

Creating a Knowledge Mining Solution



Agenda



- Implementing an Intelligent Search Solution
- Developing Custom Skills for an Enrichment Pipeline
- Creating a Knowledge Store

Implementing an Intelligent Search Solution



Learning Objectives

After completing this module, you will be able to:

- 1 Create an Azure AI Search Solution
- 2 Implement a custom skill for Azure AI Search and integrate it into a skillset
- 3 Create a knowledge store with object, file, and table projections

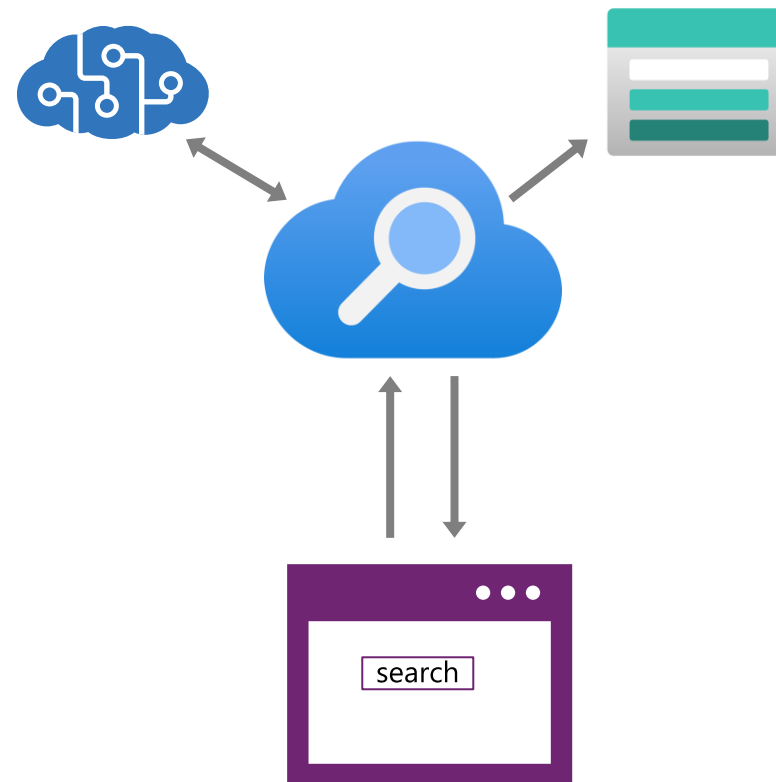
Azure AI Search

AI-Powered Knowledge Mining

- Index documents and data from a range of sources
- Use skills to enrich index data
- Store extracted insights in a knowledge store for analysis and integration

Azure Resources:

- **Azure AI Search** for core indexing and querying
- **Azure AI Services** for index enrichment
- **Storage account** for knowledge store persistence



Core Components of a AI Search Solution



Data Source

The data store to be searched:

- Blob storage container
- SQL Database
- Cosmos DB

You can also push JSON documents directly into an index



Skillset

Defines an enrichment pipeline of AI skills to enhance data during indexing:

