Microsoft Azure

AgentHack: Evaluating Agents

April Speight
Principal Cloud Advocate, AI

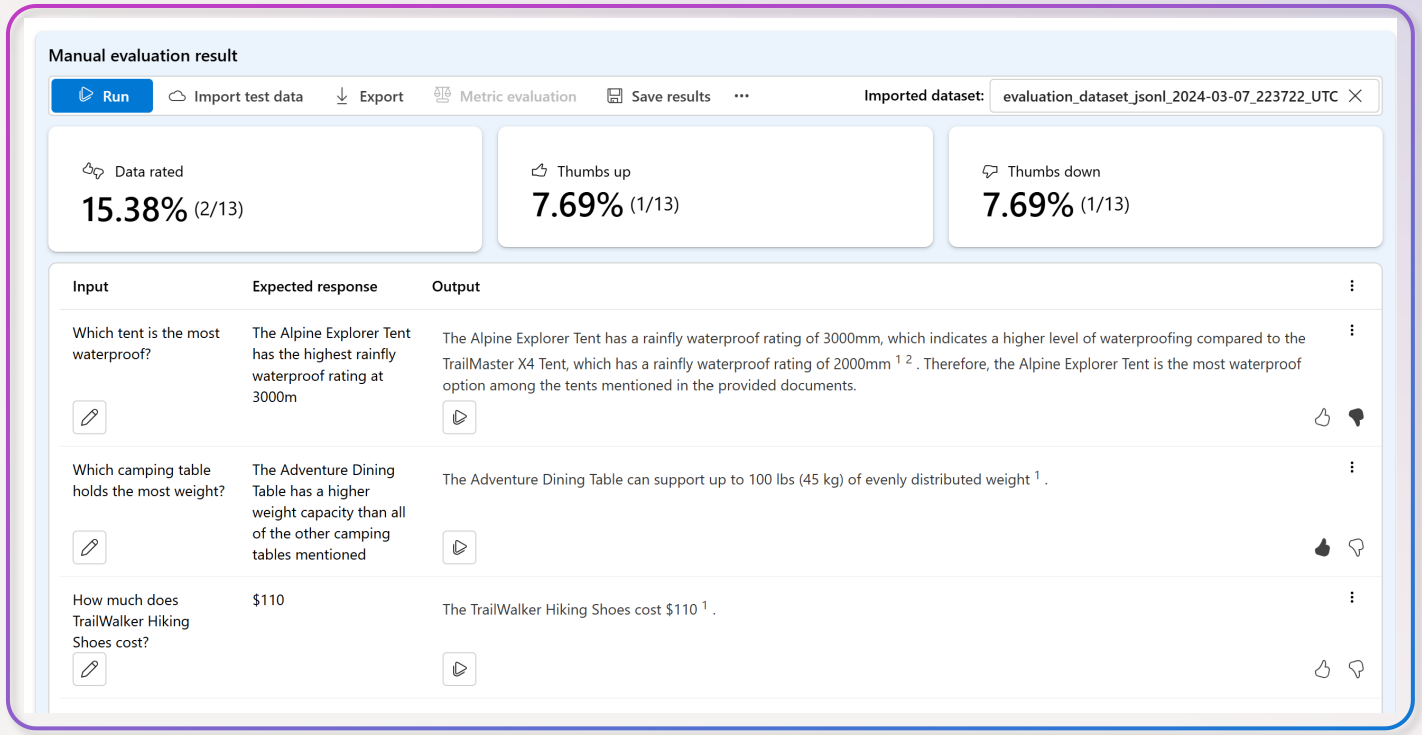
How do you typically **test** if a **model** or **app** is working as intended?

Manual evaluation

Evaluation

Enable human graders to **spot-check output performance** for a specific risk on a small dataset.

- Supports a test dataset in CSV or JSONL format, or manual data entry via the UI
- Provide thumbs-up or thumbs-down ratings
- Modify system messages, models, or parameters as needed
- Save evaluation results to share with team members or for future reference
- Note: Limited to 50 input rows



Built-in evaluators in Azure AI Foundry

Quality (AI-assisted)

- Groundedness
- Groundedness Pro
- Retrieval
- Relevance
- Coherence
- Fluency
- Similarity

Quality (NLP)

- F1 Score
- ROUGE
- GLEU
- BLEU
- METEOR

Risk & Safety (AI-assisted)

- Violence
- Sexual
- Self Harm
- Hate/Unfairness
- Indirect Attack
- Protected Material
- Ungrounded Attributes
- Code Vulnerability

