Get started with prompt flow to develop language model apps in the Microsoft Foundry

1. Introduction

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/1-introduction

Introduction

Completed

- 3 minutes

The true power of **Large Language Models** (**LLMs**) lies in their application. Whether you want to use LLMs to classify web pages into categories, or to build a chatbot on your data. To harness the power of the LLMs available, you need to create an application that combines your data sources with LLMs and generates the desired output.

To develop, test, tune, and deploy LLM applications, you can use **prompt flow**, accessible in the Azure Machine Learning studio and the Microsoft Foundry portal.

> **Note**
>
> The focus of this module is on understanding and exploring prompt flow through Microsoft Foundry. However, note that the content applies to the prompt flow experience in both Azure Machine Learning and Microsoft Foundry.

Prompt flow takes a **prompt** as input, which in the context of LLMs, refers to the query provided to the LLM application to generate a response. It's the text or set of instructions given to the LLM application, prompting it to generate output or perform a specific task.

For example, when you want to use a text generation model, the prompt might be a sentence or a paragraph that initiates the generation process. In the context of a question-answering model, the prompt could be a query asking for information on a particular topic. The effectiveness of the prompt often depends on how well it conveys the user's intent and the desired outcome.

Prompt flow allows you to create **flows**, which refers to the sequence of actions or steps that are taken to achieve a specific task or functionality. A flow represents the overall process or pipeline that incorporates the interaction with the LLM to address a particular use case. The flow encapsulates the entire journey from receiving input to generating output or performing a desired action.

2. Understand the development lifecycle of a large language model (LLM) app

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/2-understand-lifecycle

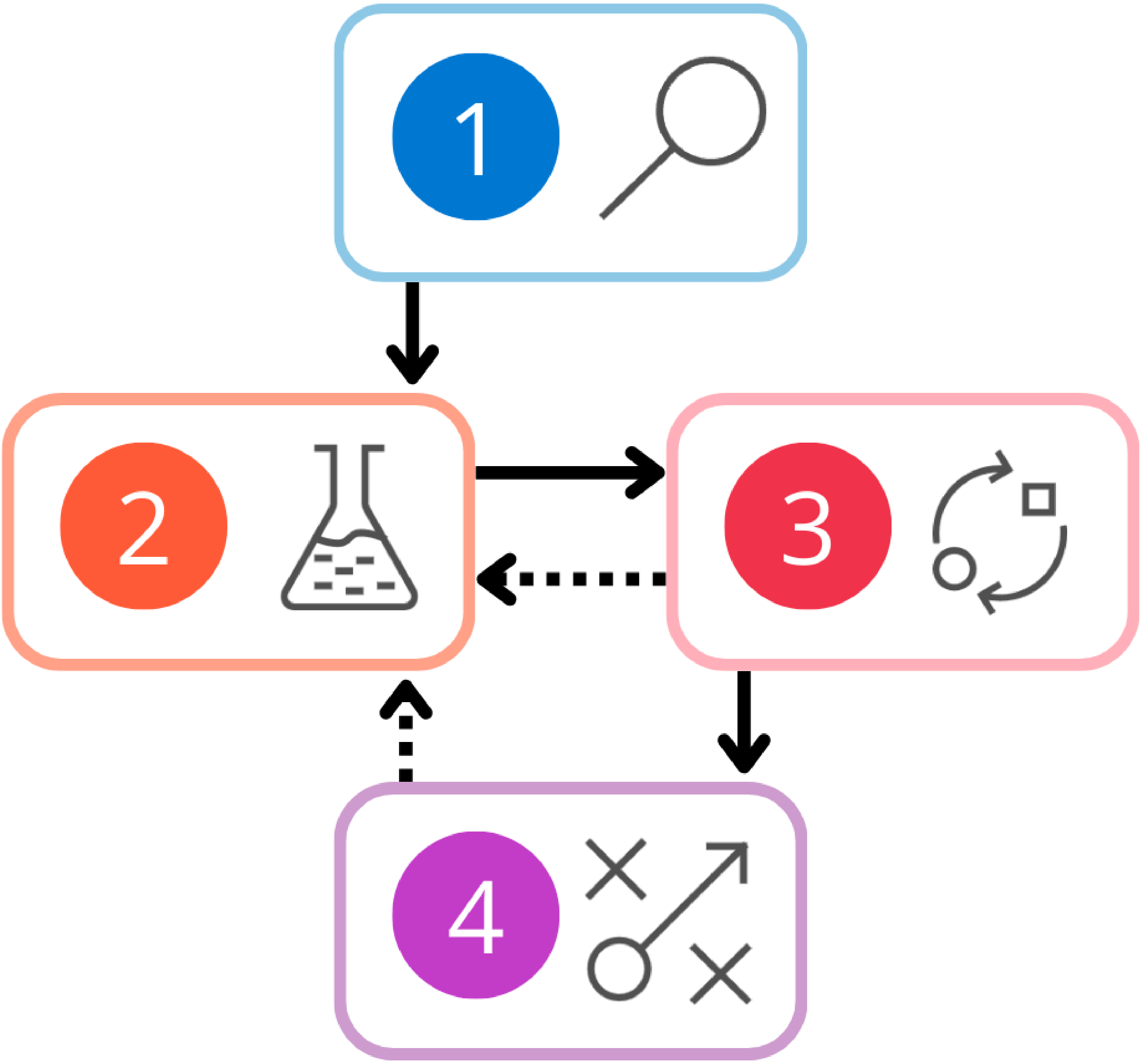
Understand the development lifecycle of a large language model (LLM) app

