Microsoft

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

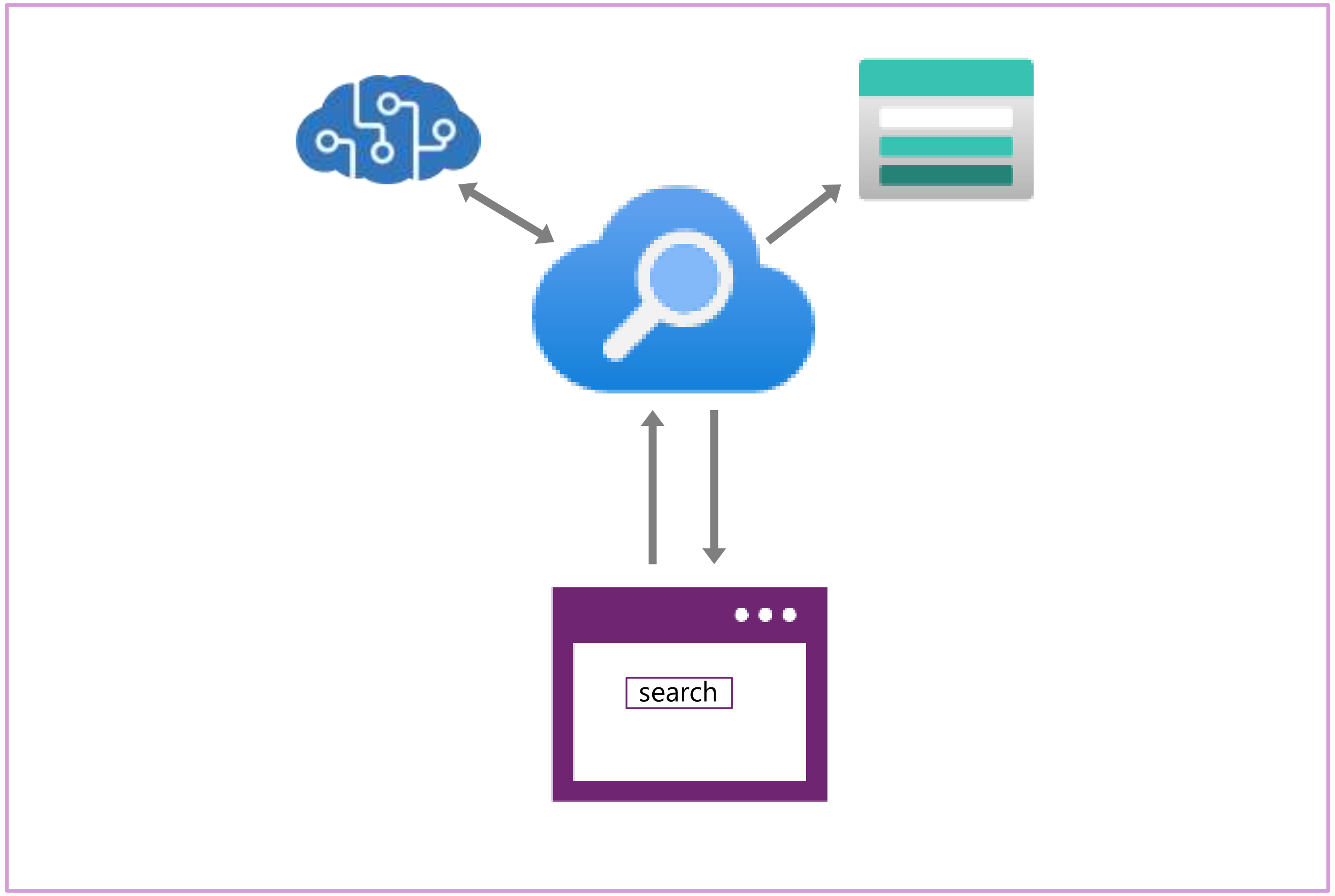
Azure AI Search

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

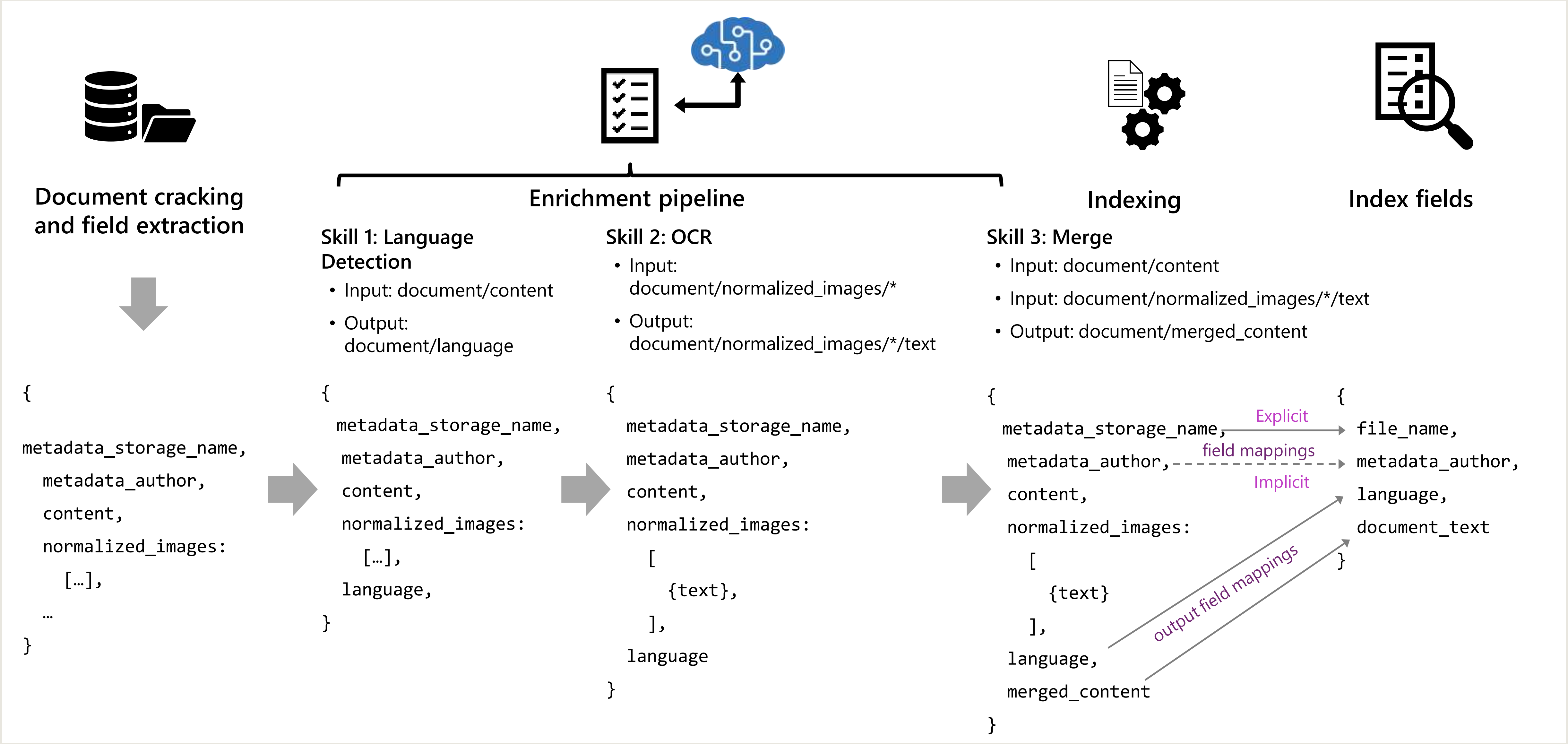
Maps data source fields and skillset outputs to index fields

- Running the indexer builds the index

Index

Searchable collection of JSON documents containing extracted and enriched fields

How an Enrichment Pipeline Works



Document cracking and field extraction

```
{
metadata_storage_name,
  metadata_author,
  content,
  normalized_images:
    […],
  …
}
```

Enrichment pipeline

Skill 1: Language Detection
- Input: document/content
- Output: document/language

```
{
  metadata_storage_name,
  metadata_author,
  content,
  normalized_images:
    […],
  language,
}
```

Skill 2: OCR
- Input: document/normalized_images/*
- Output: document/normalized_images/*/text

```
{
  metadata_storage_name,
  metadata_author,
  content,
  normalized_images:
    [
      {text},
    ],
  language
}
```

Indexing

Skill 3: Merge
- Input: document/content
- Input: document/normalized_images/*/text
- Output: document/merged_content

```
{
  metadata_storage_name,
  metadata_author,
  content,
  normalized_images:
    [
      {text}
    ],
  language,
  merged_content
}
```

Index fields

```
{
  file_name,
  metadata_author,
  language,
  document_text
}
```