- Built-in AI skills
- Custom skills



Indexer

Maps data source fields and skillset outputs to index fields

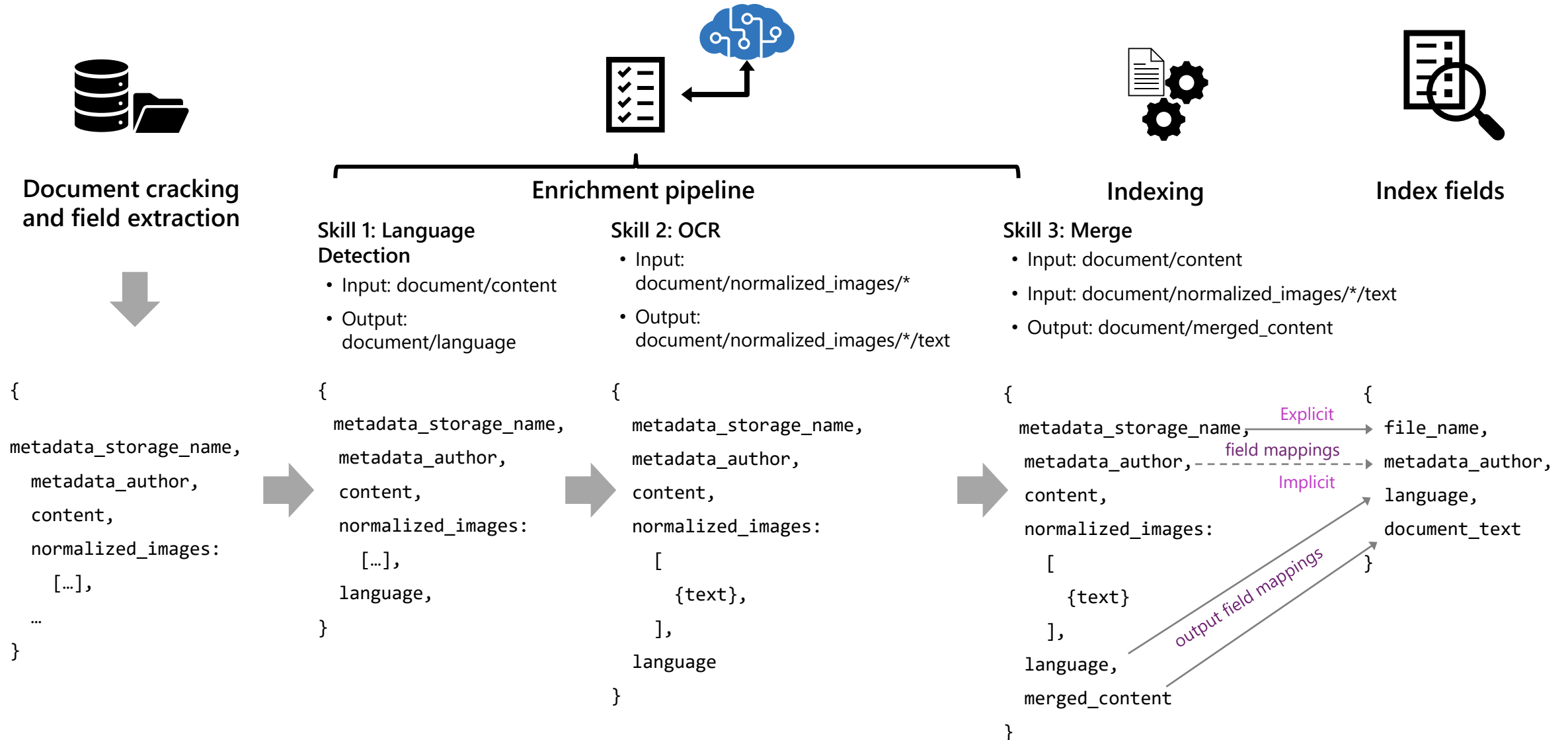
- Running the indexer builds the index



Index

Searchable collection of JSON documents containing extracted and enriched fields

How an Enrichment Pipeline Works



[Optional] Create an Azure Cognitive Search Solution



Create an indexing solution

Modify an indexing solution

Query an index from a client application

Create a custom skill for Azure AI Search



Introduction to Custom Skills

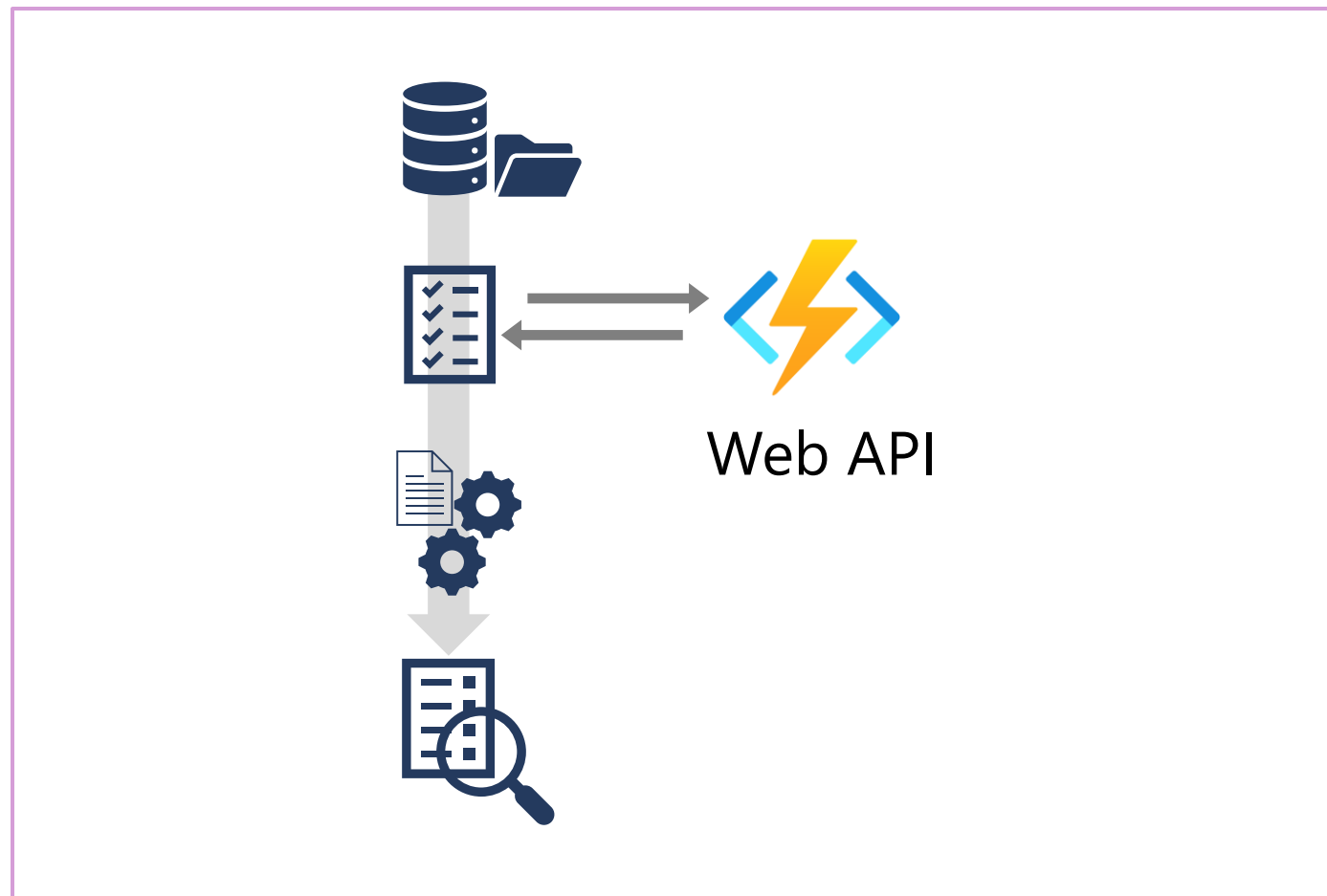
When built-in skills don't provide what you need...

Create a custom skill, for example:

- Integrate Document Intelligence
- Consume an Azure Machine Learning model
- Any other custom logic

Custom skills are implemented as Web APIs

- Commonly Azure Functions



Custom Skill Interfaces

Input Schema

```
{
  "values": [
    {
      "recordId": "<unique_identifier>",
      "data": {
        "<input1_name>": "<input1_value>",
        "<input2_name>": "<input2_value>",
        ...
      }
    },
    {
      "recordId": "<unique_identifier>",
      "data": {
        "<input1_name>": "<input1_value>",
        "<input2_name>": "<input2_value>",
        ...
      }
    },
    ...
  ]
}
```

Output Schema

```
{
  "values": [
    {
      "recordId": "<unique_identifier_from_input>",
      "data": {
        "<output1_name>": "<output1_value>",
        ...
      },
      "errors": [...],
      "warnings": [...]