Completed

- 8 minutes

Before understanding how to work with prompt flow, let's explore the development lifecycle of a Large Language Model (LLM) application.

The lifecycle consists of the following stages:



1. **Initialization**: Define the use case and design the solution.
2. **Experimentation**: Develop a flow and test with a small dataset.
3. **Evaluation and refinement**: Assess the flow with a larger dataset.
4. **Production**: Deploy and monitor the flow and application.

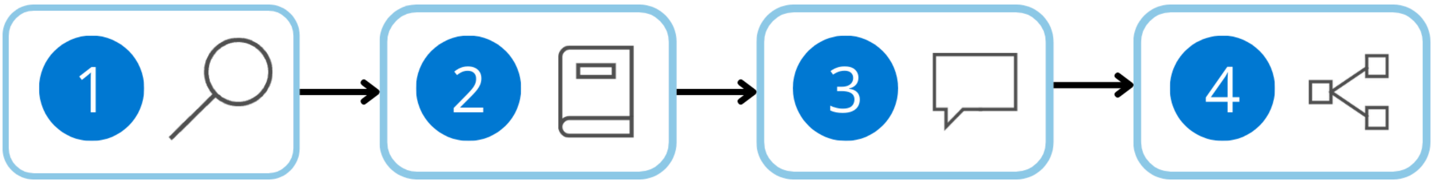
During both evaluation and refinement, and production, you might find that your solution needs to be improved. You can revert back to experimentation during which you develop your flow continuously, until you're satisfied with the results.

Let's explore each of these phases in more detail.

Initialization

Imagine you want to design and develop an LLM application to classify news articles. Before you start creating anything, you need to define what categories you want as output. You need to understand what a typical news article looks like, how you present the article as input to your application, and how the application generates the desired output.

In other words, during *initialization* you:



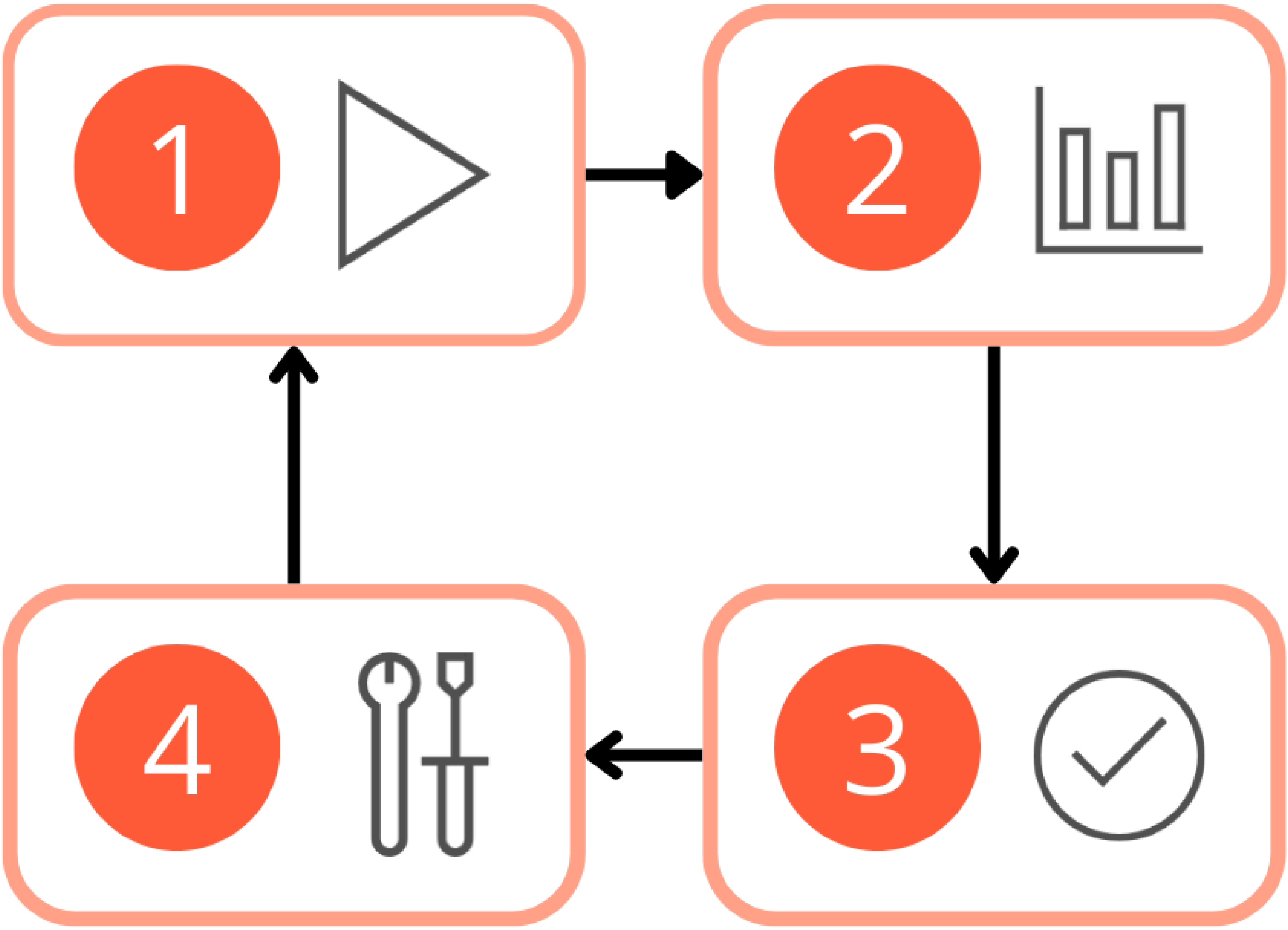
1. Define the **objective**
2. Collect a **sample dataset**
3. Build a **basic prompt**
4. Design the **flow**