Create a custom skill for Azure AI Search

Introduction to Custom Skills

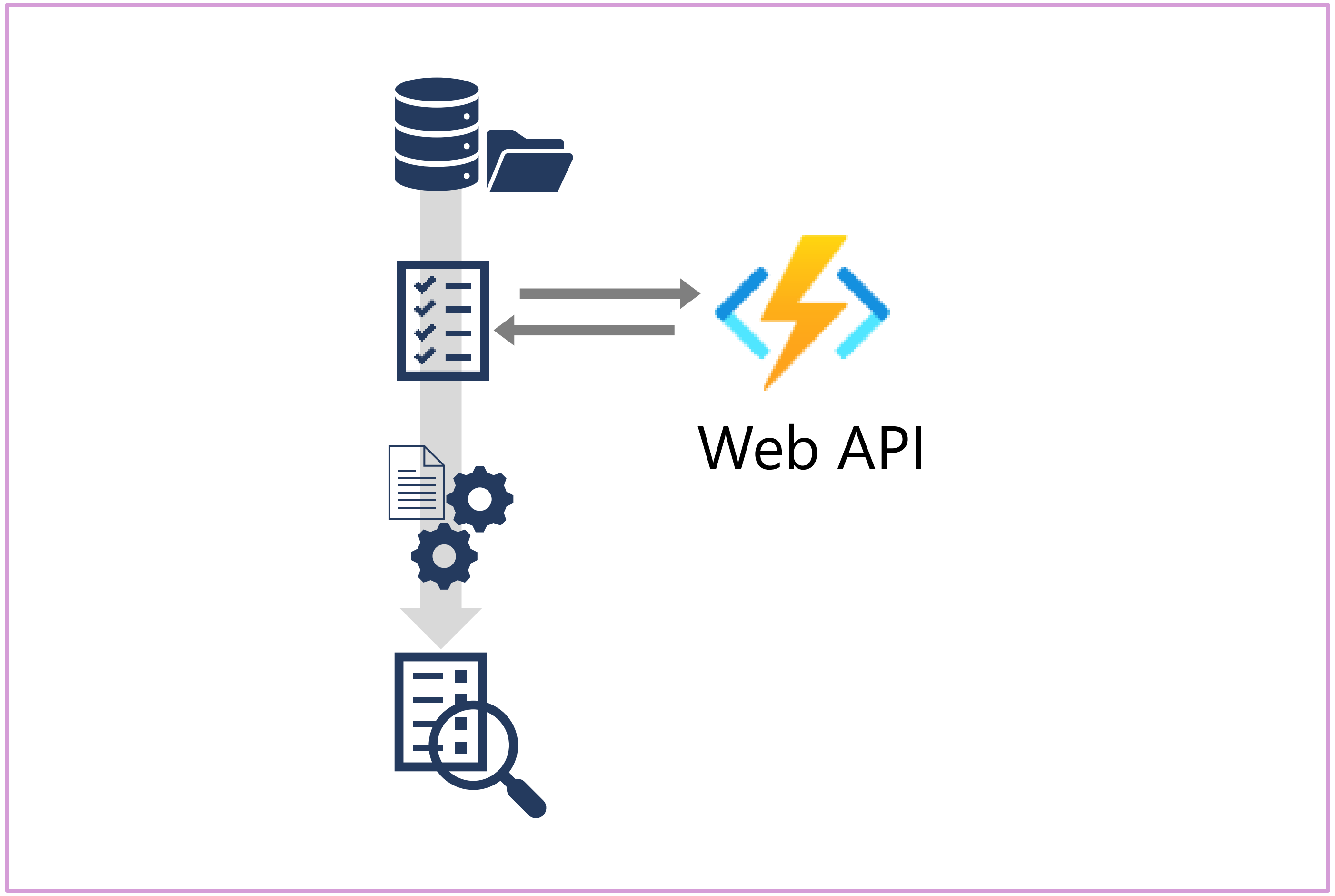
When built-in skills don't provide what you need…