    },
    {
      "recordId": "< unique_identifier_from_input>",
      "data": {
        "<output1_name>": "<output1_value>",
        ...
      },
      "errors": [...],
      "warnings": [...]
    },
    ...
  ]
}
```

This is a *property bag* of values – it can be a single value or a complex JSON structure

Adding a Custom Skill to a Skillset

Add a Custom.WebApiSkill to the skillset

Specify the URI to your web API endpoint

- Optionally add parameters and headers

Set the context to specify at which point in the document hierarchy the skill should be called

Assign input values

- Usually from existing document fields

Store output in a new field

- Optionally, specify a target field name (otherwise the output name is used)

```
{
  "skills": [
    ...,
    {
      "@odata.type": "#Microsoft.Skills.Custom.WebApiSkill",
      "description": "<custom skill description>",
      "uri": "https://<web_api_endpoint>?<params>",
      "httpHeaders": {
        "<header_name>": "<header_value>"
      },
      "context": "/document/<where_to_apply_skill>",
      "inputs": [
        {
          "name": "<input1_name>",
          "source": "/document/<path_to_input_field>"
        }
      ],
      "outputs": [
        {
          "name": "<output1_name>",
          "targetName": "<optional_field_name>"
        }
      ]
    }
  ]
}
```

Exercise – Create a Custom Skill for Azure AI Search



Use an Azure Function to implement a custom skill
Integrate a custom skill into a skillset