Custom

- Customize built-in evaluators **or** build your own evaluators **or** synthetic data similar with Azure AI Evaluation SDK

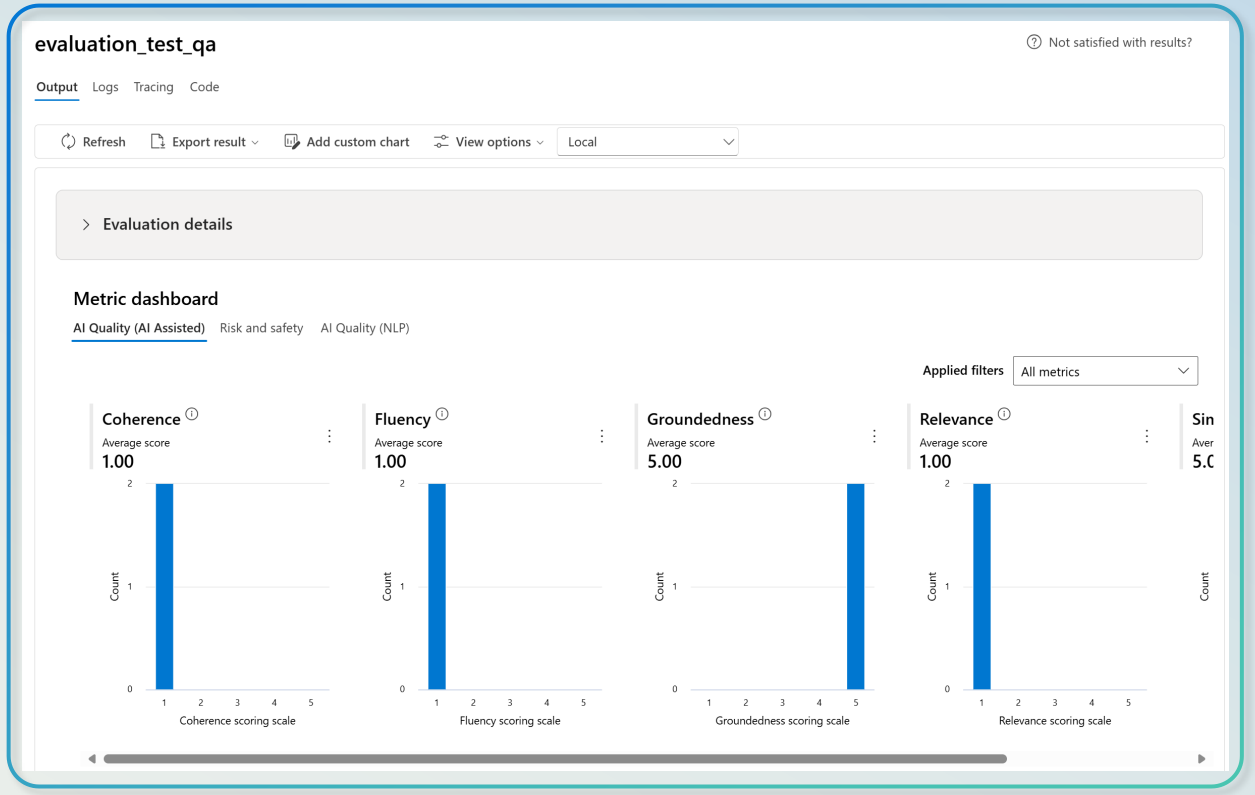
Automated evaluation

Put AI to work to **measure output performance at scale** across a broader range of risks.

- Supports a test dataset in CSV or JSONL format, or manual data entry via the UI
- Support multiple evaluation targets:
  - Model and prompt
  - Dataset
  - Prompt flow
- View results via a dashboard in the portal
- Offload resources and run evaluations in the cloud with Cloud evaluation

Try the Sample
aka.ms/aistudio/eval-samples

Evaluation



Demo

Evaluate images and multi-modal text and images

For evaluators that support conversations for image and multi-modal image and text, you can pass in image URLs or base64 encoded images into the conversation.

