



AI Tech Accelerator Community





Building Agents with Code for Production

3 March 2025



Agenda

	Conference Center 1	Conference Center 2 (next door)
10:00 – 12:00	Building Agents with Code for Production Florian Follonier, Sr. Partner Solution Architect for Data & AI	Coworking Space
12:00 – 13:30	Lunch Break	
13:00 – 17:00	Hands-on Lab: Building Agents with code for Production Supported by Florian Follonier, Juan Manuel Servera Bondroit & Martin Abrle	

Reminder: rules of the game



**REGISTER AT LEAST 3
BUSINESS DAYS BEFORE THE
NEXT APPOINTMENT
(THURSDAY BEFORE THE
EVENT)**



**IF YOU CAN'T MAKE IT –
LET US KNOW**



**PLEASE ALWAYS USE [SWISS-
SU@MICROSOFT.COM](mailto:SWISS-SU@MICROSOFT.COM) TO
CONTACT US**



**IF SOMETHING DOESN'T
WORK, DOESN'T FEEL
RIGHT OR COULD BE
BETTER – TELL US**



MUTUAL RESPECT



BE CURIOUS

Register for Coworking

- Make sure to always register – at least 3 business days in advance: [AI Tech Accelerator Attendance](#)
- A maximum of 2 people per startup / company are allowed per day

AI Tech Accelerator - Attendance Coworking

Please take a moment to fill out this poll about the coworking attendance. It is important for us to know the name of each person attending on a specific date. So please distribute this poll to all attendees from your startup joining one of the coworking sessions.

* Required

1. First Name *

Enter your answer

2. Last Name *

Enter your answer

3. E-Mail *

Enter your answer

4. When will you be attending? *

17th of February (Mon)

18th of February (Tue)

