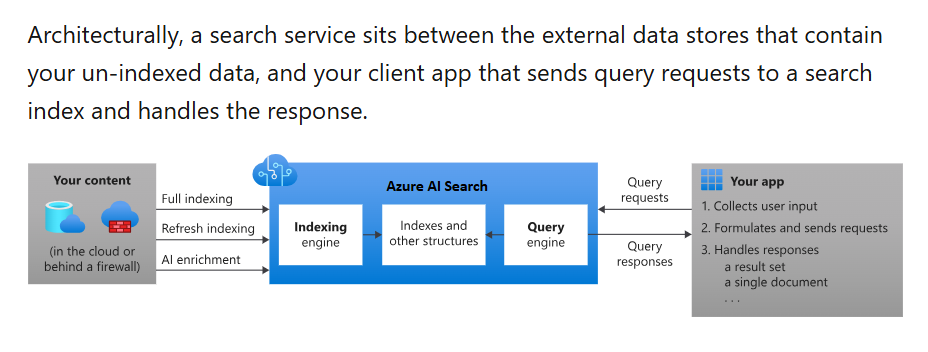
Overview of Azure Ai Search:

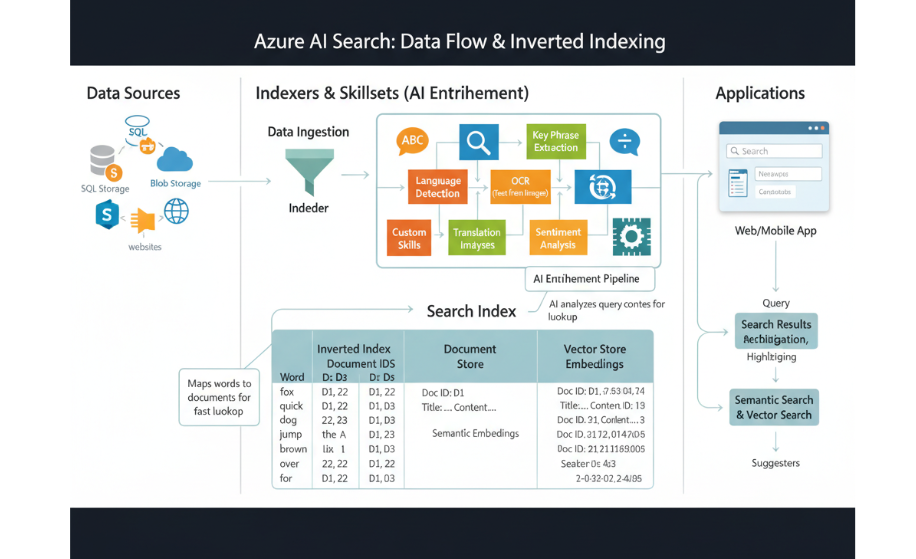


Inside search service:

Two primary workloads:

->Indexing

->Querying



Multi agent design with Azure AI search:

1. Role of agents:

* 1. Reception Agent
* 2.NLU agent
* 3.Data connector agent
* 4.Query execution agent
* 5.Summarizer agent

1. Workflow:

* 1st agent: Captures the query
* 2nd agent: detects->intent = document lookup
* 3rd agent : selects azure cognitive search index
* 4th agent: builds filter query in json format
* 5th agent: returns chatbot message

1. Architecture:

User ->Reception agent -> NLU agent -> Data connector agent -> choosing data source from any -> query execution agent -> response composer agent -> Chatbot UI

1. Models from Azure Foundry:

* Cohere Command A -> NLU(query to sql/filter) -> $17.125/gpu/hr
* Mistral large -> summarization
* Grok-4 Fast -> Reception (taking input from user/chatbot) -> Token based API pricing
* Cohere Embed 4 -> Embeddings for Azure ai search

Frameworks for Multi agent Orchestration:

1. Langchain:

* Provides Agents, tools and chains
* Easy to build multi agent pipelines
* Has direct integrationwith azure open ai and cognitive search

1. Microsoft AutoGen:

* Built for multi-agent conversation orchestration
* Agents communicate with each other in natural language

1. Semantic kernal:

* Plugin based orchestration for AI agents.
* Designed to connect with enterprise data
* Integration with Azure ecosystem.

1. CrewAI:

* Collaborative AI agents
* Each agent can have a role

1. Haystack:

* Focused on retrieval and question answering.
* Can be extended with multiple agents

Recommended for our persona:

* Langchain, Autogen and Semantic kernel

**Pre-trained Models Available in Azure AI Foundry**

1. **Cohere Command A:**
   * **Role in your project:** Used for **NLU (Natural Language Understanding)**, specifically for converting a query into a SQL-like query or a filter for your search index.
   * **Type:** A powerful large language model (LLM) developed by Cohere, designed for tasks like understanding intent, summarization, and generation.
   * **Availability:** Cohere models are available through Azure, usually within Azure Machine Learning's model catalog or potentially via specific Azure AI services that integrate third-party models.
2. **Mistral Large:**
   * **Role in your project:** Used for **Summarization**.
   * **Type:** A large language model from Mistral AI, known for its strong performance in various language tasks, including summarization.
   * **Availability:** Mistral models are becoming increasingly available on Azure, often through Azure Machine Learning's model catalog or Azure AI Studio.
3. **Grok-4 Fast:**
   * **Role in your project:** Used for **Reception (taking input from the user/chatbot)**.
   * **Type:** While "Grok" is famously associated with xAI, and "Grok-4" suggests a specific version, this naming convention within Azure's offerings might refer to a performant, potentially custom-tuned, or optimized LLM for conversational understanding and response generation. It's likely an LLM chosen for its speed and efficiency in processing user inputs.
   * **Availability:** This could be a model from xAI integrated into Azure, or a placeholder/internal name for an optimized model available via Azure OpenAI Service or Azure ML.
4. **Cohere Embed 4:**
   * **Role in your project:** Used for **Embeddings for Azure AI Search**.
   * **Type:** An embedding model from Cohere. Embedding models are specifically designed to convert text into numerical vectors (embeddings) that capture the semantic meaning of the text. These embeddings are crucial for **vector search** in Azure AI Search, allowing for highly relevant conceptual matches.
   * **Availability:** Similar to Cohere Command A, available through Azure services that host external LLMs and embedding models.

