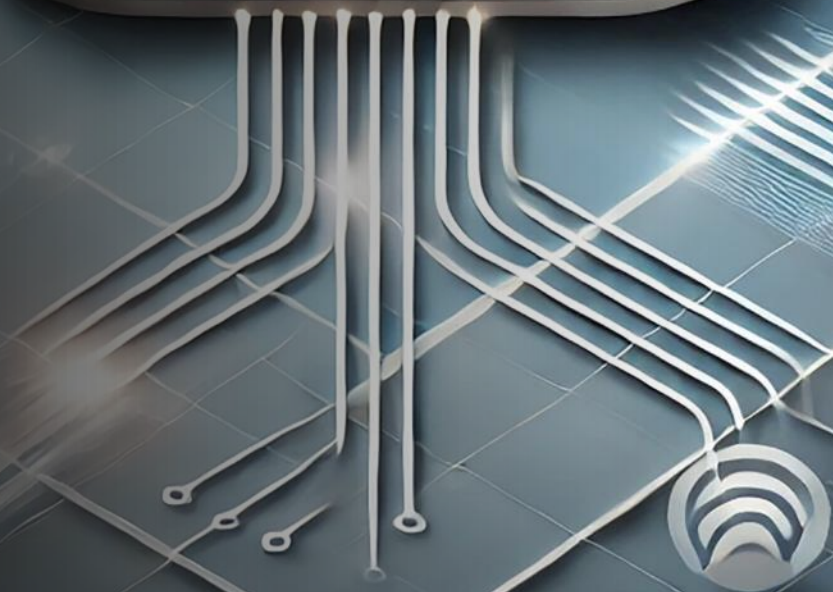




AZURE DAY

# DEMO

---





# Lessons Learned (part 1)



### Data is King

High-quality, well-prepared data is essential for success and contextually relevant responses.



### Cross-Functional Collaboration

Collaboration between AI teams, developers, and end-users is key to ensuring all needs are met.



### Master Retrieval Techniques

Efficient indexing, data enrichment, and retrieval methods ensure fast, relevant results.



### Iterative Approach

Continuously test, refine, and optimize. Constant feedback loops are vital for improving accuracy and performance over time.



# Lessons Learned (part 2)



### LLM-as-a-Judge

Use both human feedback and automated evaluation to measure the overall Assistant's impact.



### Continuous Improvement

RAG-based solutions evolve rapidly. Stay updated with new methods and technologies.



### Scalability and Flexibility

Ensure the Assistant can handle increasing data volume and user demand while remaining adaptable to new scenarios.



### Security & Access Management

Implement robust security measures and ensure compliance to protect sensitive information.



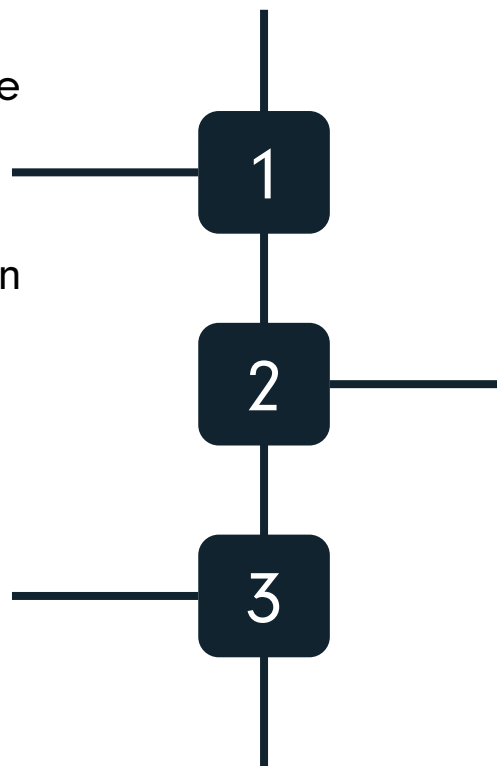
# What's Next

## New Retrieval Approaches

- Knowledge **graphs** (GraphRAG) to enhance semantic understanding
- **Advanced Retrieval** to reduce hallucinations and improve contextualization

## Architectural Enhancements & Solutions

- **Semantic Cache** to improve efficiency by reusing cached context.
- **New Frameworks:** Evolving architectural solutions in the RAG landscape



## Evaluation and CI Empowerment

- Synthetic Ground Truth to strengthening LLM-as-a-Judge evaluation
- **Advanced Monitoring Tools** in production like Azure Prompt Flow, Nebuly, and Langfuse







From Click-Deploy to Production

# AI Assistants



**Vote our session**





AI Assistants

Q&A





# Additional References

- <https://learn.microsoft.com/en-us/azure/ai-services/openai/overview>
- <https://learn.microsoft.com/en-us/azure/search/>
- <https://learn.microsoft.com/en-us/azure/ai-services/openai/use-your-data-quickstart>
- <https://www.trulens.org/>
- <https://github.com/microsoft/chat-copilot/>
- <https://github.com/microsoft/sample-app-aoai-chatGPT>
- <https://learn.microsoft.com/en-us/semantic-kernel/overview/>
- <https://github.com/microsoft/kernel-memory>
- <https://github.com/microsoft/kernel-memory/tree/main/examples>
- <https://github.com/Andrew-Jang/RAGHub>



AI Assistants

# About Us



Ing. Morgana **LALLI**

R&D Data Scientist @ **deltatre**

**Deltatre**  
Innovation  
Lab

- Biomedical Engineering, focused on sports data analysis.
- AI, Machine Learning, Deep Learning on multimedia content
- Skilled in exploratory data analysis techniques, handling diverse and complex data types.
- Currently working on RAG-based solutions to advance NLP capabilities
- LinkedIn: <https://www.linkedin.com/in/morgana-lalli-b5ab0a172/>





AI Assistants

# About Us



Ing. Gianni **ROSA GALLINA**

R&D Technical Lead @ **deltatre**

**Deltatre**  
Innovation  
Lab

✕ @giannirg



**Microsoft**  
Specialist

Programming in C#  
Programming in HTML5  
with JavaScript & CSS3

**Microsoft**  
CERTIFIED

Solutions Developer

Windows Store Apps Using C#  
Web Applications



- AI, Machine Learning, Deep Learning on multimedia content
- Virtual/Augmented/Mixed Reality
- Immersive video streaming & 3D graphics for sport events
- Cloud solutions, web backends, serverless, video workflows
- Mobile apps dev (Windows / Android / .NET MAUI / Avalonia)
- End-to-end solutions with Microsoft Azure
- Blog: <https://gianni.rosagallina.com/en/>
- LinkedIn: <https://www.linkedin.com/in/gianni-rosa-gallina-b206a821/>



**PLURALSIGHT**  
Author



AZURE DAY

## Platinum Sponsor



## Technical Sponsor