Try the Sample
aka.ms/aistudio/eval-samples

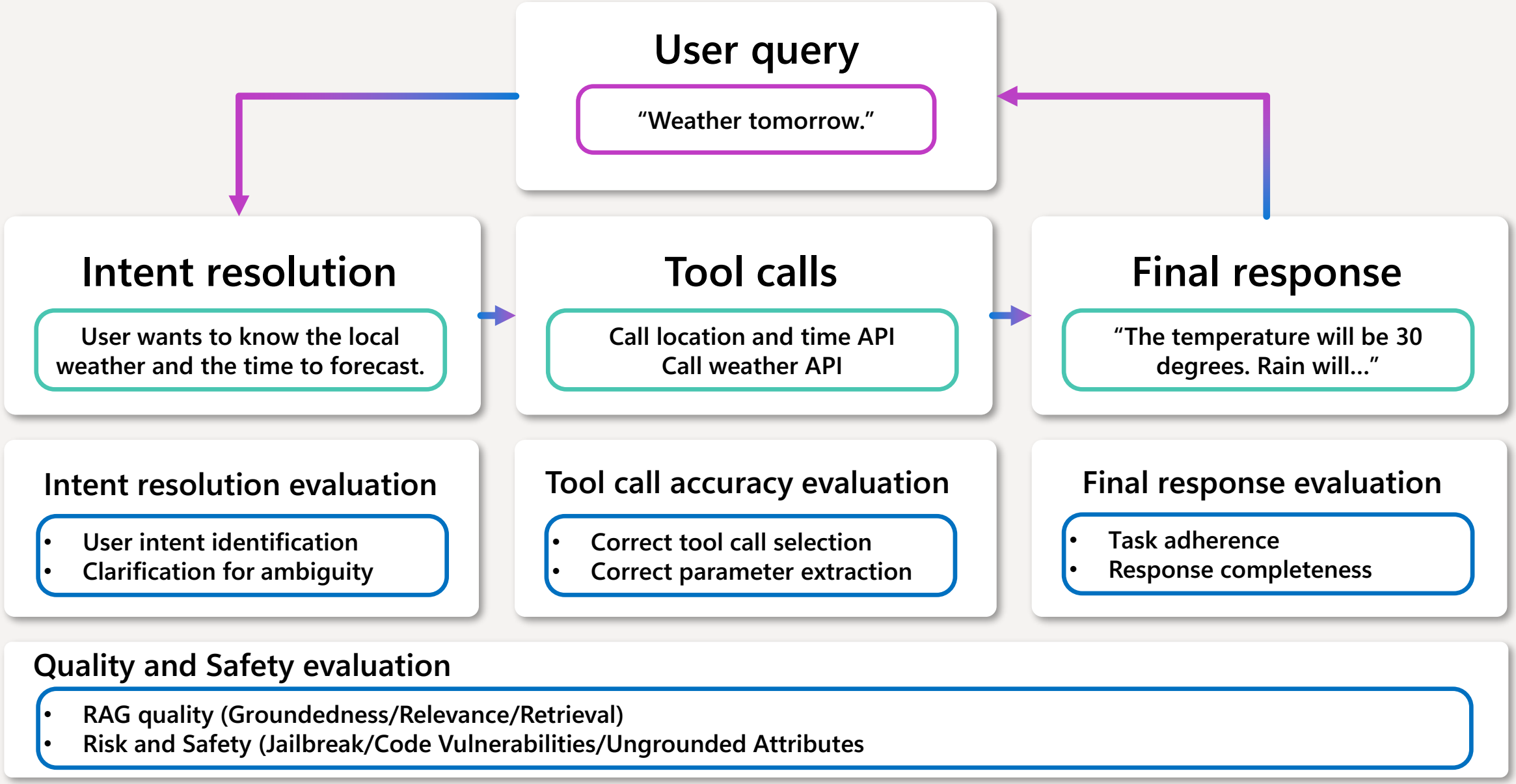
Supported Scenarios

- Multiple images with text input to image or text generation
- Text only input to image generations
- Image only inputs to text generation

Demo

How do you typically **test** if an **agent** is working as intended?

Evaluate agentic workflows



Intent resolution

Agent Evaluation

Measures how well the agent identifies the user's request, including how well it scopes the user's intent, asks clarifying questions, and reminds end users of its scope of capabilities

Try the Sample
aka.ms/intentresolution-sample

```python
import os
from azure.ai.evaluation import AzureOpenAIModelConfiguration
from azure.identity import DefaultAzureCredential
from azure.ai.evaluation import IntentResolutionEvaluator
from pprint import pprint

model_config = AzureOpenAIModelConfiguration(
    azure_endpoint=os.environ["AZURE_OPENAI_ENDPOINT"],
    api_key=os.environ["AZURE_OPENAI_API_KEY"],
    api_version=os.environ["AZURE_OPENAI_API_VERSION"],
    azure_deployment=os.environ["MODEL_DEPLOYMENT_NAME"],
)

intent_resolution_evaluator = IntentResolutionEvaluator(model_config)

result = intent_resolution_evaluator(query="What are the opening hours of the Eiffel Tower?",
                                     response="Opening hours of the Eiffel Tower are 9:00 AM to 11:00
PM.",

                     )
pprint(result)
```

Demo

Tool call accuracy

Agent Evaluation

Evaluates the agent's ability to select the appropriate tools, and process correct parameters from previous steps.

