Microsoft

[Project Moo]

# Azure AI Search와 Azure OpenAI를 활용한 RAG 구현 및 검색

조승민
Partner Solution Architect
Microsoft

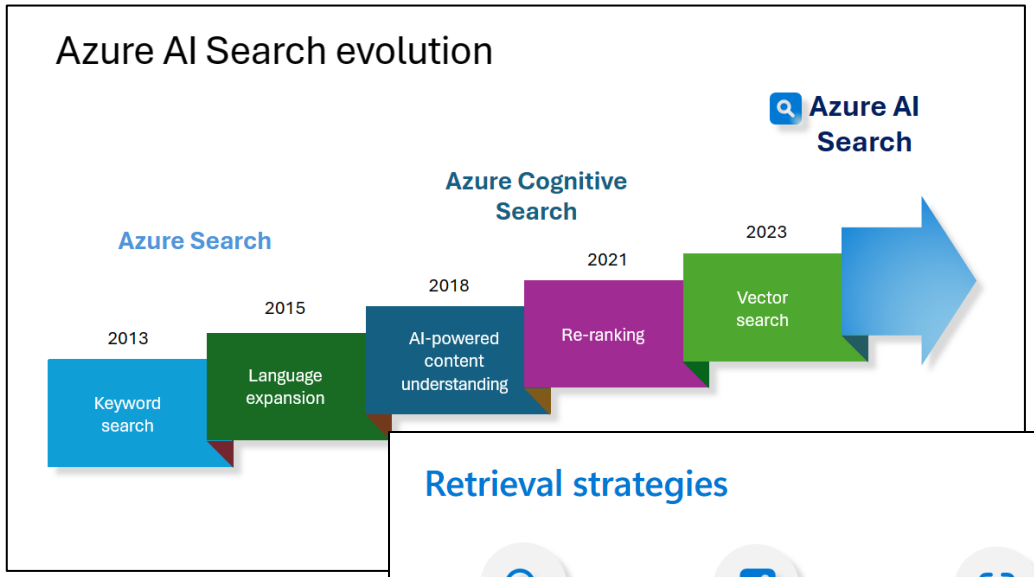
# Azure AI Search

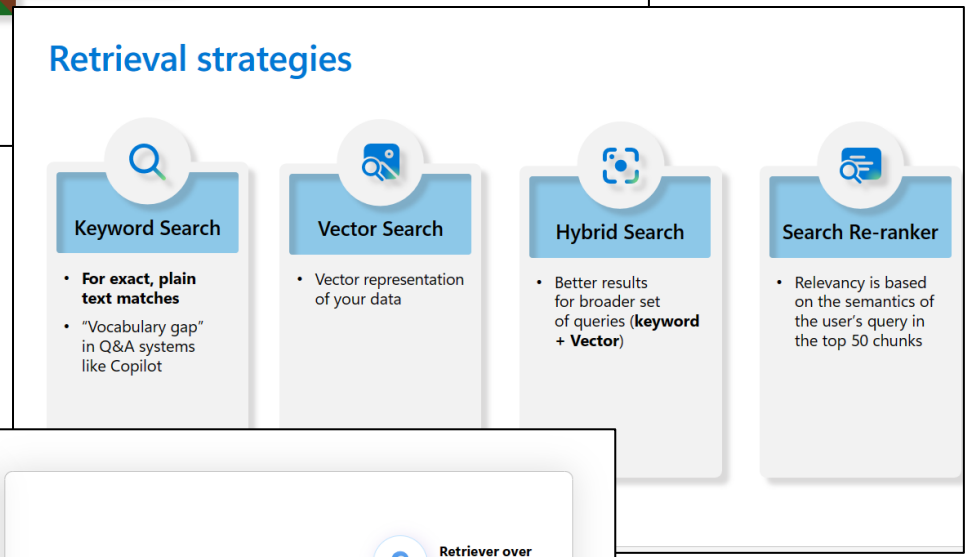
AZURE AI

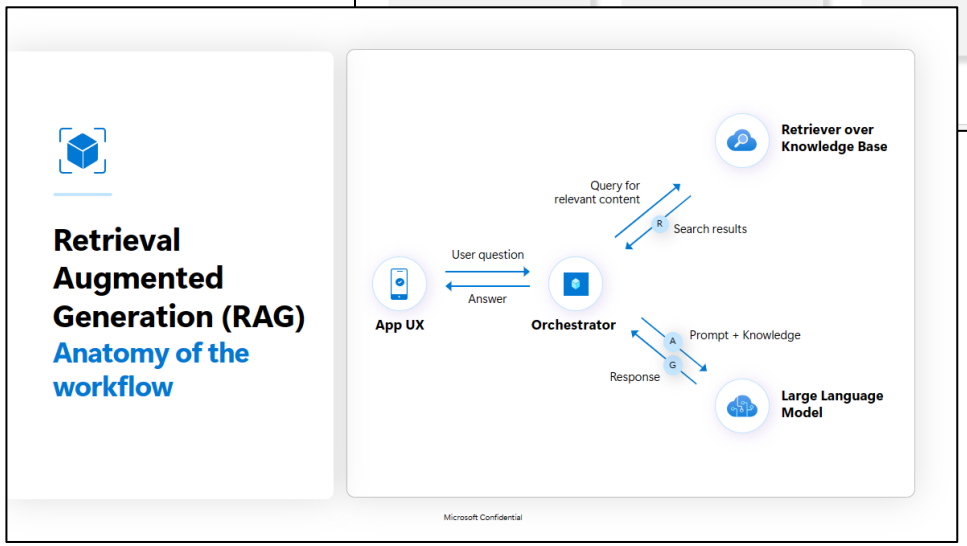
# History & Features

A brief history

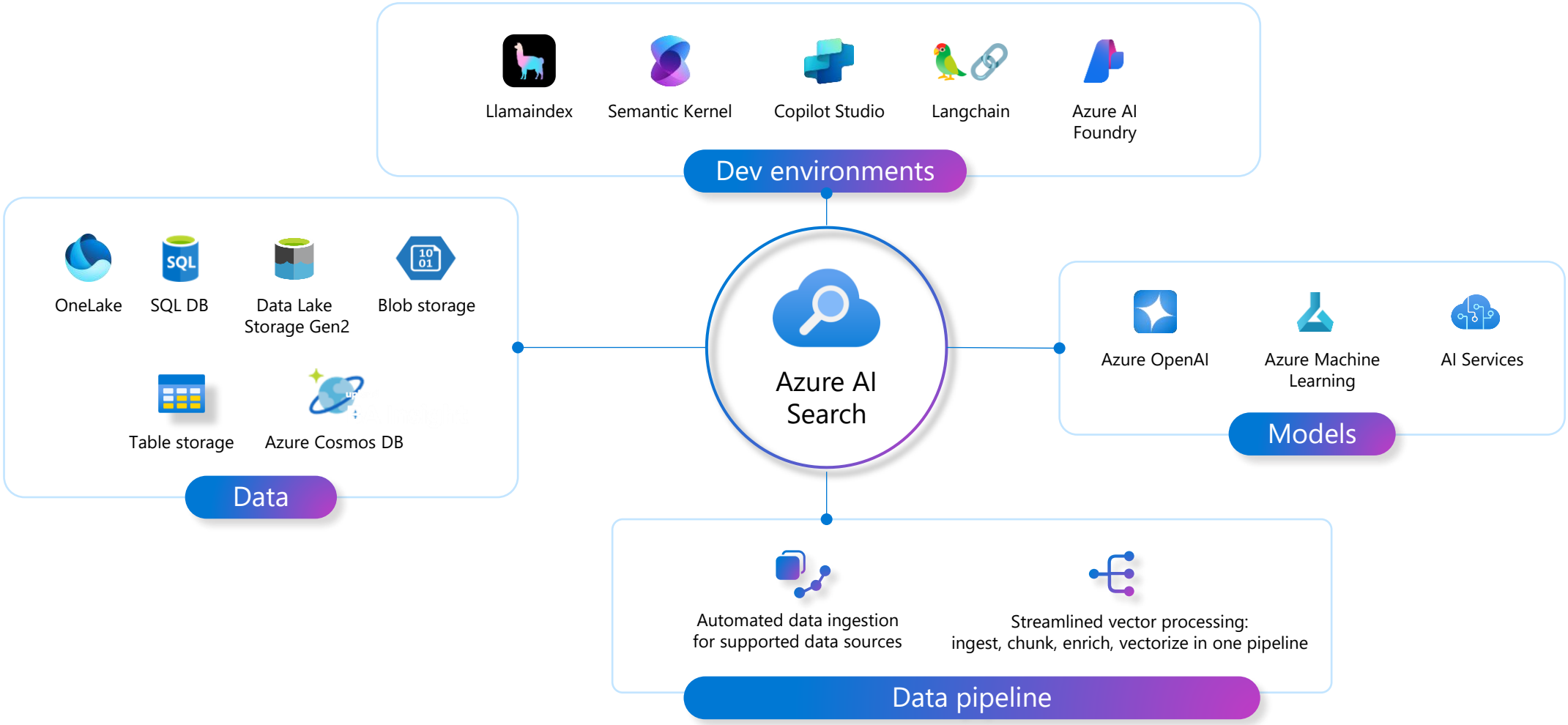
- Mar 2015 – AZS 1.0 GA
- May 2018 – AI enrichment via Cognitive services
- Mar 2021 – AI reranking via partnership with Bing
- Mar 2021 – Serverless R&D begins
- May 2023 – Serverless preview
- Nov 2023 – Vector Search GA
- Dec 2023 – Serverless shelved (focus on Vector) 😢
- Apr 2024 – Megastorage GA
- May 2024 – AIP integration: AI Vision, AML model catalog
- May 2024 – OneLake Connector
- Nov 2025 – RAG for Github
- Nov 2025 – Semantic Query Engine