Creating a Knowledge Store



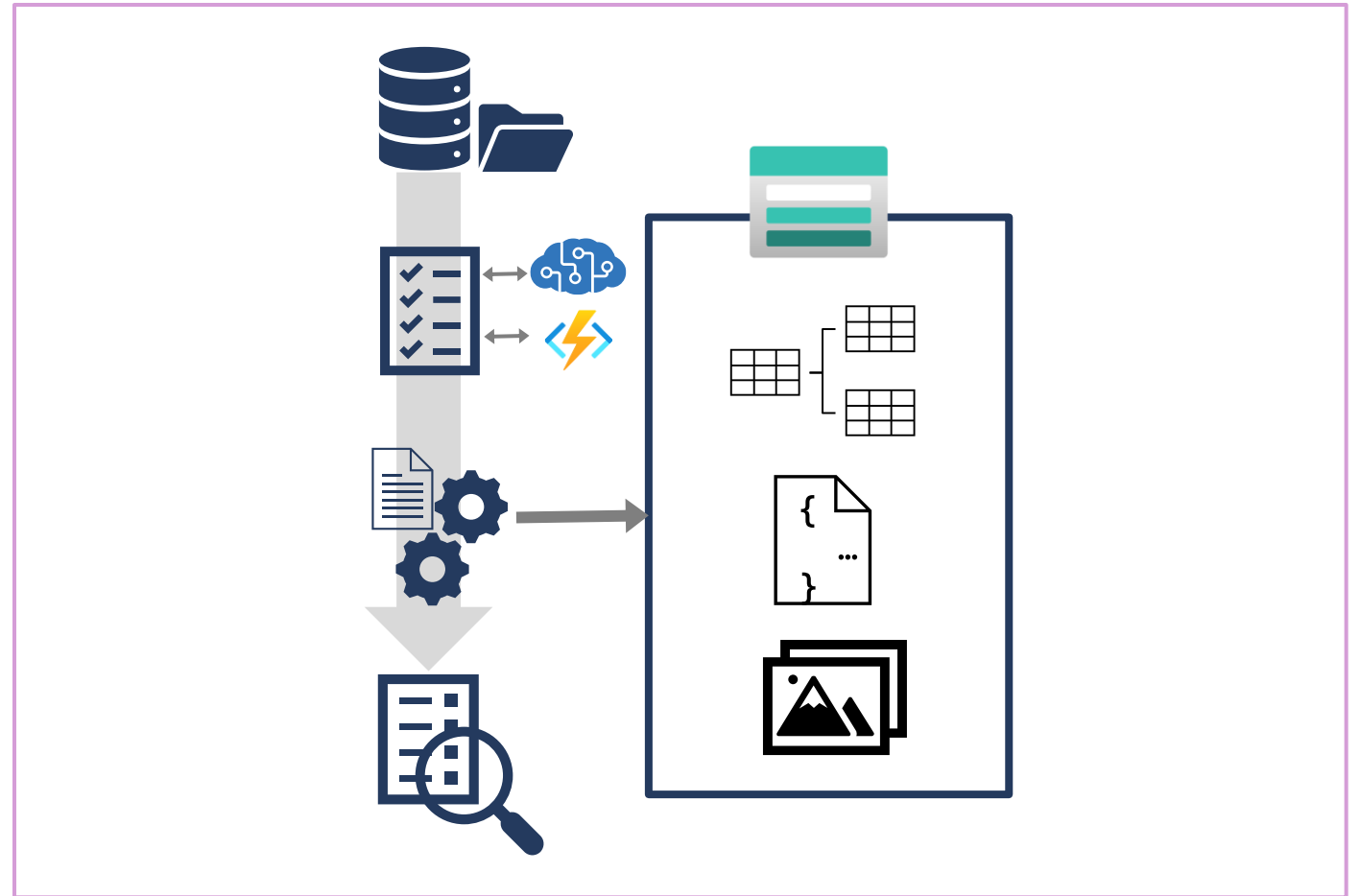
What is a Knowledge Store?

Persisted insights extracted by indexing process

Stored as *projections* in *Azure Storage*

- **Tables:** Relational tables with keys for joining
- **Objects:** JSON structures of document fields
- **Files:** Extracted images saved in JPG format