Try the Sample
aka.ms/toolcallaccuracy-sample

```python
import os
from azure.ai.evaluation import ToolCallAccuracyEvaluator, AzureOpenAIModelConfiguration
from pprint import pprint

model_config = AzureOpenAIModelConfiguration(
    azure_endpoint=os.environ["AZURE_OPENAI_ENDPOINT"],
    api_key=os.environ["AZURE_OPENAI_API_KEY"],
    api_version=os.environ["AZURE_OPENAI_API_VERSION"],
    azure_deployment=os.environ["MODEL_DEPLOYMENT_NAME"],
)


tool_call_accuracy = ToolCallAccuracyEvaluator(model_config=model_config)

query = "How is the weather in Seattle ?"
tool_call = {
    "type": "tool_call",
    "tool_call_id": "call_CUdbkBfvVBla2YP3p24uhElJ",
    "name": "fetch_weather",
    "arguments": {"location": "Seattle"},
}

tool_definition = {
    "id": "fetch_weather",
    "name": "fetch_weather",
    "description": "Fetches the weather information for the specified location.",
    "parameters": {
        "type": "object",
        "properties": {"location": {"type": "string", "description": "The location to fetch weather
for."}},
    },
}

response = tool_call_accuracy(query=query, tool_calls=tool_call, tool_definitions=tool_definition)
pprint(response)
```

Demo

Task adherence

Agent Evaluation

Measures how well the agent's final response adheres to its assigned tasks, according to its system message and prior steps.

Try the Sample
aka.ms/taskadherence-sample

```python
import os
from azure.ai.evaluation import TaskAdherenceEvaluator, AzureOpenAIModelConfiguration
from pprint import pprint

model_config = AzureOpenAIModelConfiguration(
    azure_endpoint=os.environ["AZURE_OPENAI_ENDPOINT"],
    api_key=os.environ["AZURE_OPENAI_API_KEY"],
    api_version=os.environ["AZURE_OPENAI_API_VERSION"],
    azure_deployment=os.environ["MODEL_DEPLOYMENT_NAME"],
)
task_adherence_evaluator = TaskAdherenceEvaluator(model_config)

result = task_adherence_evaluator(
    query="What are the best practices for maintaining a healthy rose garden during the summer?",
    response="For optimal summer care of your rose garden, start by watering deeply early in the
morning to ensure the roots are well-hydrated without encouraging fungal growth. Apply a 2-3 inch layer
of organic mulch around the base of the plants to conserve moisture and regulate soil temperature.
Fertilize with a balanced rose fertilizer every 4 to 6 weeks to support healthy growth. Prune away any
dead or diseased wood to promote good air circulation, and inspect regularly for pests such as aphids
or spider mites, treating them promptly with an appropriate organic insecticidal soap. Finally, ensure
that your roses receive at least 6 hours of direct sunlight daily for robust flowering.",
)
pprint(result)
```

Demo

Response completeness

Assesses the quality of an agent response by examining how well it aligns with the provided ground truth. The evaluation is based on the following scoring system

Try the Sample
aka.ms/rescompleteness-sample

```python
from azure.ai.evaluation import ResponseCompletenessEvaluator, AzureOpenAIModelConfiguration
from pprint import pprint
import os

model_config = AzureOpenAIModelConfiguration(
    azure_endpoint=os.environ["AZURE_OPENAI_ENDPOINT"],
    api_key=os.environ["AZURE_OPENAI_API_KEY"],
    api_version=os.environ["AZURE_OPENAI_API_VERSION"],
    azure_deployment=os.environ["MODEL_DEPLOYMENT_NAME"],
)

from azure.ai.evaluation import AzureOpenAIModelConfiguration

response_completeness_evaluator = ResponseCompletenessEvaluator(model_config=model_config)

result = response_completeness_evaluator(
    response="Itinery: Day 1 check out the downtown district of the city on train; for Day 2, we can rest in hotel.",
    ground_truth="Itinery: Day 1 take a train to visit the downtown area for city sightseeing; Day 2 rests in hotel.",
)
result
```