# Seamless integrations for your GenAI deployments



# Battle-tested retrieval

| | | | |
|---|---|---|---|
| Vector Search (ANN, KNN) | ✓ | Query rewriting | ✓ |
| Hybrid Search (RRF) | ✓ | Multi-vector search | ✓ |
| Search reranking (BM25) | ✓ | Multi-lingual search | ✓ |
| Exact keyword match | ✓ | Geospatial search | ✓ |
| Facets | ✓ | Auto-complete | ✓ |

# Enterprise-ready platform

Data encryption

Authentication

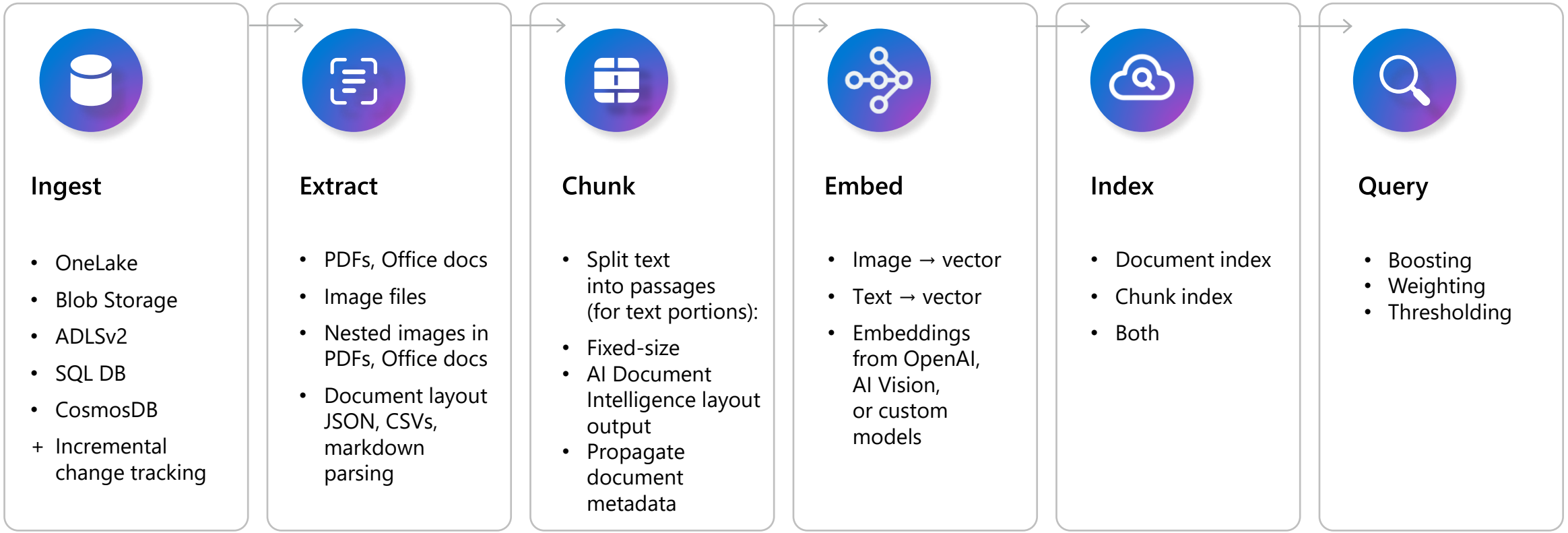
Network security

Data privacy

Compliance

# Streamlined RAG pipeline

## Integrated vectorization and Azure AI Search



# Data ingestion & connectors

Built-in pull indexer support for Azure data sources including OneLake, Blob Storage, Azure SQL, Cosmos DB