To design, develop, and test an LLM application, you need a sample dataset that serves as the input. A sample dataset is a small representative subset of the data you eventually expect to parse as input to your LLM application.

When collecting or creating the sample dataset, you should ensure diversity in the data to cover various scenarios and edge cases. You should also remove any privacy sensitive information from the dataset to avoid any vulnerabilities.

Experimentation

You collected a sample dataset of news articles, and decided on which categories you want the articles to be classified into. You designed a flow that takes a news article as input, and uses an LLM to classify the article. To test whether your flow generates the expected output, you run it against your sample dataset.



The *experimentation* phase is an iterative process during which you (1) **run** the flow against a sample dataset. You then (2) **evaluate** the prompt's performance. If you're (3) satisfied with the result, you can **move on** to evaluation and refinement. If you think there's room for improvement, you can (4) **modify** the flow by changing the prompt or flow itself.

Evaluation and refinement

When you're satisfied with the output of the flow that classifies news articles, based on the sample dataset, you can assess the flow's performance against a larger dataset.

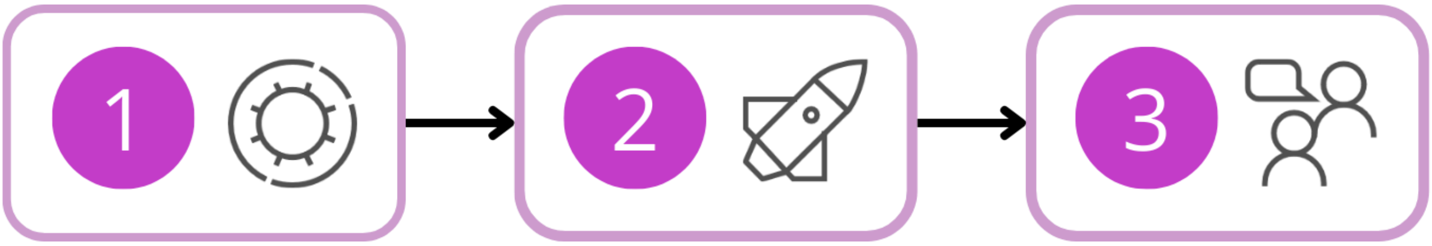
By testing the flow on a larger dataset, you can evaluate how well the LLM application generalizes to new data. During evaluation, you can identify potential bottlenecks or areas for optimization or refinement.

When you edit your flow, you should first run it against a smaller dataset before running it again against a larger dataset. Testing your flow with a smaller dataset allows you to more quickly respond to any issues.

Once your LLM application appears to be robust and reliable in handling various scenarios, you can decide to move the LLM application to production.

Production

Finally, your news article classification application is ready for *production*.