Demo

Evaluate AI agents End-to-End

Create Agent

- Deploy a **GPT** model supporting JSON mode
- Create an agent with **Azure AI Agent Service**
- Create **Thread**

Conversation

- Create the **Message**
- Execute
- List the messages

Evaluate

- Get **data** from the agent
- Set up the **evaluators**
- Run the evaluators

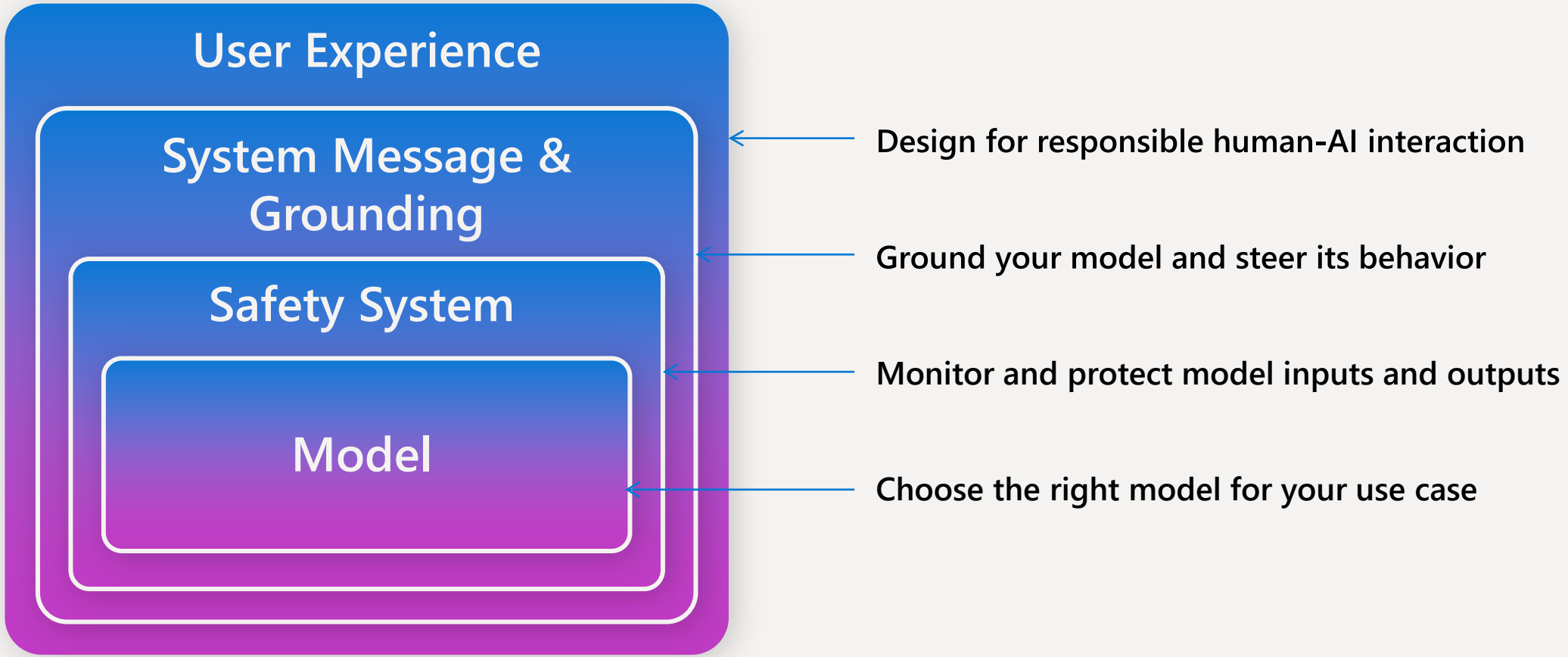
View Results

- Inspect results on Azure AI Foundry (portal)

Demo

What should I do
after I run **evaluations**?