VOTE FOR
THE NEXT
1: MANY
SESSION



REQUEST 1:1 EXPERT SESSION



<https://aka.ms/Alrepo>



Exploring Agentic Systems with Azure AI Agent Service

Florian Follonier

Sr. Partner Solution Architect Data & AI



Agenda



What is an Agent



Agentic building blocks and patterns



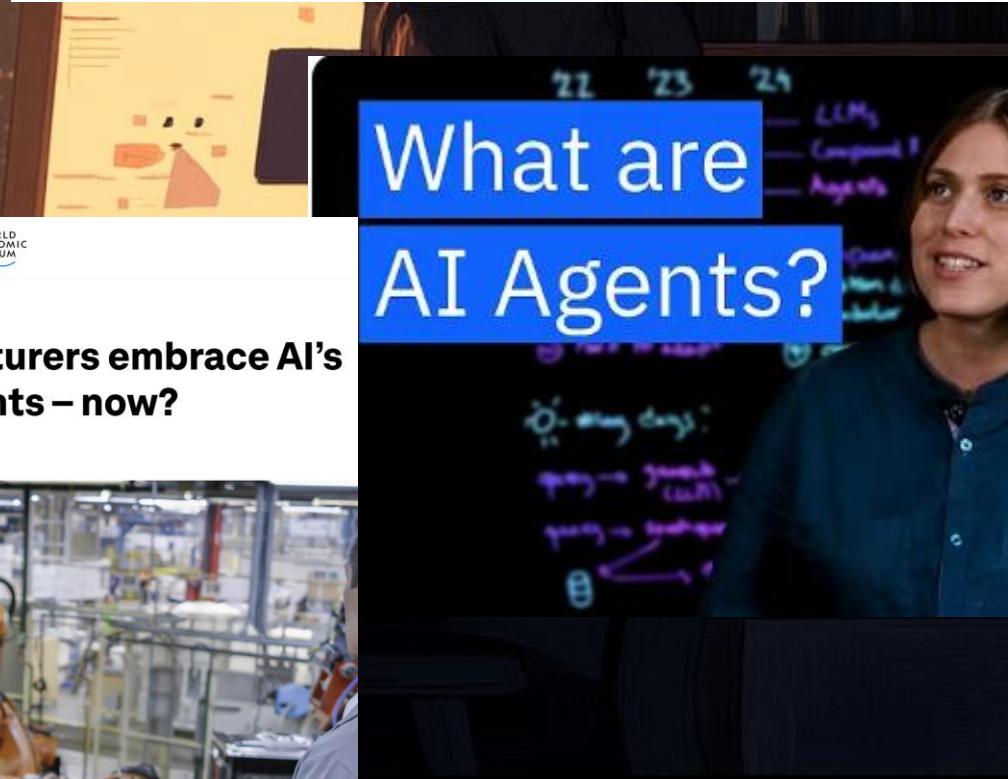
Single-Agent demo



Multi-Agent patterns



Build Your Own Agent



GenAI has come a long way



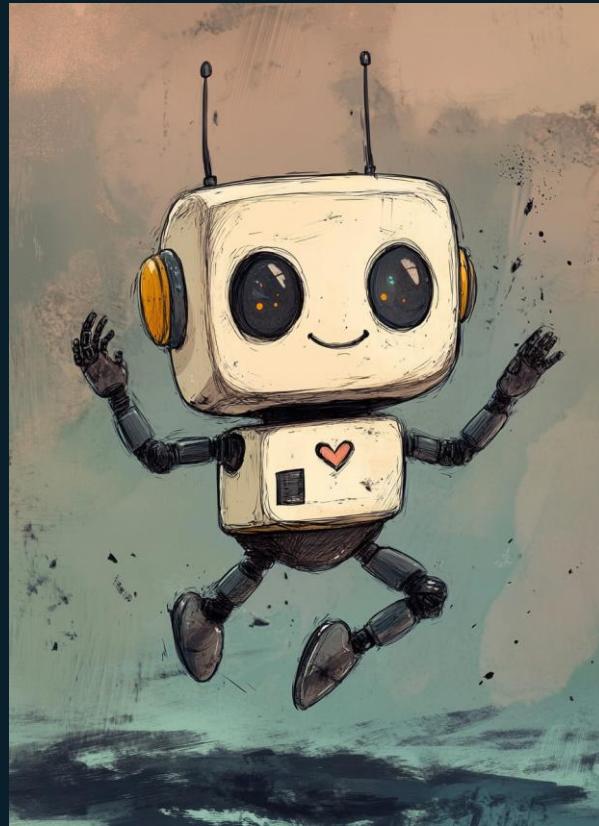
**ML, Deep Learning &
BERT, etc.**
<2021



