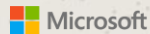




Building an AI Agent with Prompty and Azure AI Foundry.

Bethany Jepchumba | [@bethanyjep](#)



In partnership with **Reactor**



Agenda



Agentic AI

AI Agents

Unlike existing automation, AI Agents can automate requests as soon as they come in by...



Reasoning over the request



Proactively retrieving more context



Performing required actions with APIs



What are agents?

AI designed to perform a task

Tasks can vary in level of complexity and capabilities depending on your need

Simple



Generation

Generate summaries, images, audio, and more with an AI model and inputs.



Retrieval

Retrieve information from grounding data, reason, summarize, and answer user questions



Action

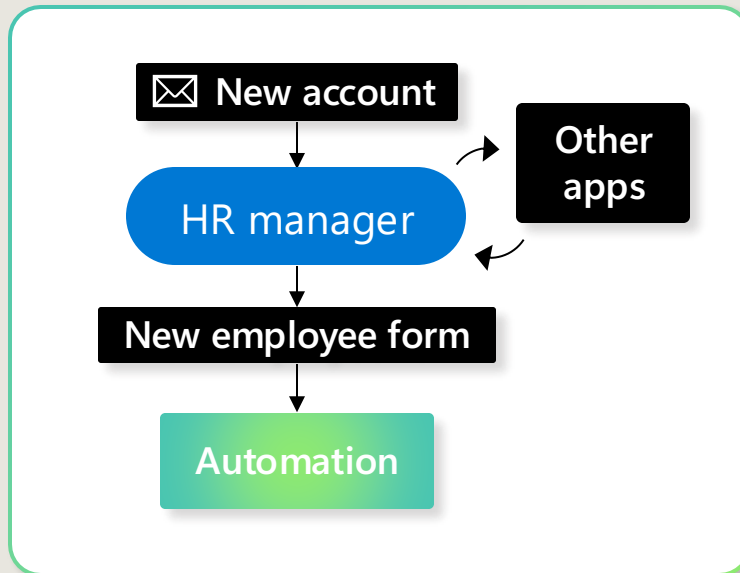
Take actions to automate workflows, and replace repetitive tasks for users