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>"
          }
        ]
      }
  ]
}
```

Creating a Knowledge Store

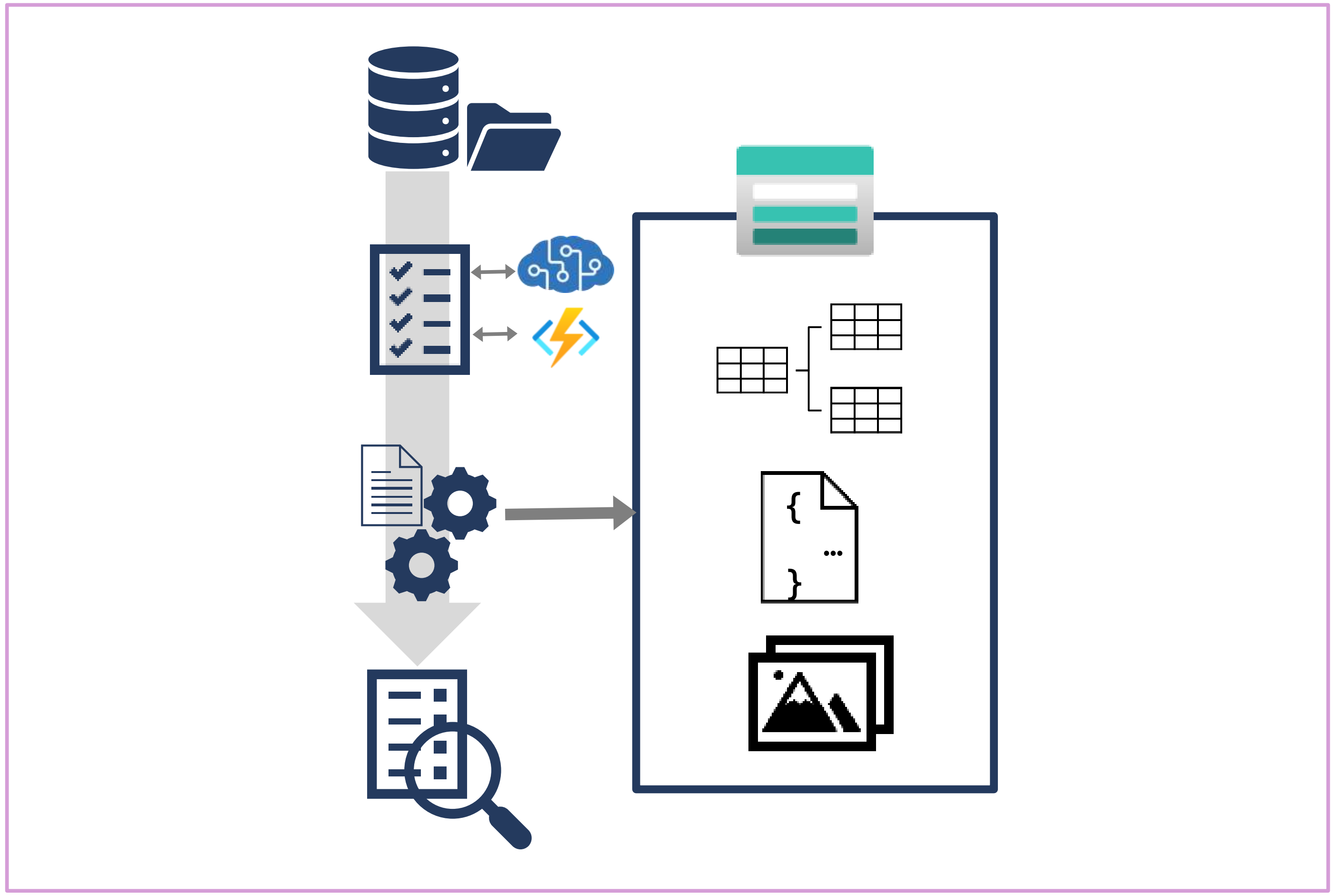
What is a Knowledge Store?

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

```
"knowledgeStore": {
      "storageConnectionString": "<storage_connection_string>",
      "projections": [    ]
 }
```

Object Projections

```
{
  "objects": [
    {
      "storageContainer": "<container>",
      "source": "/projection"
    }
  ],
  "tables": [],
  "files": []
}
```

This projection defines objects, so no tables or files

Table Projections

```
{
 "objects": [],
 "tables": [
   {
     "tableName": "KeyPhrases",
     "generatedKeyName": "keyphrase_id",
     "source": "projection/key_phrases/*",
   },
   {
     "tableName": "docs",
     "generatedKeyName": "document_id",
     "source": "/projection"
   }
 ],
 "files": []
}
```

This projection defines tables, so no objects or files

Primary Key

Contains all fields not explicitly included in a table

File Projections

```
{
  "objects": [],
  "tables": [],
  "files": [
    {
      "storageContainer": "<container>",
      "source":
"/document/normalized_images/*"
    }
  ]
}
```

This projection defines files, so no objects or tables

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

Microsoft