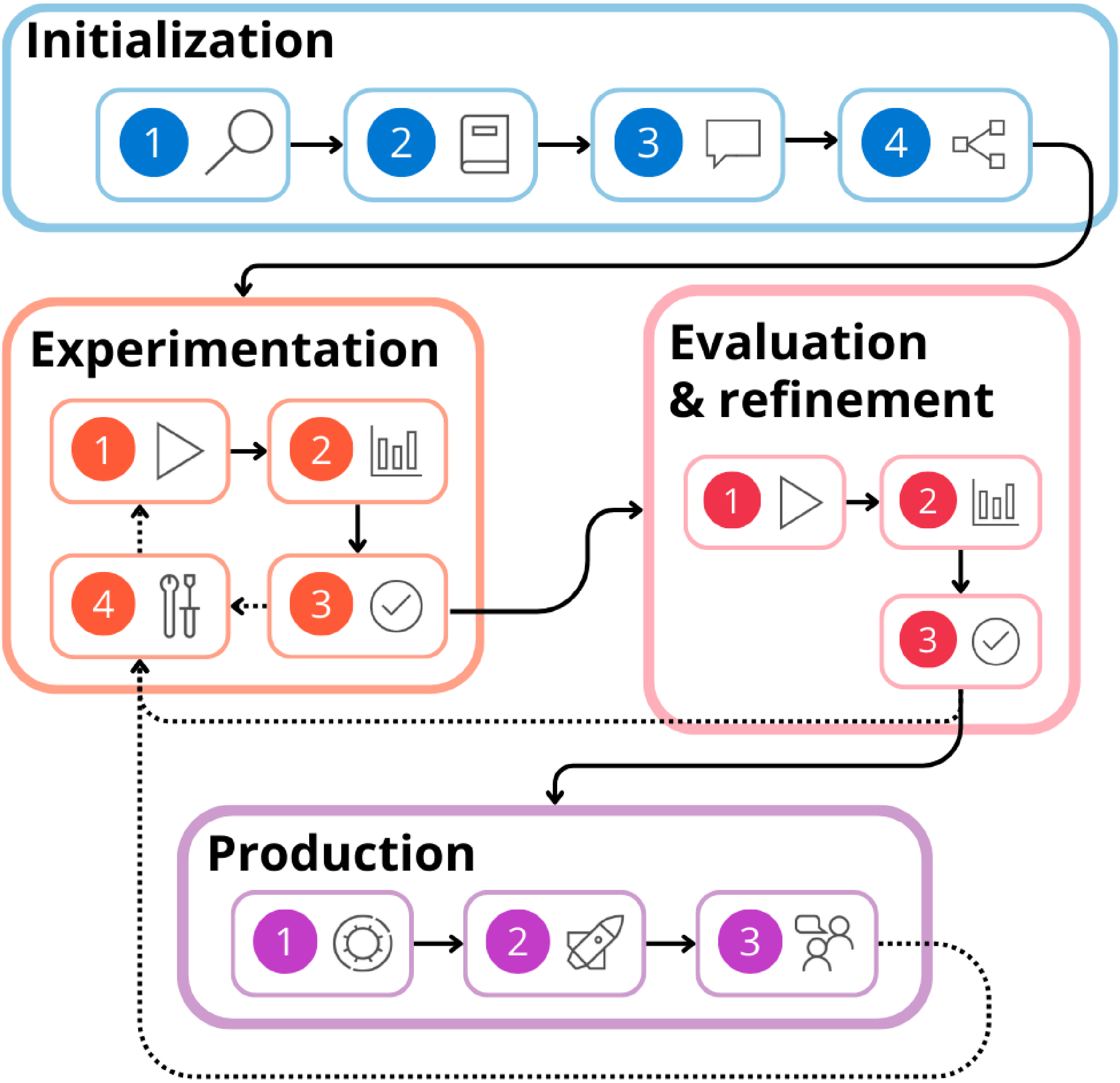


During production, you:

1. **Optimize** the flow that classifies incoming articles for efficiency and effectiveness.
2. **Deploy** your flow to an endpoint. When you call the endpoint, the flow is triggered to run and the desired output is generated.
3. **Monitor** the performance of your solution by collecting usage data and end-user feedback. By understanding how the application performs, you can improve the flow whenever necessary.