Advanced

With AI agents, you can make your processes faster than ever

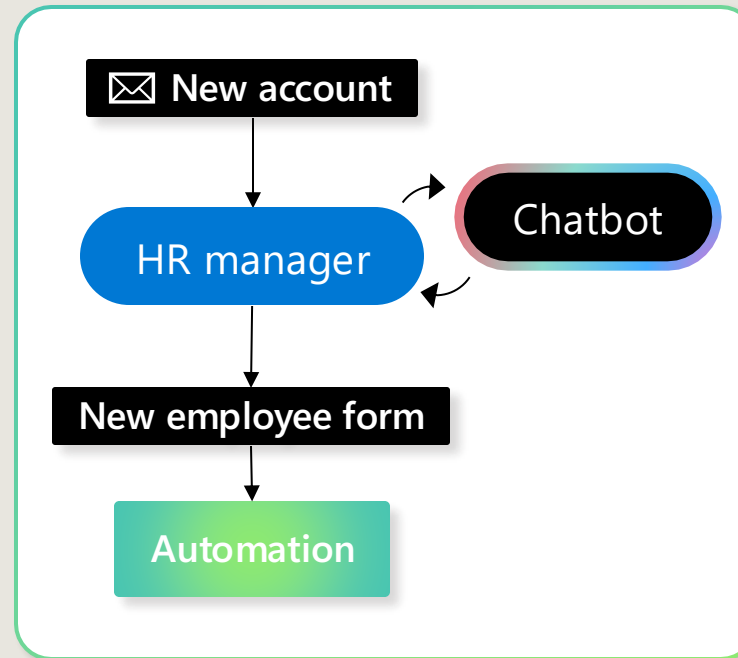
Yesterday – manual

Forms over data



Today – RAG

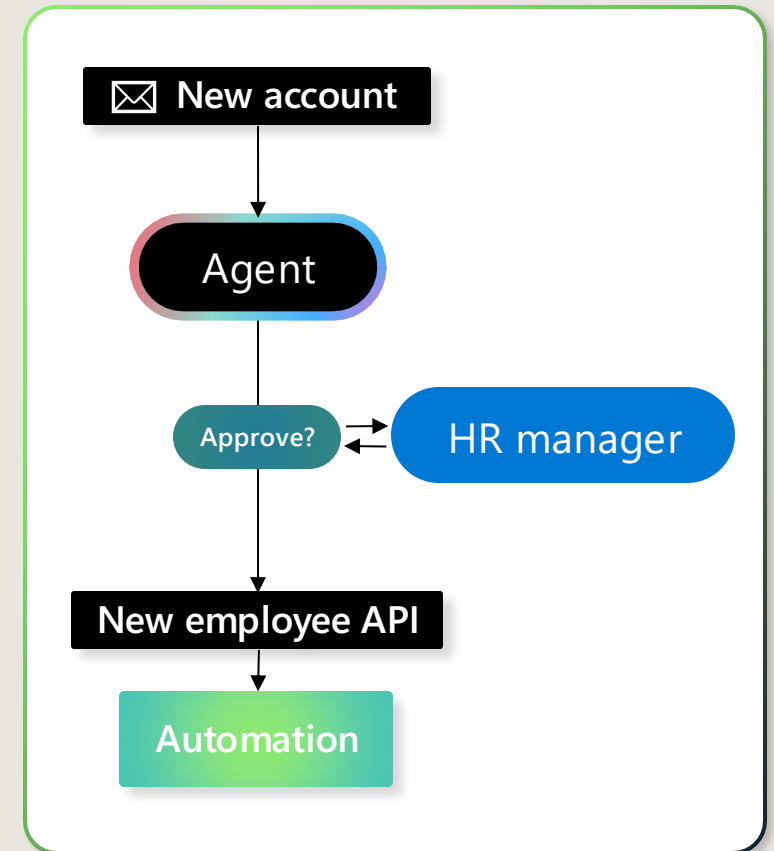
Chatbots



Improve employee productivity

Tomorrow

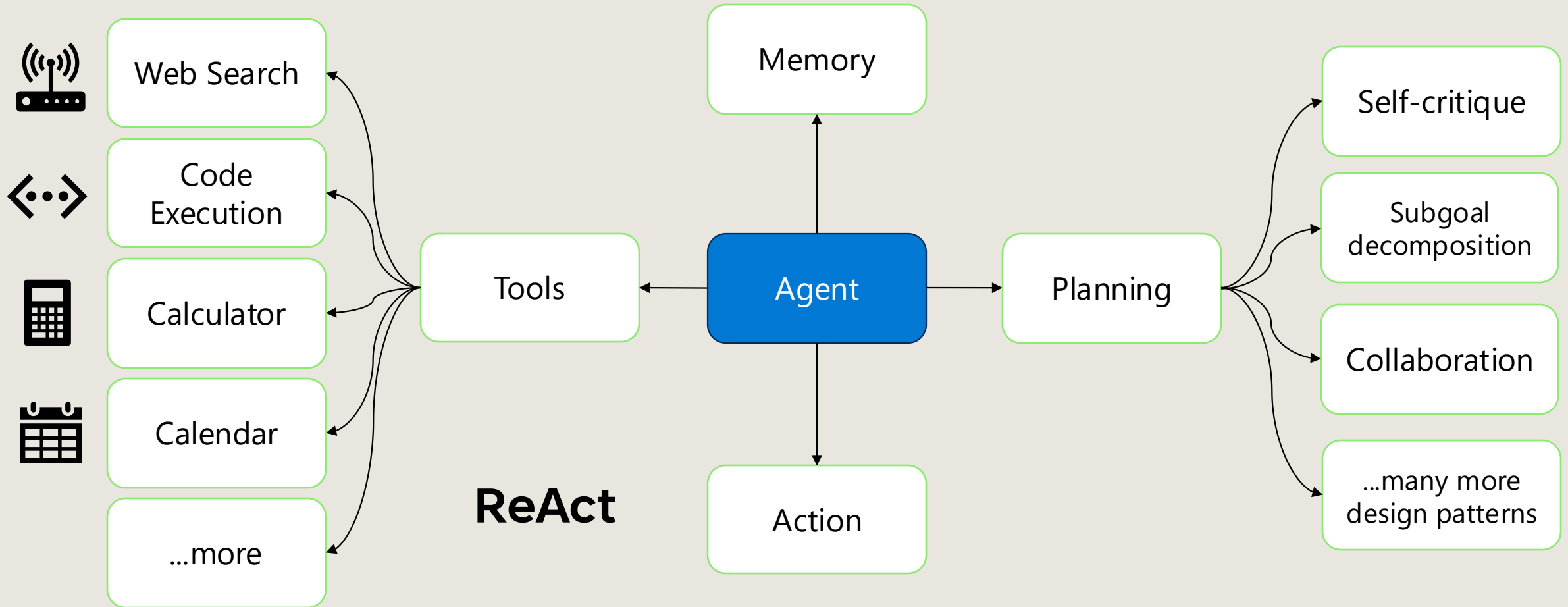
Automated agents



Improve process efficiency

VALUE

Agentic AI capabilities



Azure AI Foundry



Copilot Studio



Visual Studio



GitHub



Azure AI
Foundry SDK



Model Catalog

Foundational
models

Open-source
models

Task
models

Industry models



Azure OpenAI
Service



Azure
AI Search



Azure AI
Agent Service



Azure AI
Content
Safety

Evaluations

Customization

Governance

Monitoring

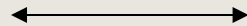
Observability

How does your agent work?

"Help me book a trip to New York for a client meeting? I need to fly out next Monday and return on Friday."



User



Travel Booking Agent

"I've booked your trip to New York as requested. Here are details:..."



Knowledge Sources

(search, files, databases, storage etc.)



Models

(Azure OpenAI Service, Models-as-a-Service)



Actions

(Pre-built or custom tools to automate processes)

How does your agent work?

- Step 1:**
Create an Agent
- Step 2:**
Create a Thread
- Step 3:**
Run the Agent
- Step 5:**
Check the Run status
- Step 6:**
Display the Agent's Response



Agent
Travel Planning Agent

Instructions
You are a travel booking and expense management assistant designed to help employees plan, book, and manage business travel.

Model



Your data (optional)

-  Azure AI Search
-  Files (local or Azure Blob)

Tools (optional)

- File Search
- Code Interpreter
- Function Calling
- Bing Search
- Microsoft SharePoint (coming soon)
- Microsoft Fabric (coming soon)
- Azure Logic Apps (coming soon)
- Azure Functions
- OpenAPI 3.0 specified tools

Thread
Travel Planning

User's message
I need to book a hotel in New York for 2 stays.

Agent's message
Here are some suggestions:

User's message
What's the daily meal allowance for the business trip?

Agent's message
The daily allowance for your business trip is \$75, as per company policy.