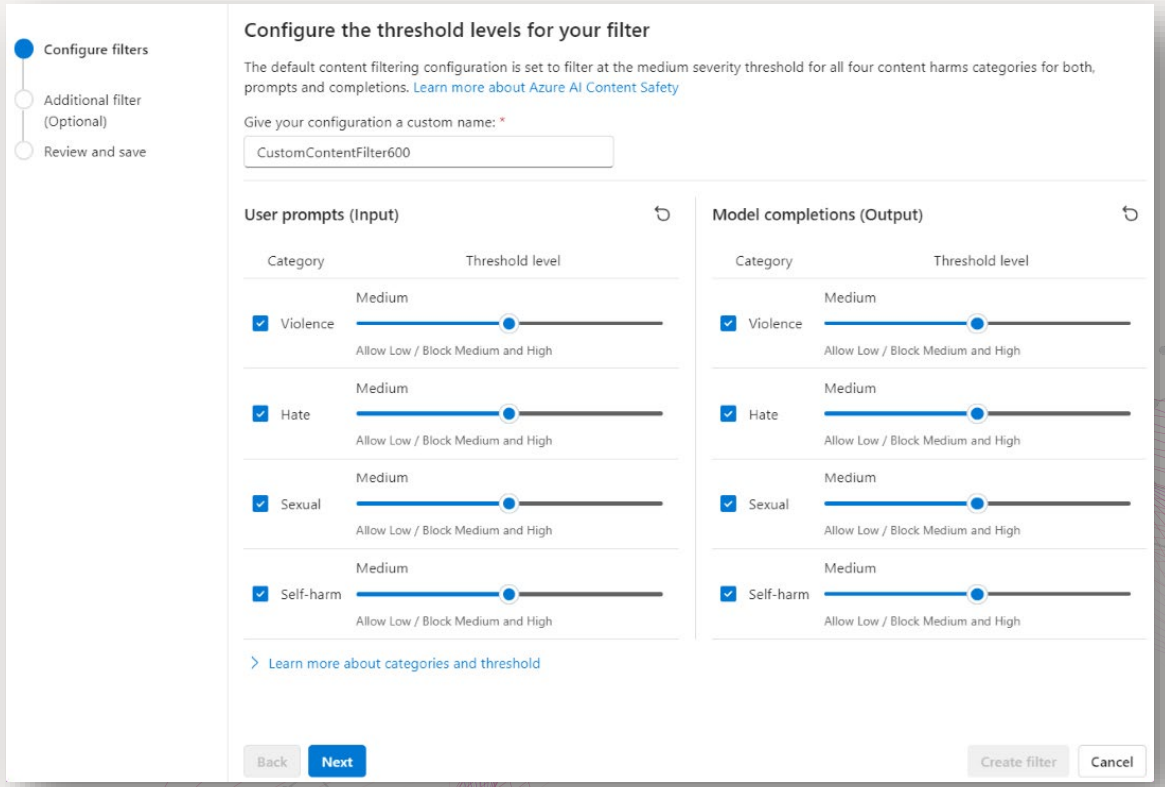
Sjtl !n jujhbujpo!rbzf st



Implement a built-in safety system

Safeguard your app with Azure AI Content Safety

- Detect and **mitigate problematic text and images**
- Configurable for **inputs and outputs**
- **Call as an API** or **deploy it from** Azure AI Foundry and Azure Machine Learning to apply to any model in the Azure AI model catalog
- Available as a **built-in safety system for models deployed as a service** (e.g. Llama 3, Mistral)
- Available at no cost as a built-in safety system for Azure OpenAI Service