Explore the complete development lifecycle

Now that you understand each stage of the development lifecycle of an LLM application, you can explore the complete overview:



3. Understand core components and explore flow types

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/3-understand-flows

Understand core components and explore flow

types

Completed

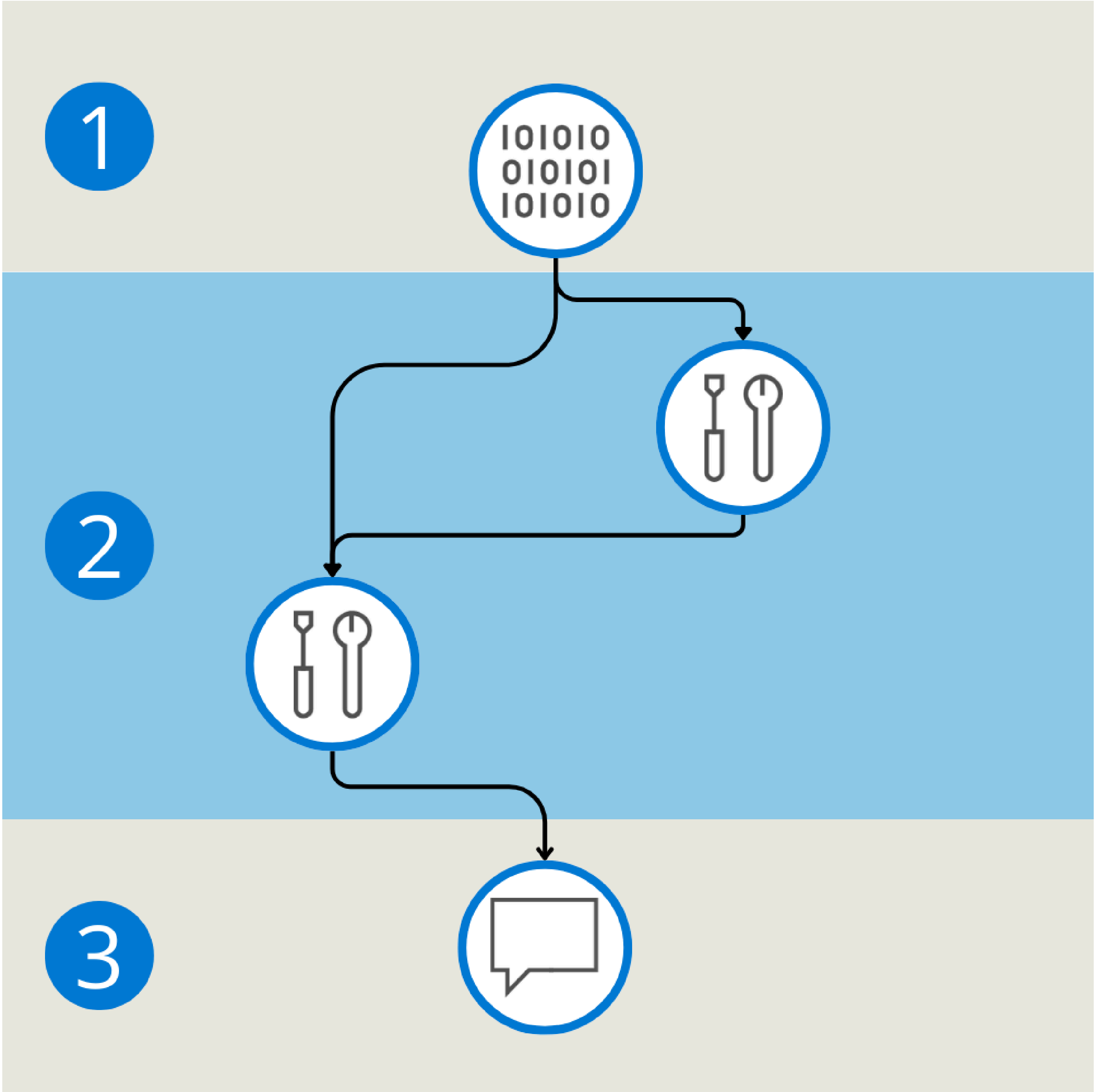
- 5 minutes

To create a Large Language Model (LLM) application with prompt flow, you need to understand prompt flow's core components.

Understand a flow

Prompt flow is a feature within Microsoft Foundry that allows you to author **flows**. Flows are executable workflows often consist of three parts:

1. **Inputs**: Represent data passed into the flow. Can be different data types like strings, integers, or boolean.
2. **Nodes**: Represent *tools* that perform data processing, task execution, or algorithmic operations.
3. **Outputs**: Represent the data produced by the flow.



Similar to a pipeline, a flow can consist of multiple nodes that can use the flow's inputs or any output generated by another node. You can add a node to a flow by choosing one of the available types of **tools**.

Explore the tools available in prompt flow

Three common tools are:

- **LLM tool**: Enables custom prompt creation utilizing Large Language Models.
- **Python tool**: Allows the execution of custom Python scripts.
- **Prompt tool**: Prepares prompts as strings for complex scenarios or integration with other tools.

Each tool is an executable unit with a specific function. You can use a tool to perform tasks like summarizing text, or making an API call. You can use multiple tools within one flow and use a tool multiple times.