Run 1

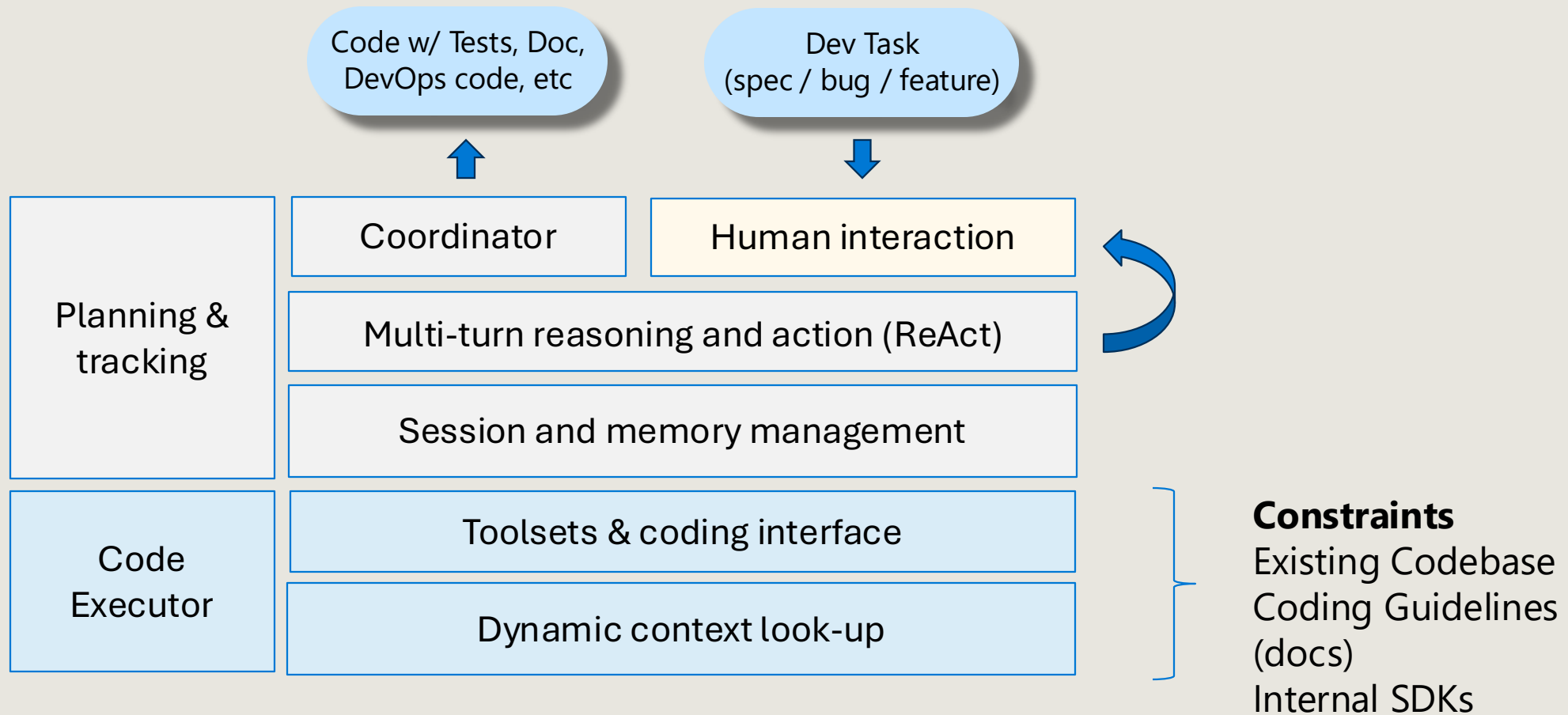
- 1 Use Tripadvisor API to search the nearest hotel
- 2 Create message

Run 2

- 1 Use Microsoft SharePoint to query the company travel policy
- 2 Create message

Code Generation Agent

Generates code based on natural language requirements, leveraging existing code base, templates, guidelines, libraries to match policies and best practices while interacting with humans to clarify, validate and deliver functionality as intended.



Create a code generator Agent



Scenario

An open-source maintainer for Prompty looking for a way to simplify tagging issues.



Introduction to Prompty



What is Prompty?

Specification:

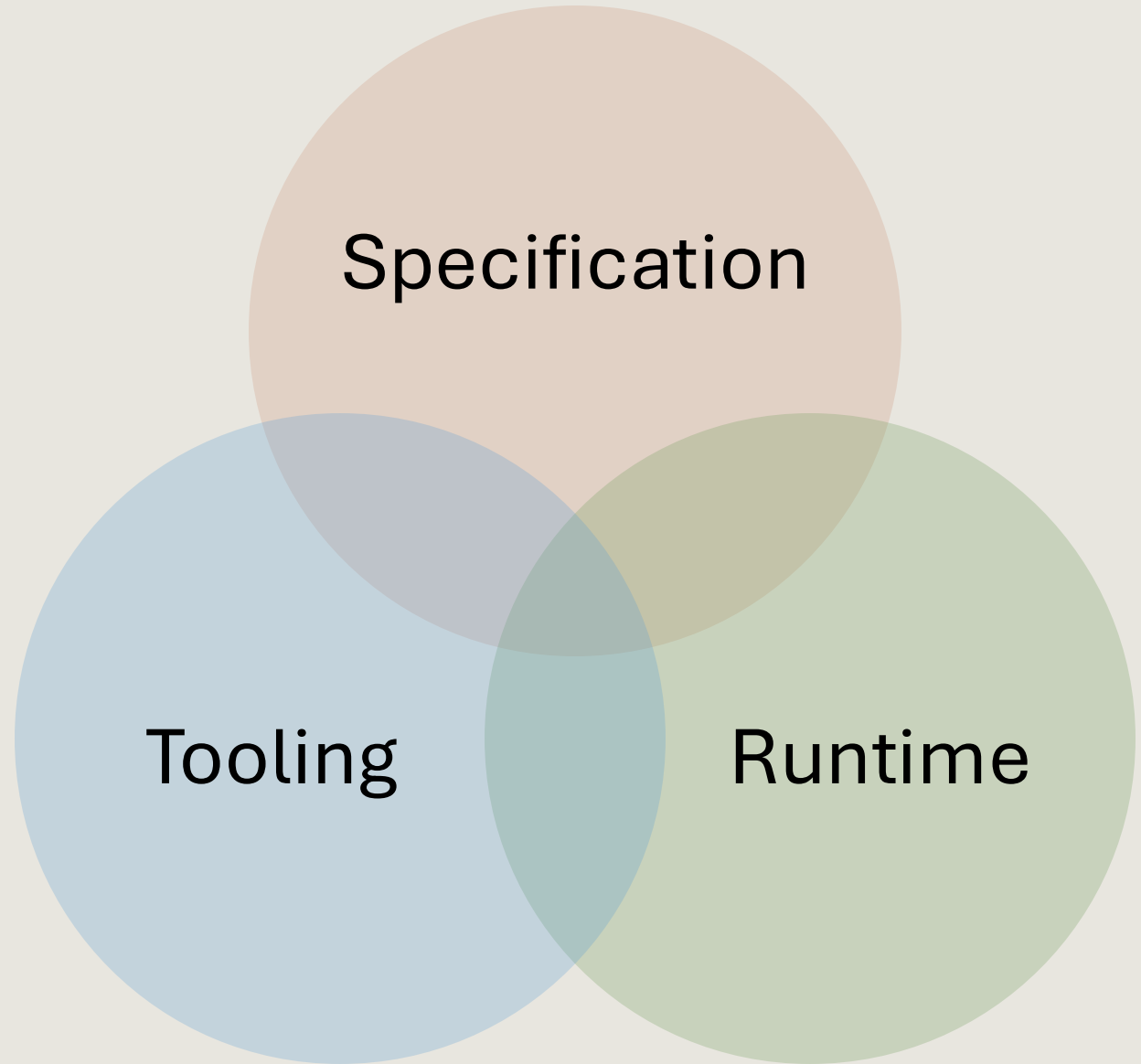
A standard format that unifies the prompt and how it is executed into a single asset.

Tooling:

Developer friendly ways to interact with the asset, create new assets, and manage supporting code.

Runtime:

A simple way to transition this new asset into code.



Prompt as a language-agnostic prompt asset

Specification

Prompt is intended to be a language agnostic asset class for creating prompts and managing the responses.

The goal is to simplify your workflow by creating a standard that can be used by any language, any framework, and any tool to create a prompt and manage the response.