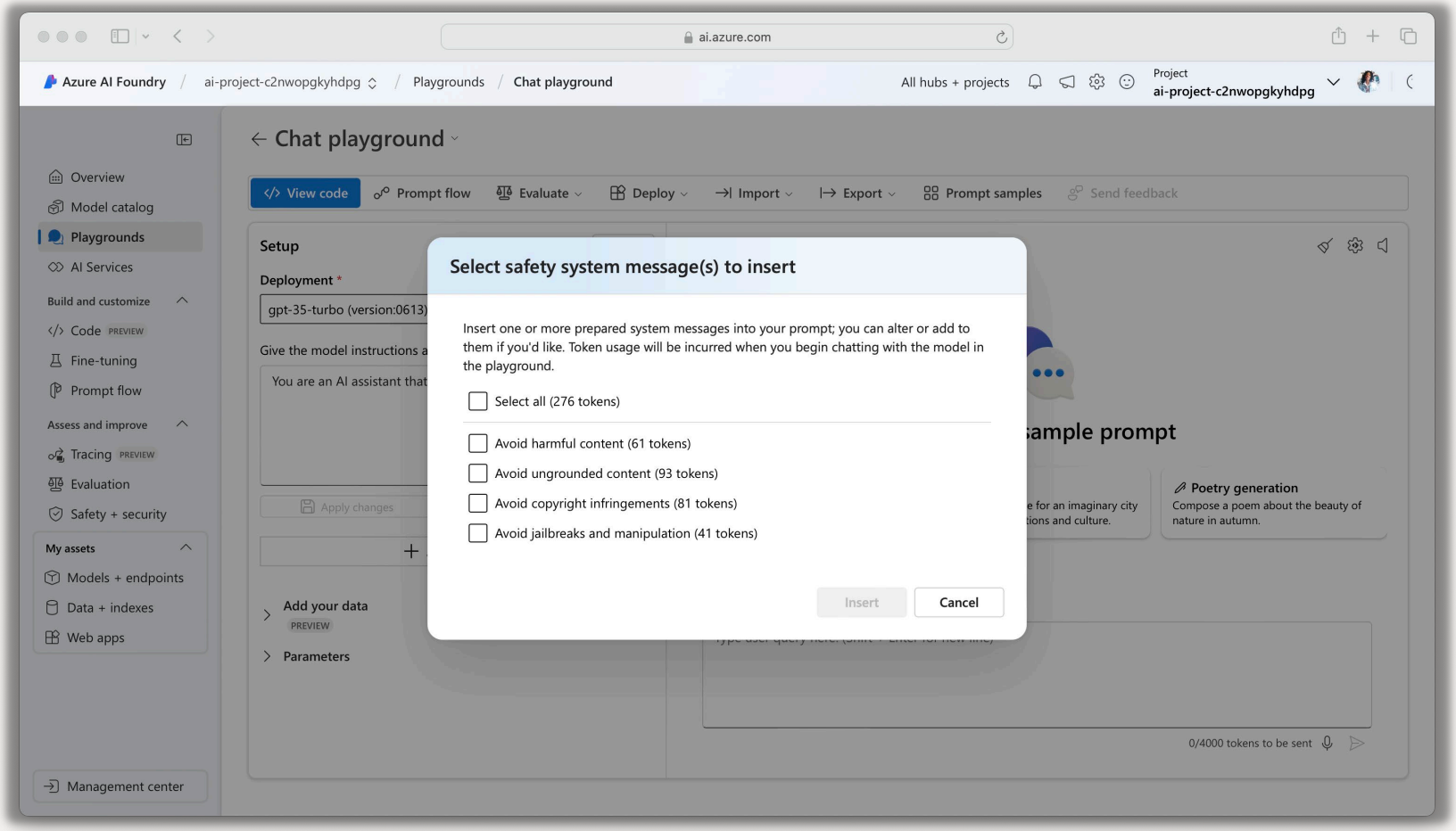
Learn more: aka.ms/ContentSafety

Steer your model's behavior with a system message

Define the model's profile, capabilities, and limitations for your scenario	• **Define the specific task(s)** you would like the model to complete. Describe who the users of the model will be, what inputs will be provided to the model, and what you expect the model to output • **Define how the model should complete the tasks**, including any additional tools (like APIs, code, plug-ins) the model can use. • **Define the scope and limitations** of the model's performance by providing clear instructions • **Define the posture and tone** the model should exhibit in its responses.
Define the model's output format	• **Define the language and syntax** of the output format. For example, if you want the output to be machine parseable, you may want to structure the output to be in JSON, XSON or XML. • **Define any styling or formatting** preferences for better user readability like bulleting or bolding certain parts of the response
Provide example(s) to demonstrate the intended behavior of the model	• **Describe difficult use cases** where the prompt is ambiguous or complicated, to give the model additional visibility into how to approach such cases. • **Show chain-of-thought** reasoning to better inform the model on the steps it should take to achieve the desired outcomes.
Define additional behavioral and safety guardrails	• **Define specific guardrails to mitigate harms** that have been identified and prioritized for the scenario

Safety system message templates

Get started with research-backed templates, available in Azure AI Foundry and Azure OpenAI Service playgrounds