> **Tip**
>
> If you're looking for functionality that is not offered by the available tools, you can create your own custom tool.

Whenever you add a new node to your flow, adding a new tool, you can define the expected inputs and outputs. A node can use one of the whole flow's inputs, or another node's output, effectively linking nodes together.

By defining the inputs, connecting nodes, and defining the desired outputs, you can create a flow. Flows help you create LLM applications for various purposes.

Understand the types of flows

There are three different types of flows you can create with prompt flow:

- **Standard flow**: Ideal for general LLM-based application development, offering a range of versatile tools.
- **Chat flow**: Designed for conversational applications, with enhanced support for chat-related functionalities.
- **Evaluation flow**: Focused on performance evaluation, allowing the analysis and improvement of models or applications through feedback on previous runs.

Now that you understand how a flow is structured and what you can use it for, let's explore how you can create a flow.

4. Explore connections and runtimes

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/4-connections-runtimes

Explore connections and runtimes

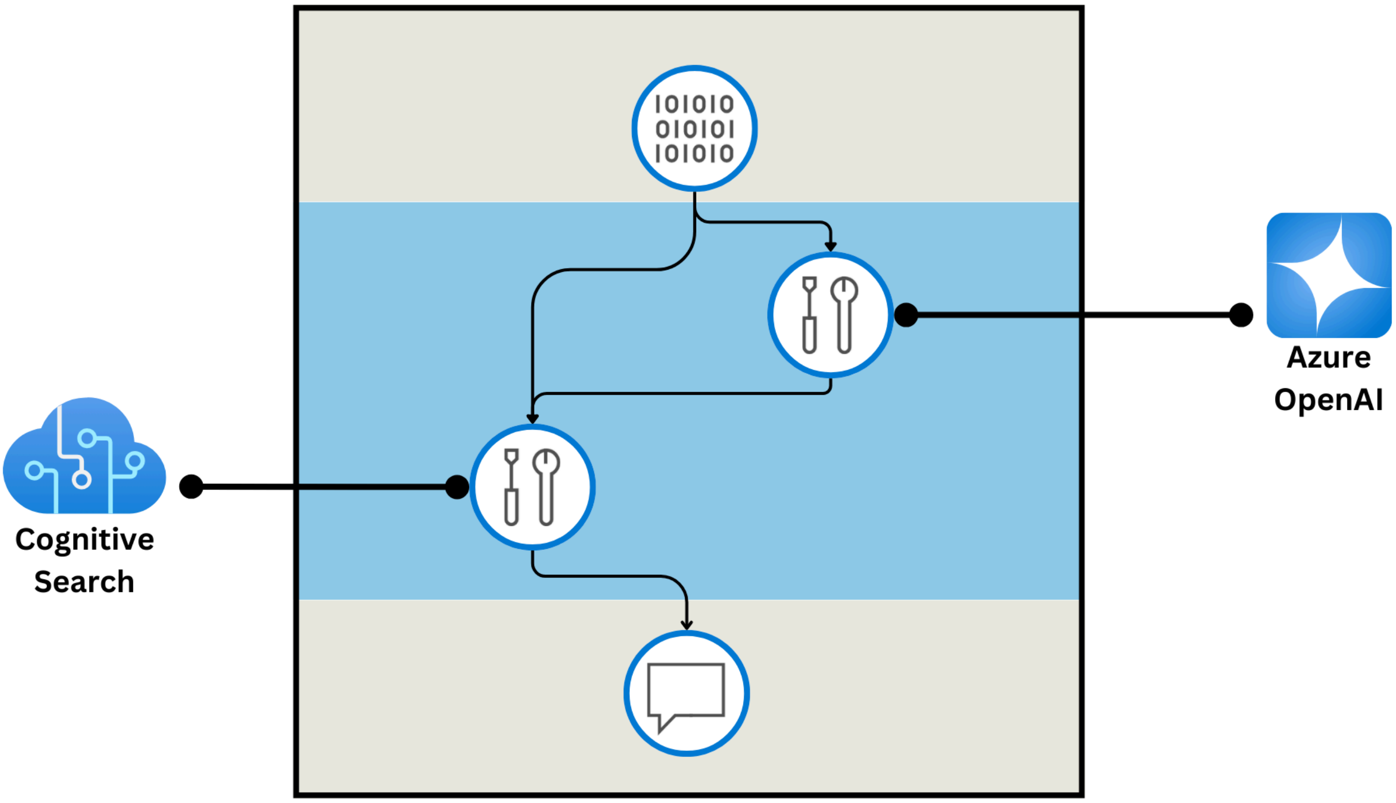
Completed

- 4 minutes

When you create a Large Language Model (LLM) application with prompt flow, you first need to configure any necessary **connections** and **runtimes**.