Data sources supported by Microsoft Partners: BA Insights has 92 connectors to pull data from including SharePoint Online, Confluence, OpenText

Push API/SDK for any data source not supported with pull method:

- Supported data sources
- Integrated Vectorization
- Data import and ingestion
- Push SDK in RAG

# 1. Data ingestion & connectors

## Use data from all over Azure

- Azure Storage
  - Blob
  - Data Lake Storage Gen2
  - Table
  - Files
- OneLake
- Azure Cosmos DB
  - NoSQL
  - Gremlin
  - MongoDB
- Azure SQL
- Azure Database for MySQL
- A variety of partner-supported data sources

# 2. Extract with AI skillset

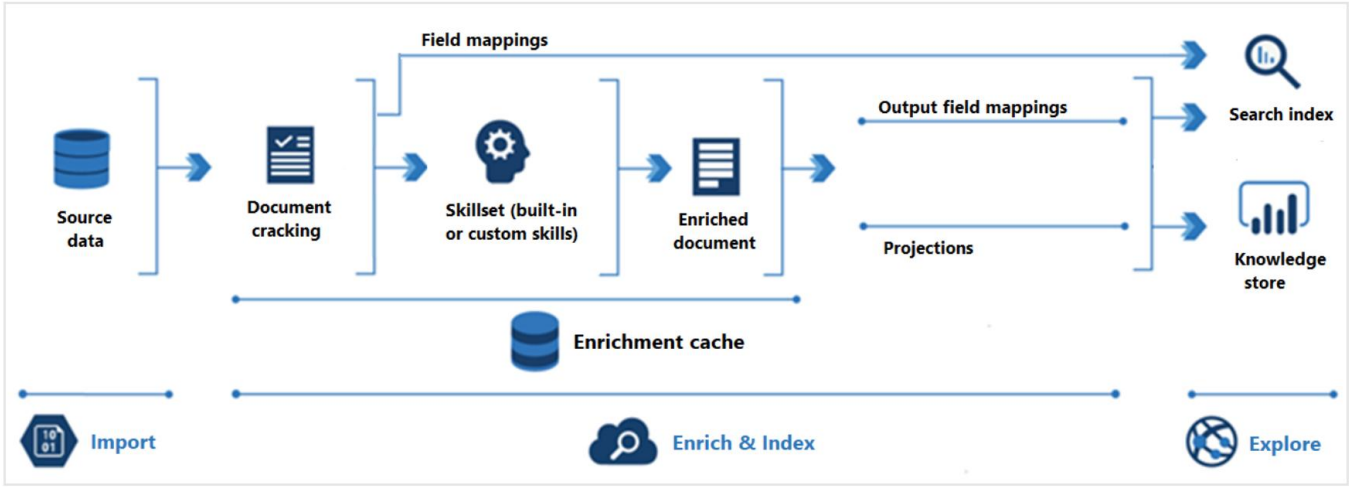
AI enrichment integrates with Azure AI services (among other services) for content processing.