Prompt can be imported to/exported from Azure AI Studio for smooth transition between local and cloud. Prompt assets can be also shared across organizations.

```
1  ---
2  name: Basic Prompt
3  description: A basic prompt that uses the GPT-3 chat API to answer questions
4  authors:
5    - sethjuarez
6    - jietong
7  model:
8    api: chat
9    configuration:
10      azure_deployment: gpt-35-turbo
11  sample:
12    firstName: Jane
13    lastName: Doe
14    question: What is the meaning of life?
15  ---
16  system:
17    You are an AI assistant who helps people find information.
18    As the assistant, you answer questions briefly, succinctly,
19    and in a personable manner using markdown and even add some
20    personal flair with appropriate emojis.
21
22  # Customer
23    You are helping {{firstName}} {{lastName}} to find answers to their questions.
24    Use their name to address them in your responses.
25
26  user:
27    {{question}}
```

Prompt Tooling in Visual Studio Code

Tooling

Visual Studio Code Extension

- Frontmatter Autocompletion
- Colorization / Syntax Highlighting
- Validation
- Quick Run
- Code Generation
- Evaluations – Future
- ASK US

```
20 sample:
21   firstName: Seth
22   context: >
23     The Alpine Explorer Tent boasts a detachable divider for privacy,
24     numerous mesh windows and adjustable vents for ventilation, and
25     a waterproof design. It even has a built-in gear loft for storing
26     your outdoor essentials. In short, it's a blend of privacy, comfort,
27     and convenience, making it your second home in the heart of nature!
28   question: What can you tell me about your tents?
29   ---
30   system:
31     You are an AI assistant who helps people find information. As the assistant,
32     you answer questions briefly, succinctly, and in a personable manner using
33     markdown and even add some personal flair with appropriate emojis.
34
35   🌟 Customer
36   You are helping {{firstName}} to find answers to their questions.
37   Use their name to address them in your responses.
38
39   # Context
40   Use the follow contex to provide a more personalized response to {{firstName}}:
41   {{context}}
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE PROMPT FLOW COMMENTS Prompt Ou

2024-05-18 16:34:12.750 [info] Calling <https://vizier.openai.azure.com/openai/deployments/gpt-4o/chat/completions?api-version=2024-02-01>

2024-05-18 16:34:14.778 [info] Hey Seth! Our Alpine Explorer Tent is pretty awesome 🥰. It has all you need for a camping adventure: a detachable divider for privacy, plenty of mesh windows and vents for a breezy experience, a waterproof design to keep you dry. Plus, there's a built-in gear loft to store your essentials. It's truly like a cozy home outdoors! 🏕️🌟

Prompt Runtime Library

Runtime

Prompt runtime is the whatever engine that understands and can execute the format.

- **Azure AI Foundry**
- **Python** – LangChain
- **C#** – Semantic Kernel
- **TypeScript/JavaScript** – Extension
- **Your Language** – Implement your own!

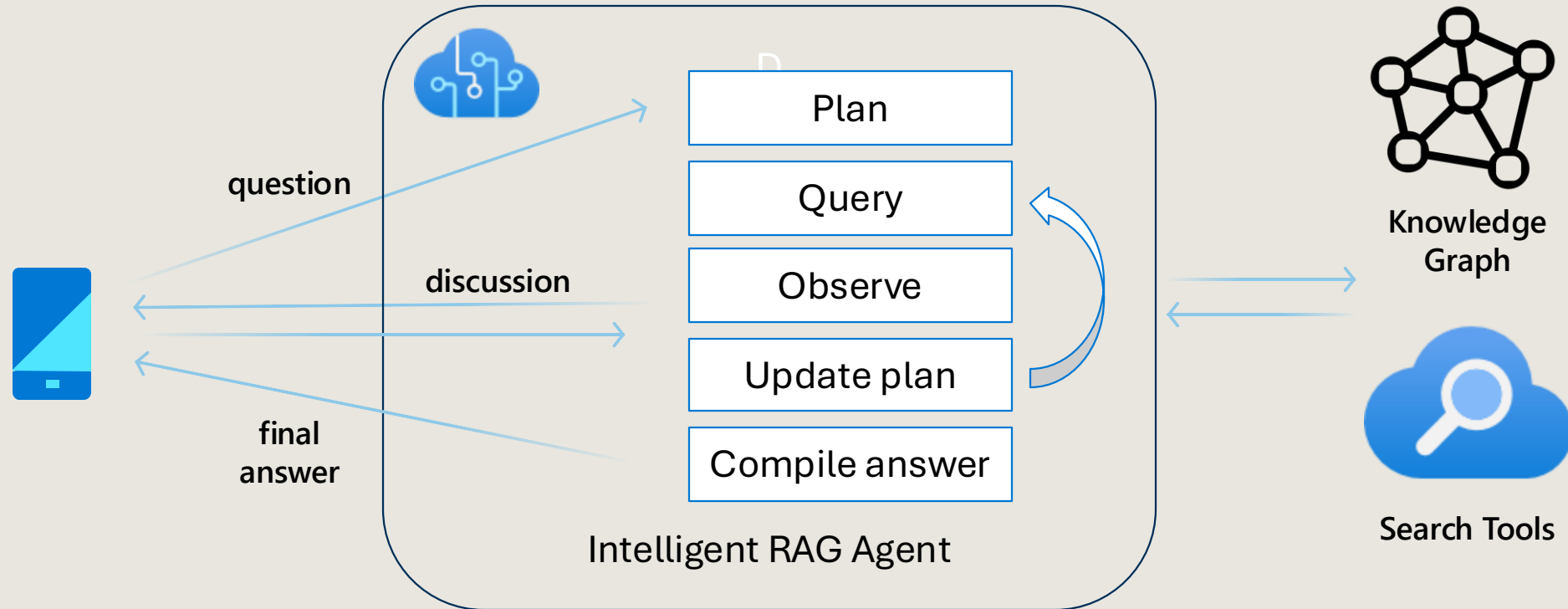
```
9
10 @trace
11 def write(researchContext, research, productContext, products, assignment
12     configuration = AzureOpenAIModelConfiguration(
13         azure_deployment="gpt-4o",
14         api_version="2024-02-01",
15         azure_endpoint=os.getenv("AZURE_OPENAI_ENDPOINT_40"),
16         api_key=os.getenv("AZURE_OPENAI_KEY_40"),
17     )
18 | flow = Flow.load(
19     base / "writer.prompt",
20     model={"configuration": configuration, "parameters": {"max_tokens
21 )
22 result = flow(
23     researchContext=researchContext,
24     research=research,
25     productContext=productContext,
26     products=products,
27     assignment=assignment,
28 )
29 return {"article": result.replace("`markdown\n", "").replace("\n`"
30
```

Create your first Prompt



Retrieval Augmented Generation Agent

Translates questions into a research problem with human in the loop to produce high quality answers to complex questions within the scope of its domain



Implement RAG in Prompt



Tracing and debugging with Prompty

benefits

- ✓ Provides a detailed view of the execution flow of the application
- ✓ Support both local and cloud runs
- ✓ Detail every LLM call including message, prompt, parameters, token, and latency