**LLMs,
ChatGPT & Dall-E3**
2022



**Chat with Your Data
(RAG)**
Early-2023



Agents

Mid-2024

First wave of generative AI Apps

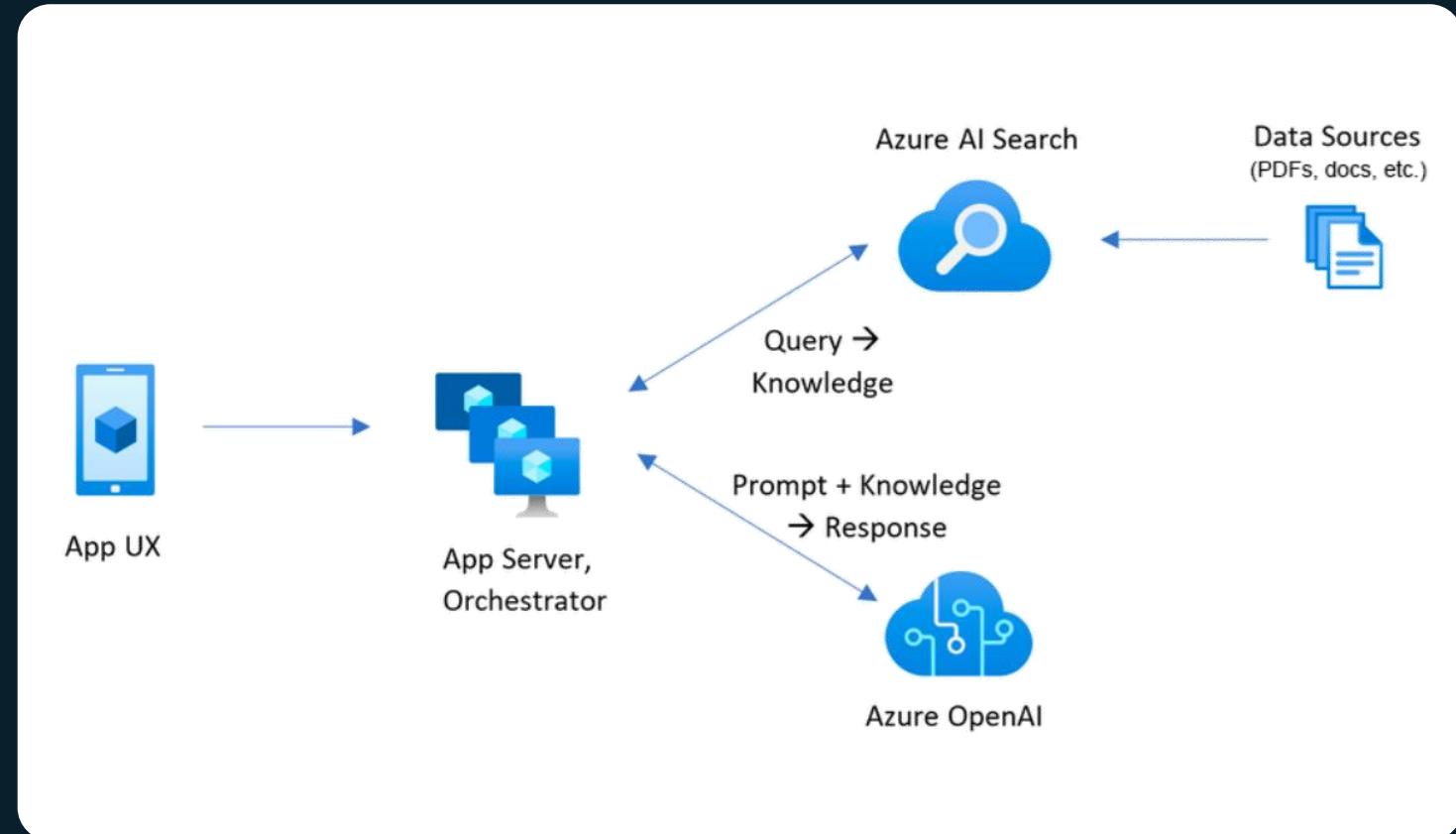
Common use cases:

- Conversational chat on private data
- Text/Document/Audio summarization and classification
- Image description and entity extractions
- Personalized content generation

Prompt engineering

RAG pattern

Application flow is hard-coded



Next wave: Agents

Complex interactions & orchestration

- Virtual assistants
- Customer support
- Intelligent code editors

Tools calling

Many LLM tasks + steps
undefined sequence = agentic reasoning

Improve efficiency and accuracy

Ask a question on a topic?

Do web search? First draft response.
Need more research?
Do revision on response.
Iterate for more details?
Revise, act and respond.

Agentic Reasoning

Question



Search

Response



Revise

Iterate



Research



Agent frameworks and services



Semantic Kernel Agent Framework



Autogen



Langgraph



Azure AI Agent Service

Agents

What is an Agent?

"System that uses a LLM to decide the control flow of an application."