Transforms non-searchable or irrelevant raw content into searchable and structured data

Uses enrichment, analysis, and inference to create structure from unstructured data

Essential for data transformation in RAG scenarios, making data suitable and relevant for the Language Model

I.e., Chunking and Vectorization take place as part of this feature when using integrated Vectorization while indexing



# 3. Chunking strategies

| | | |
|---|---|---|
| **Fixed-size Chunking** | Simple and common, maintaining a balance between overlap and semantic context | AI Search Built-in Split Skill |
| **Specialized Chunking** | For structured content like Markdown and LaTeX | AI Search Built-in Markdown Parsing |
| **“Structure-aware” Chunking** | Content splitting using NLP libraries such as AI Document Intelligence Layout API | AI Search Built-in Doc Layout Skill |
| **Recursive Chunking** | Divides text into smaller Chunks in a hierarchical manner | AI Search Custom Skill |

Preprocessing data for quality

Selecting a range of potential Chunk sizes considering content nature and embedding model capabilities

Evaluating performance of each Chunk size

# 4. embedding..

# 5. Index

## Inverted indexes

- Tokenized content →keyword search
- Values → fast filters
- Edge n-grams → suggestions, autocomplete
- Tracks frequencies, offsets, etc.

## DocValue indexes

- Column store-like structures
- Sorting, faceting

## HNSW indexes

- Approximate nearest neighbor (ANN) vector search
- Original precision or quantized vectors in the index

# Query

# 6. Retrieval strategies

## Keyword search

- **For exact, plain text matches**
- “Vocabulary gap” in Q&A systems like Copilot

## Vector search

- **For conceptual similarity, or underlying meaning**
- Weak performance on exact matches (like a product ID or code)