**In summary, for project, the specific pre-trained models leveraged from "Azure AI Foundry" are Cohere Command A, Mistral Large, Grok-4 Fast, and Cohere Embed 4.** These represent a blend of large language models for understanding, generation, and summarization, and a specialized embedding model for vector search.

**Metrics That may be considered:**

**1. General Search Relevance & User Experience Metrics (Overall System)**

These are crucial for evaluating the end-to-end performance of your search and chatbot system.

* **Precision:** Out of all the documents/answers returned, how many were actually relevant?
* **Recall:** Out of all the truly relevant documents/answers available, how many did the system actually retrieve?
* **F1-Score:** A harmonic mean of Precision and Recall, providing a single metric that balances both.
* **NDCG (Normalized Discounted Cumulative Gain):** A sophisticated metric for ranking quality. It gives higher scores to relevant results that appear higher in the search ranking.
* **Click-Through Rate (CTR):** For a search UI, how often do users click on the top results?
* **User Satisfaction (Surveys/Feedback):** Direct feedback from users on how helpful, accurate, and easy to use the system is.
* **Task Completion Rate:** Can users successfully accomplish their goals using the chatbot/search?
* **Response Latency:** How quickly does the system respond to a query? (Critical for a good user experience).

**2. NLU Agent (Cohere Command A) Metrics**

* **Intent Recognition Accuracy:** How often does the NLU agent correctly identify the user's intent (e.g., "document lookup," "asking a question," "performing an action")?
* **Slot Filling Accuracy:** If the NLU agent extracts entities (like product names, dates, locations) from the query, how accurate are these extractions?
* **Query Transformation Accuracy:** How accurately does the agent convert the natural language query into the correct JSON filter query or SQL statement? This is often measured by comparing the generated query to a human-authored "gold standard."

**3. Data Connector Agent & Query Execution Agent Metrics**

* **Retrieval Accuracy:** How often does the chosen Azure AI Search index contain the information needed to answer the query? (This depends on your indexing quality).
* **Query Performance:**
  + **Latency:** Time taken to execute the search query against Azure AI Search.
  + **Throughput:** Number of queries the service can handle per second.
* **Recall (of documents):** How many of the relevant documents in the index were retrieved by the search query?
* **Precision (of documents):** Out of the documents retrieved, how many were actually relevant to the query?

**4. Summarizer Agent (Mistral Large) Metrics**

* **RAG (Retrieval Augmented Generation) Metrics:**
  + **Faithfulness/Factuality:** Is the summary consistent with the source document(s)? Does it hallucinate information?
  + **Relevance:** How relevant is the summary to the original query or the core content of the document?
  + **Coherence/Readability:** Is the summary grammatically correct, well-structured, and easy to understand?
  + **Conciseness:** Is the summary brief and to the point without losing essential information?
* **ROUGE Scores (Recall-Oriented Understudy for Gisting Evaluation):** Commonly used for summarization, comparing system-generated summaries to human-generated reference summaries based on overlapping words/phrases.
* **BERTScore / MAUVE / Other Embedding-based Metrics:** More advanced metrics that use embeddings to compare the semantic similarity between generated and reference summaries, going beyond simple word overlap.

**5. Embeddings for Azure AI Search (Cohere Embed 4) Metrics**

* **Recall@K:** For a given query, if we consider the top K vector search results, how often do they contain a known relevant document?
* **MRR (Mean Reciprocal Rank):** Measures the rank of the first relevant document. Higher is better.
* **MAP (Mean Average Precision):** A metric that considers the ranking of all relevant documents.
* **Embedding Quality:** Often evaluated indirectly through the performance of downstream tasks like vector search relevance.

**6. Cost Metrics**

Given the pricing models you mentioned ($17.125/gpu/hr for Cohere Command A, Token based API pricing for Grok-4 Fast), **cost-efficiency** will be a crucial metric for your project.

* **Cost per Query:** The total cost of processing a single user query through the entire multi-agent pipeline.
* **Cost per Interaction:** Similar to cost per query, but might encompass a multi-turn conversation.
* **GPU Hours Consumed:** For models with GPU-hour pricing.
* **Tokens Consumed:** For token-based API pricing.

**How to Measure?**

* **Test Datasets:** Create diverse datasets of queries and corresponding ideal responses/documents.
* **Human Evaluation:** For subjective aspects like summarization quality or overall user satisfaction, human evaluators are indispensable.
* **Automated Metrics:** For quantifiable aspects (accuracy, latency, ROUGE scores), use automated tools and scripts.
* **A/B Testing:** For changes, deploy new versions and compare metrics against a baseline.