Explore connections

Whenever you want your flow to connect to external data source, service, or API, you need your flow to be authorized to communicate with that external service. When you create a **connection**, you configure a secure link between prompt flow and external services, ensuring seamless and safe data communication.



Depending on the type of connection you create, the connection securely stores the endpoint, API key, or credentials necessary for prompt flow to communicate with the external service. Any necessary secrets aren't exposed to users, but instead are stored in an Azure Key Vault.

By setting up connections, users can easily reuse external services necessary for tools in their flows.

Certain built-in tools require you to have a connection configured:

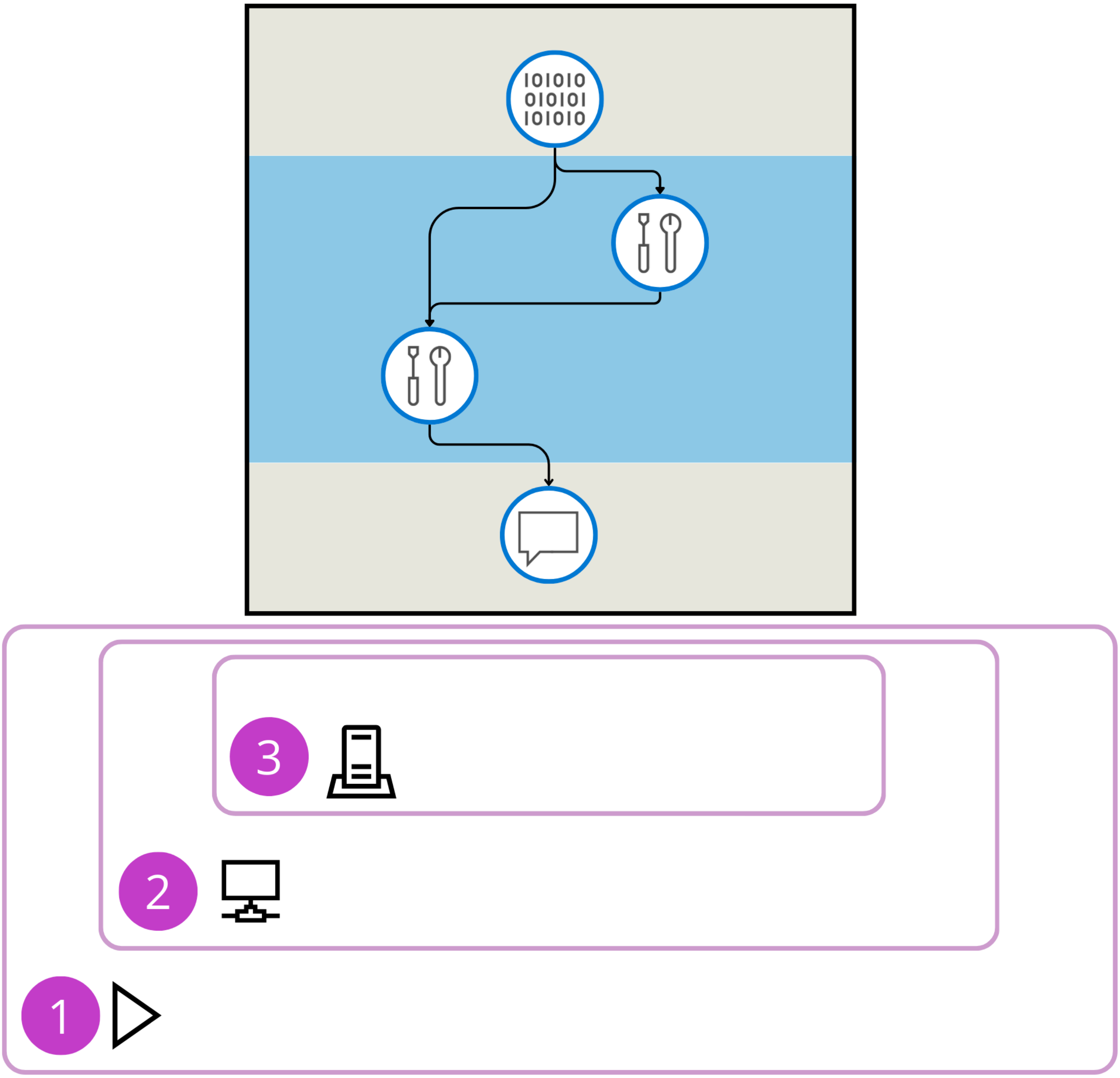
Connection type	Built-in tools
Azure OpenAI	LLM or Python
OpenAI	LLM or Python

Connection type	Built-in tools
Azure AI Search	Vector DB Lookup or Python
Serp	Serp API or Python
Custom	Python

Prompt flow connections play pivotal roles in two scenarios. They automate API credential management, simplifying and securing the handling of sensitive access information. Additionally, they enable secure data transfer from various sources, crucial for maintaining data integrity and privacy across different environments.