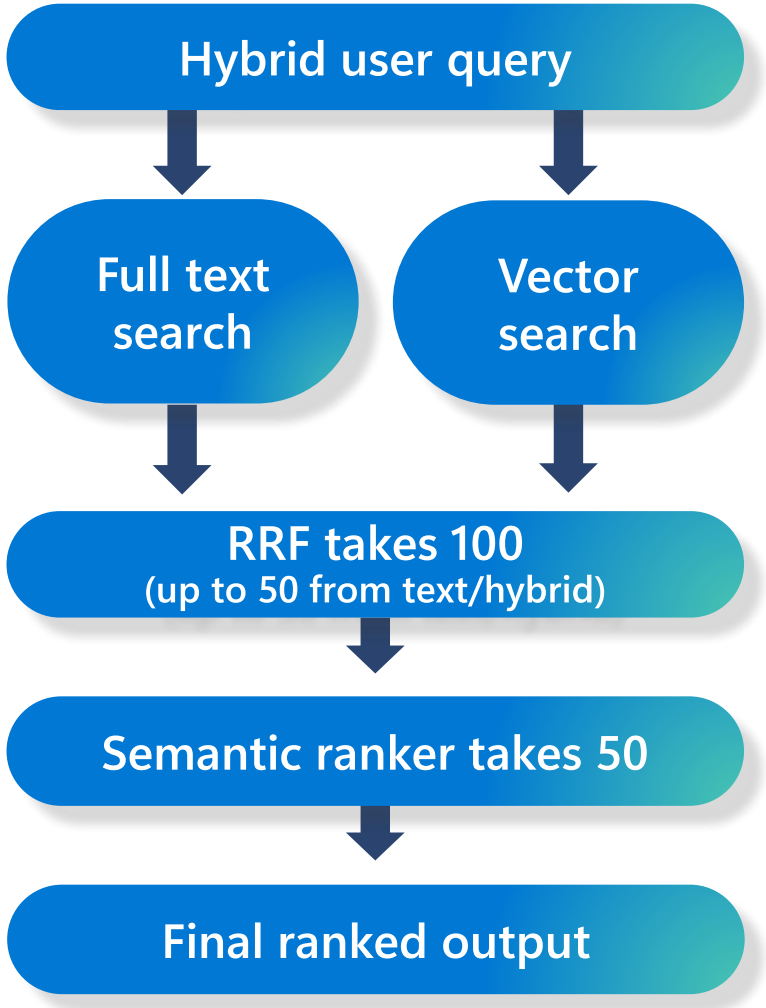
## Hybrid search

- **Best of both vectors and keywords**
- Brings more accurate responses across various scenarios

## Search re-ranking

- **Scores and ranks all retrieved documents by relevance**
- Reranking runs after performing search strategy (can’t retrieve information)

# 7. User Query




## Enhancing semantic ranker performance:

- Semantic ranker output is only as good as your L1 retrieval results
- Larger Chunk sizes represent lossy compression; Chunk to 512 tokens if possible
- RRF can take up to 50 from both full text and vector so set “k” to 50 for balanced retrieval representation
- Optimize the rankers’ performance by prioritizing the fields in your semantic config
- Use and understand @search.rerankerScore to meet the relevancy threshold of your use case

# 실습:
# Azure Portal에서 문서 및 이미지 벡터 검색

# Azure Portal에서 문서 벡터 검색

1. Azure AI Search 구성
2. Azure OpenAI 구성: 텍스트 임베딩 모델
3. Azure AI Foundry Hub 구성: 스토리지
4. Azure AI Search 데이터 가져오기 및 벡터화 마법사
5. Azure AI Search 쿼리

# Azure Portal에서 문서 이미지 검색

1. Azure AI Search 구성
2. Azure OpenAI 구성: 텍스트 임베딩 모델
3. Azure AI Foundry Hub 구성: 스토리지
4. Azure AI Foundry Project 구성: 이미지 임베딩 모델
5. Azure AI Multimodel Service 구성: skillset
6. Azure AI Search 데이터 가져오기 및 벡터화 마법사
7. Azure AI Search 쿼리