Used for analysis or integration into data processing workflows



Using the Shaper Skill for Projections

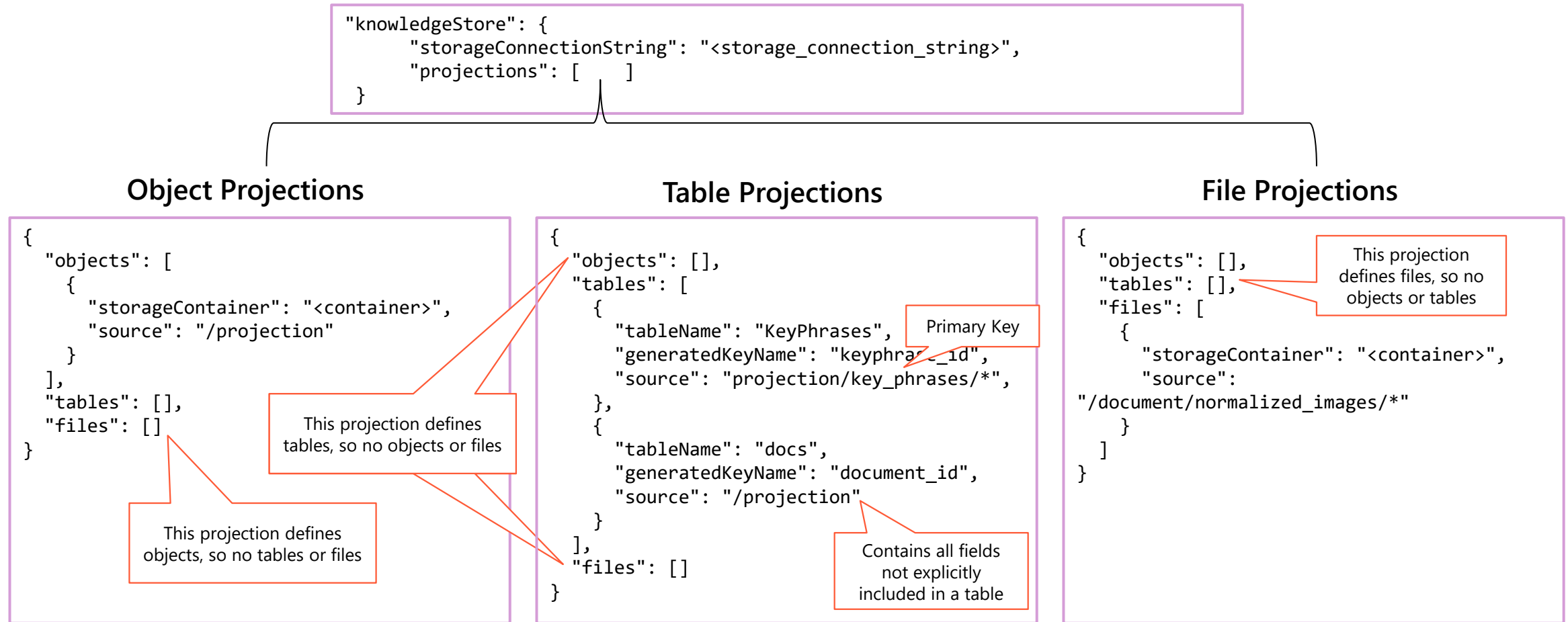
Restructure fields to simplify projections

- Create a JSON object with the fields you want to persist
- Use sourceContext and inputs to map primitives to well-formed JSON objects

```
{
  "@odata.type": "#Microsoft.Skills.Util.ShaperSkill",
  "name": "define-projection",
  "description": "Prepare projection fields",
  "context": "/document",
  "inputs": [
    {
      "name": "url",
      "source": "/document/url"
    },
    {
      "name": "sentiment",
      "source": "/document/sentiment"
    },
    {
      "name": "key_phrases",
      "source": null,
      "sourceContext": "/document/merged_content/keyphrases/*",
      "inputs": [
        {
          "name": "phrase",
          "source": "/document/merged_content/keyphrases/*"
        }
      ]
    }
  ],
  "outputs": [
    {
      "name": "output",
      "targetName": "projection"
    }
  ]
}
```


Implementing a Knowledge Store

Knowledge Store and Projections are defined in the Skillset



Extended interactive exercises – Create a Knowledge Store with Azure AI Search



<https://aka.ms/km-ai-lp>

Knowledge check



1 You want to enrich an index by extracting any geographical locations mentioned in the source data. Which built-in skill should you use?

- ☒ Entity Recognition
- ☐ Key Phrase Extraction
- ☐ Language Detection

2 You have implemented a custom skill as an Azure function. How can you include the custom skill in your indexing process?

- ☐ Add a Merge skill to the skillset to combine output from built-in skills with your custom skill.
- ☒ Add a WebApiSkill to a skillset, referencing the Azure function's URI
- ☐ Add a Shaper skill to the skillset to create a collection of records with unique IDs generated by your custom

3 You want to create a knowledge store that contains JSON representations of the extracted data. What kind of projection should you define?

- ☐ File
- ☐ Object
- ☒ Table

Learning Path Recap

In this learning path, we learned how to:

Create an Azure AI Search Solution

Implement a custom skill for Azure AI Search and integrate it into a skillset

Create a knowledge store with object, file, and table projections