Explore runtimes

After creating your flow, and configuring the necessary connections your tools use, you want to run your flow. To run the flow, you need compute, which is offered through prompt flow **runtimes**.



Runtimes (1) are a combination of a **compute instance** (2) providing the necessary compute resources, and an **environment** (3) specifying the necessary packages and libraries that need to be installed before being able to run the flow.

When you use runtimes, you have a controlled environment where flows can be run and validated, ensuring that everything works as intended in a stable setting. A default environment is available for quick development and testing. When you require other packages to be installed, you can create a custom environment.

5. Explore variants and monitoring options

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/5-variants-monitor

Explore variants and monitoring options

Completed

- 6 minutes

During production, you want to optimize and deploy your flow. Finally, you want to monitor your flows to understand when improving your flows is necessary.

You can optimize your flow by using **variants**, you can deploy your flow to an **endpoint**, and you can monitor your flow by evaluating key metrics.

Explore variants

Prompt flow **variants** are versions of a tool node with distinct settings. Currently, variants are only supported in the LLM tool, where a variant can represent a different prompt content or connection setting. Variants allow users to customize their approach for specific tasks, like, summarizing news articles.

Some benefits of using variants are:

- **Enhance the quality of your LLM generation**: Creating diverse variants of an LLM node helps find the best prompt and settings for high-quality content.
- **Save time and effort**: Variants allow for easy management and comparison of different prompt versions, streamlining historical tracking and reducing the effort in prompt tuning.
- **Boost productivity**: They simplify the optimization of LLM nodes, enabling quicker creation and management of variations, leading to better results in less time.
- **Facilitate easy comparison**: Variants enable side-by-side result comparisons, aiding in choosing the most effective variant based on data-driven decisions.

Deploy your flow to an endpoint

When you're satisfied with the performance of your flow, you can choose to deploy it to an **online endpoint**. Endpoints are URLs that you can call from any application. When you make an API call to an online endpoint, you can expect (almost) immediate response.

When you deploy your flow to an online endpoint, prompt flow generates a URL and key so you can safely integrate your flow with other applications or business processes. When you invoke the endpoint, a flow is run and the output is returned in real-time. As a result, deploying flows to endpoints can for example generate chat or agentic responses that you want to return in another application.

Monitor evaluation metrics

In prompt flow, monitoring evaluation metrics is key to understanding your LLM application's performance, ensuring they meet real-world expectations and deliver accurate results.

To understand whether your application is meeting practical needs, you can collect end-user feedback and assess the application's usefulness. Another approach to understanding whether your application is performing well, is by comparing LLM predictions with expected or *ground truth* responses to gauge accuracy and relevance. Evaluating the LLM's predictions is crucial for keeping LLM applications reliable and effective.

Metrics

The key metrics used for monitoring evaluation in prompt flow each offer unique insight into the performance of LLMs:

- ***Groundedness***: Measures alignment of the LLM application's output with the input source or database.
- ***Relevance***: Assesses how pertinent the LLM application's output is to the given input.
- ***Coherence***: Evaluates the logical flow and readability of the LLM application's text.
- ***Fluency***: Assesses the grammatical and linguistic accuracy of the LLM application's output.
- ***Similarity***: Quantifies the contextual and semantic match between the LLM application's output and the ground truth.

Metrics like *groundedness*, *relevance*, *coherence*, *fluency*, and *similarity* are key for quality assurance, ensuring that interactions with your LLM applications are accurate and effective. Whenever your LLM application doesn't perform as expected, you need to revert back to experimentation to iteratively explore how to improve your flow.

6. Exercise - Get started with prompt flow

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/6-exercise

Exercise - Get started with prompt flow

Completed

- 15 minutes

Now, it's your chance to explore how to develop LLM apps with prompt flow.

In this exercise, you create a standard flow with prompt flow in the Microsoft Foundry portal.

> **Note**
>
> To complete this lab, you need an Azure subscription in which you have administrative access.

Launch the exercise and follow the instructions.

Launch Exercise

7. Module assessment

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/7-knowledge-check

Module assessment

Completed

- 3 minutes

8. Summary

https://learn.microsoft.com/en-us/training/modules/get-started-prompt-flow-ai-studio/8-summary

Summary

Completed

- 1 minute

In this module, you learned:

- The development lifecycle when creating LLM applications.
- What a flow is in prompt flow.
- The core components when working with prompt flow.

Learn more

- Prompt flow in Microsoft Foundry portal
- Prompt engineering techniques
- Microsoft Foundry Discord
- Microsoft Foundry Developer Forum