- ✓ Collect & attach user feedback using OpenTelemetry's semantic conventions

Use tracing to view performance and debug your app PREVIEW

Check out your [Insights for Generative AI applications dashboard](#)

View query Manage data source Refresh 11/01/2024 - 11/08/2024 Last day 7D 1M

Search Filter Columns

Created on	Name	Input ↓	Output	Input tokens	Output tokens
Nov 08, 2024 01:09 AM	> chat gpt-4o	what are some of the coolest hiking t...			
Nov 08, 2024 01:09 AM	> chat gpt-4o-mini	what are some of the coolest hiking t...	Hey Amii! 🌟 Ready to hit the trails? ...	80	248
Nov 08, 2024 01:10 PM	▼ chat_with_products				
Nov 08, 2024 01:10 PM	> get_product_documents				
Nov 08, 2024 01:10 PM	> chat gpt-4o-mini	Tell me about hiking sandals that are...	The TrekStar Hiking Sandals from Tre...	1184	115
Nov 08, 2024 02:09 PM	> chat_with_products				
Nov 07, 2024 04:52 PM	▼ chat_with_products				
Nov 07, 2024 04:52 PM	> get_product_documents				
Nov 07, 2024 04:52 PM	> chat gpt-4o-mini	I'm looking for some hiking shoes. C...	I can recommend the TrekReady Trail...	1199	107
Nov 07, 2024 02:28 PM	▼ chat_with_products				
Nov 07, 2024 02:28 PM	> get_product_documents				

< Prev Next > 25/Page

Implement RAG in Prompt

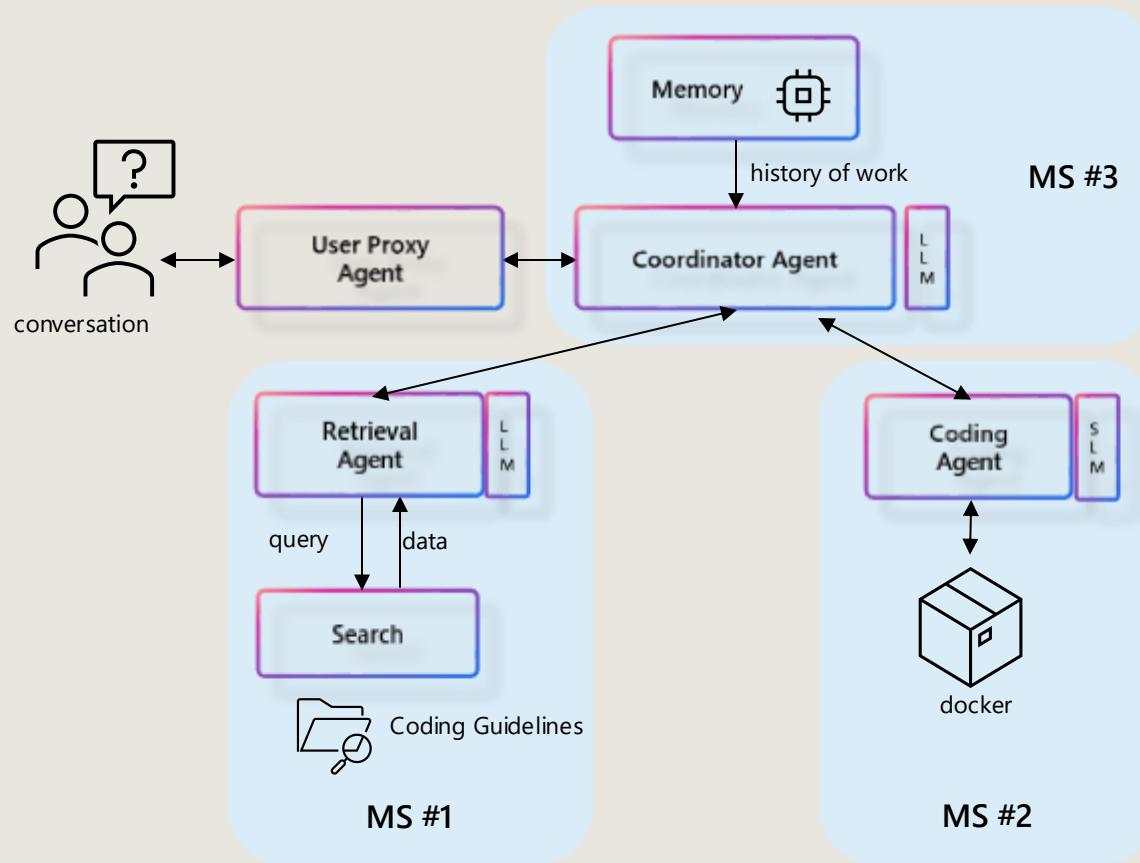


Tag GitHub issues with Prompty



Multi-Agent System

A complex problem is decomposed into smaller, manageable parts, each addressed by specialized agents, effectively a micro-service (MS). These agents work together in a coordinated manner within a workflow to efficiently solve the overall problem.



Critical Design Elements

- ✓ Adaptive planning within scope of existing tightly scoped skills (agents)
- ✓ Handles ambiguity by discussing and refining requirements with human
- ✓ Memory to handle complex long running execution of a plan
- ✓ Effective inter agent communications
- ✓ Test, monitor, release & maintain each agent independently to quickly handle quality & safety issues



<https://aka.ms/ai-agents-prompty>

Resources and code!

What will you learn next?

Register for your next session and keep hacking!

Hackathon runs April 8–30, 2025

aka.ms/agentshack