# 실습:
# REST API로 RAG 구성 및 벡터 검색

# REST API로 RAG 구성 및 벡터 검색

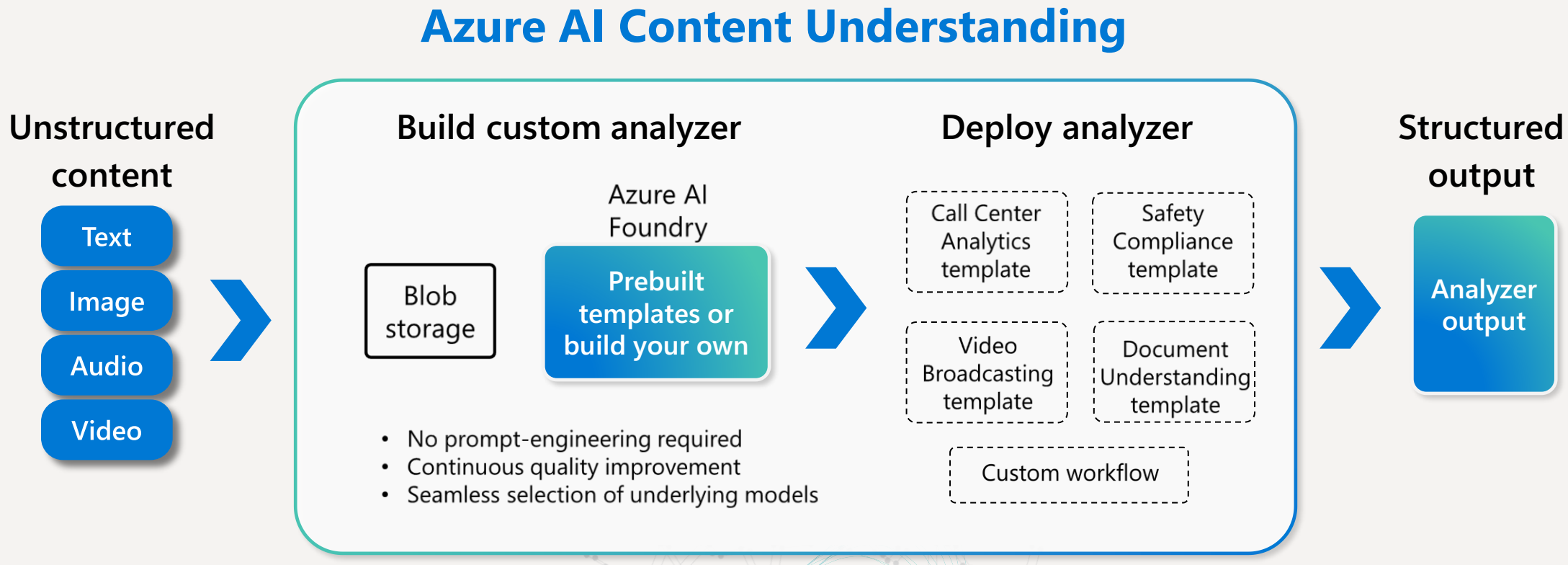
1. Azure AI Search 구성
2. Azure OpenAI 구성: 텍스트 임베딩 모델
3. Python 환경 구성 (VS Code)
4. .ipynb 실행

# Azure AI Content Understanding

# Azure AI Content Understanding

Built on the success of Document Intelligence, Azure AI Content Understanding offers a new way to reason over large amounts of unstructured data to build customizable workflows, ultimately accelerating time-to-value (TTV), while using a variety of AI models. Ingest and extract any data type into customizable output formats and use your domain expertise to ensure accuracy and improve input.

# Product overview



# Azure AI Content Understanding capabilities

## Multimodal data ingestion

- Ingest a range of modalities, including documents, images, audio, or video.
- Use a range of AI models in Azure AI to convert input data into a structured format.
- Process and analyze through downstream services or applications.

## Information extraction schema

- Define schemas of extracted results.
- Generate task-specific representations of the output, including insights, features, or summaries.
- Build enterprise GenAI apps or agentic workflows to automate business processes through LLMs and RAG.

## Grounding and confidence scores

- Ensure schema-identified values are accurate and usable with GenAI tools.
- Ground extracted information in underlying content.
- Provide confidence scores to reduce human intervention.
- Enable continuous improvement through user feedback.

# 실습:
# Azure AI Foundry에서 문서 및 영상 분석기 빌드

# Azure AI Foundry에서 문서 및 영상 분석기 빌드

1. Azure AI Foundry Hub 구성: 스토리지 및 Azure AI Service
2. Azure AI Foundry Project 구성: Content Understanding
3. 스키마 정의
4. 데이터 레이블
5. 분석기 테스트
6. 분석기 빌드

# 실습:
# REST API로 RAG 구성 및 질문

# REST API로 RAG 구성 및 질문

1. Azure AI Search 구성
2. Azure AI Service 구성
3. Azure OpenAI 구성: 텍스트 임베딩 모델, GPT 모델
4. Python 환경 구성 (VS Code)
5. .ipynb 실행

AZURE AI

# Agenda

- **Azure AI Search** 소개
- 실습: Azure Portal에서 문서 및 이미지 벡터 검색
- 실습: REST API로 RAG 구성 및 벡터 검색 (python)

- **Azure AI Content Understanding** 소개
- 실습: Azure AI Foundry에서 문서 및 영상 분석기 빌드
- 실습: REST API로 RAG 구성 및 질문 (python)

# 리소스 삭제