Monitoring generative AI applications

Tracing

Capture and store telemetry data for analysis

Online Evaluation

Continuously evaluate trace data remotely as it's collected

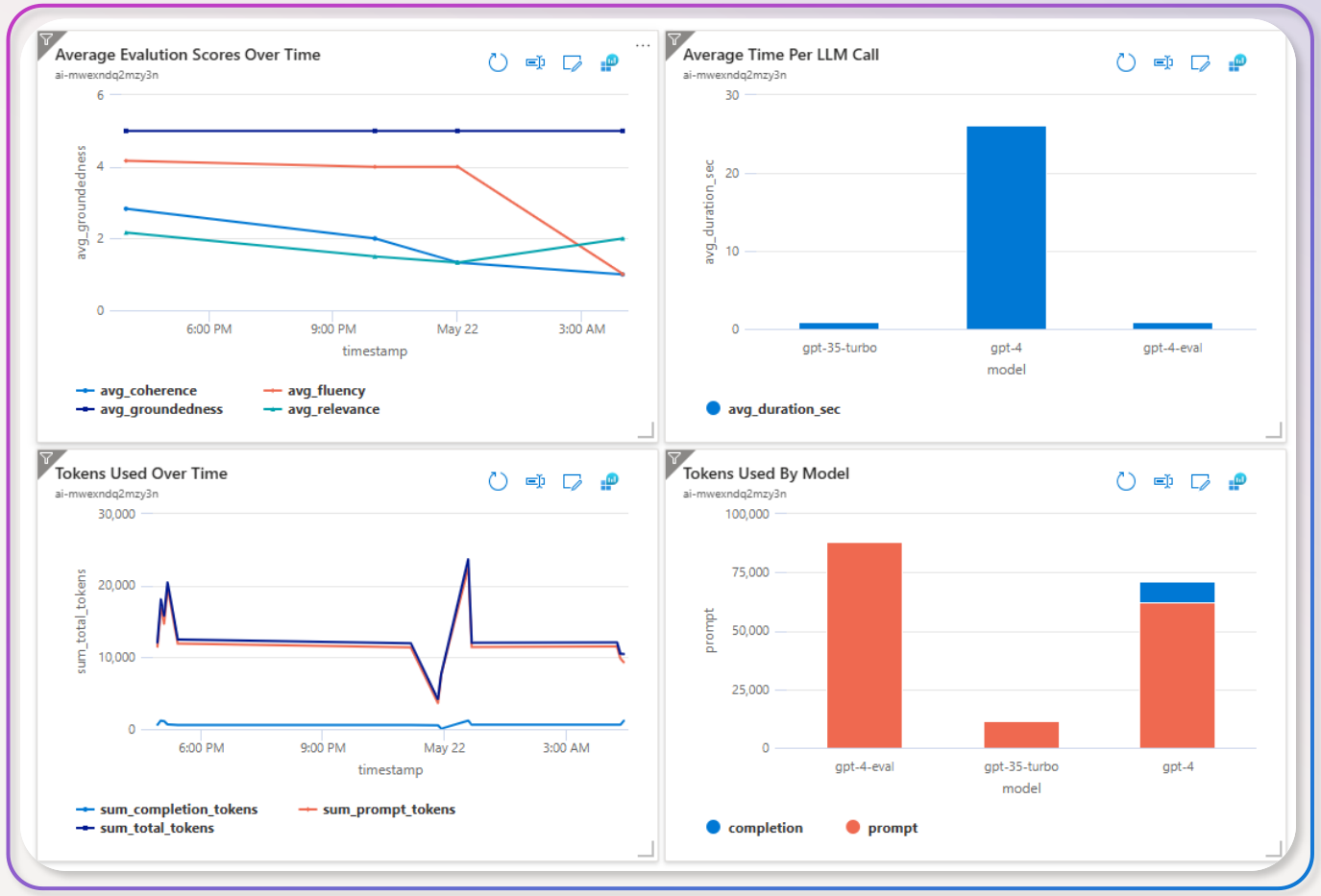
Monitor with Azure Monitor App Insights

Monitor token consumption, evaluation metrics, plus more in a customized dashboard

Observability

Monitor data in production with App Insights in Azure Monitor

- Effortless production data logging
- Enable monitoring for operational (error rate, latency) and token (token usage) metrics for GenAI deployments
- View extensive monitoring results for your prompt flow deployment within a comprehensive UI
- Configure alerts for violations based on organizational targets and run monitoring on a recurring basis



Get started with Azure AI Evaluation today

Get started with the Azure AI Foundry:

ai.azure.com

Learn more about Evaluating Agents on Microsoft Learn:

https://aka.ms/AgentEvaluators

Access the sample Notebooks:

- **Intent resolution**
  aka.ms/intentresolution-sample
- **Tool call accuracy**
  aka.ms/toolcallaccuracy-sample
- **Task adherence**
  aka.ms/taskadherence-sample
- **Response completeness**
  aka.ms/rescompleteness-sample
- **End-to-end Azure AI agent evaluation**
  aka.ms/e2e-agent-eval-sample