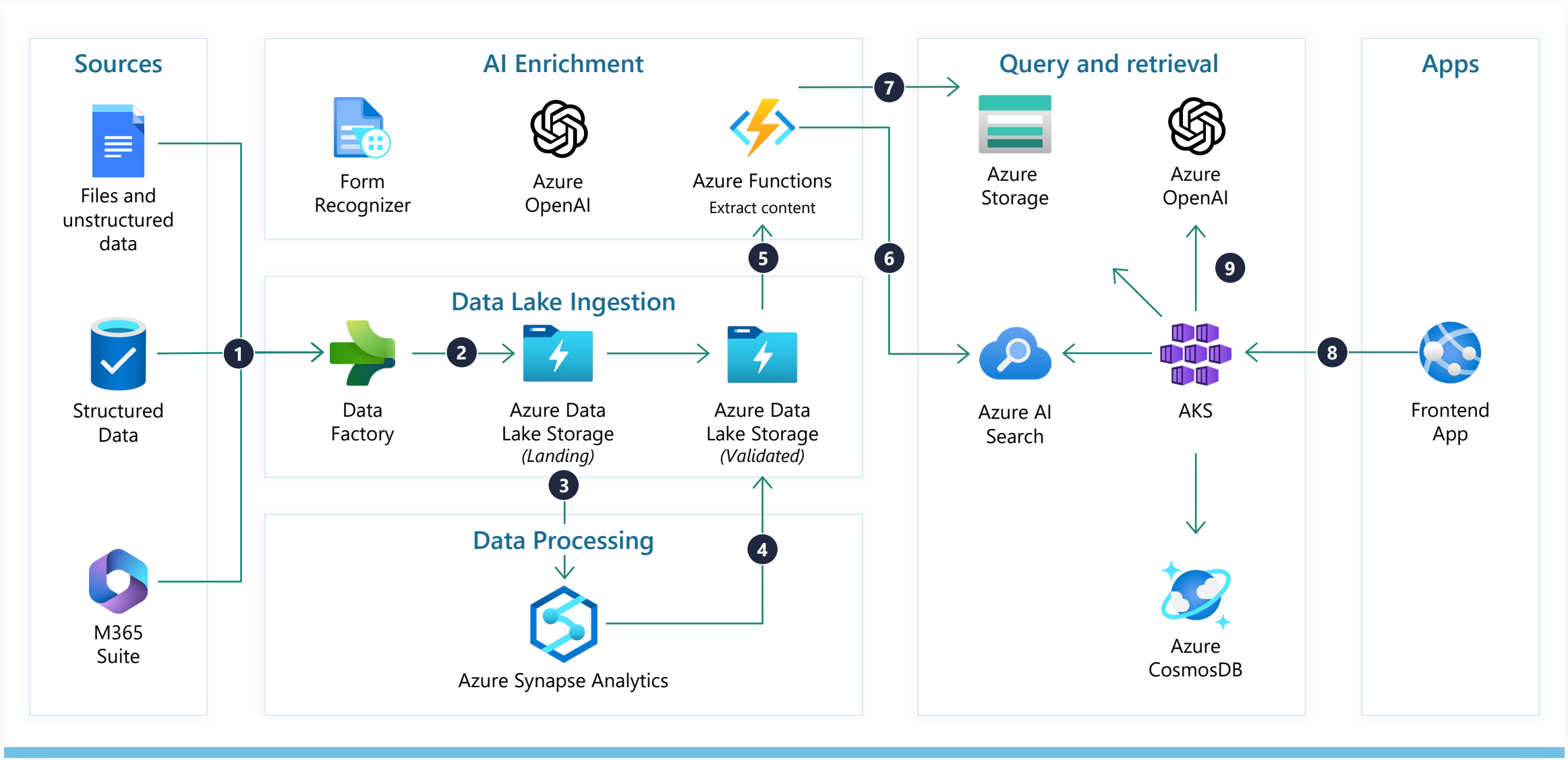
Create knowledge hubs

Create knowledge hubs reference architecture



Create knowledge hubs reference architecture

1. Azure Data Factory pipelines are designed to handle a wide range of data types, including unstructured, semi-structured, and structured data. The pipelines transfer this data into Azure Data Lake Storage, a centralized repository for storage and analysis. This integration allows for combining data with other sources like SharePoint. The extraction of data from source systems is facilitated by Azure Data Factory's diverse range of pre-built connectors.

2. The data ingested through Data Factory pipelines undergoes processing in distinct stages. During these stages, the data is stored in Azure Storage accounts, organized within specific directories in Azure Data Lake Storage Gen 2. Each directory corresponds to a particular stage of the data processing pipeline, ensuring organized and efficient data management.

   Initially, Azure Data Factory activities ingest data storing it in its raw format. Subsequent activities involve applying data quality rules to cleanse and enrich this raw data.

3. Following the initial data ingestion, Synapse Analytics steps in to facilitate data analysis and transformation. Through Synapse Analytics, complex queries are executed, and data exploration is conducted, allowing for further refinement and enrichment. The outcome is a structured and enriched dataset, prepared for further analysis and processing in subsequent stages.

4. The refined data is then stored in a 'validated' directory within the data lake, signifying its readiness for further analysis or processing. Documents ingested into Azure Data Lake Storage trigger Azure functions based on the document type. For PDFs or image files, Azure AI Document Intelligence is used to extract text. In contrast, Python code is employed for text extraction from Excel, CSV, Word, or plain text files. The extracted text is then segmented ('chunked'), and Azure OpenAI embedding models are used to convert each text chunk into embeddings, which are a form of data representation useful for further analysis. Additionally, predefined custom skills can be leveraged to further process and analyze the extracted data, enhancing the capabilities of Azure AI Search pipelines. These custom skills allow for the application of specific processing or transformation logic, such as entity recognition, text summarization, or content classification, tailored to the needs of the application.

Create knowledge hubs reference architecture

5. Following the enrichment and conversion to embeddings, these data representations are stored in a vector database within Azure AI Search. This database is optimized for searching and retrieving information based on the embeddings, facilitating efficient data access and analysis.

6. The final output can optionally include the creation of a ‘knowledge store’. This store comprises blobs and tables within Azure Storage, which are made accessible for data exploration and analysis through various tools. This knowledge store can also be integrated into downstream processes for further applications, such as data analytics, machine learning, or business intelligence.

7. User submits a query through a web application interface. This query is directed to a chatbot application, which is designed to handle user interactions and process queries in real time.

8. Upon receiving the query, the Azure OpenAI embedding model is employed to transform the user’s query into vector embeddings. These embeddings capturing its semantic meaning in a format that is suitable for machine processing.

   The vector database, which contains embeddings from various sources, is then searched using the query vector. A vector similarity search is conducted to find the most relevant content. The search returns the top ‘k’ matching pieces of content, where ‘k’ is a predefined number, and the matching threshold is determined by the similarity score. This ensures that the results are closely aligned with the user's query.

   Integration with Azure OpenAI Language Model: The selected top ‘k’ pieces of content, along with a system-generated prompt, are then sent to an Azure OpenAI language model, such as GPT-3.5 Turbo or GPT-4. This advanced language model processes the content and the prompt to generate a coherent and contextually relevant response.