Autonomy Levels:

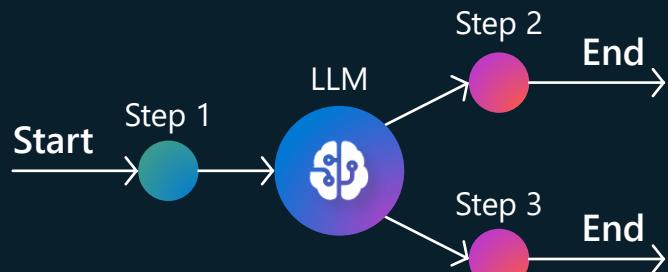
- No Autonomy: Traditional RAG
- Simple: Paths routing
- Fully: Multi-step reasoning & acting

Architectures:

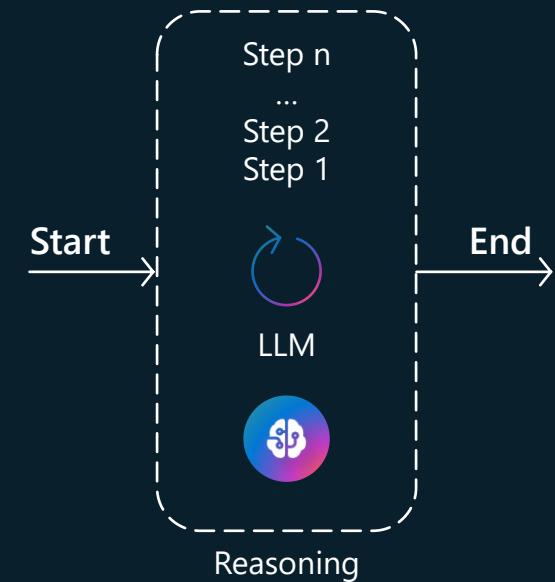
- Single agent
- Multi agent

Wave 1 (2022) -> Wave 2 (2025)

Simple LLM-enabled



Fully autonomous



Less

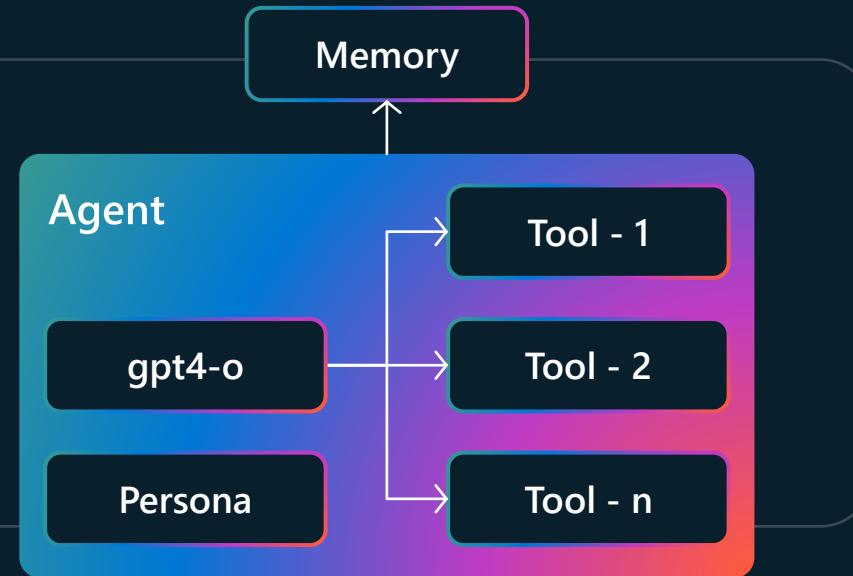
LLM Control

More

Agent Abstractions - Agent First-Class Citizen

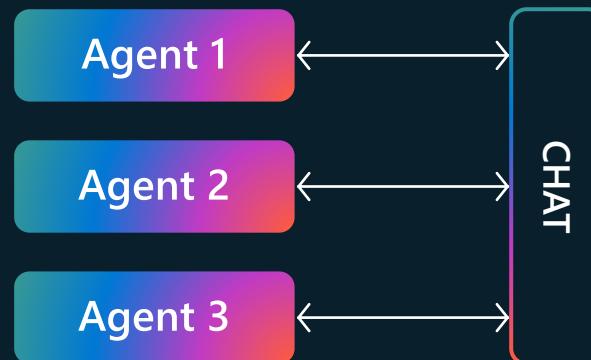
Agent as high-level abstraction

- LLM (gpt4-o, o1 etc.)
- Persona (system prompt)
- Tools (function code calls)

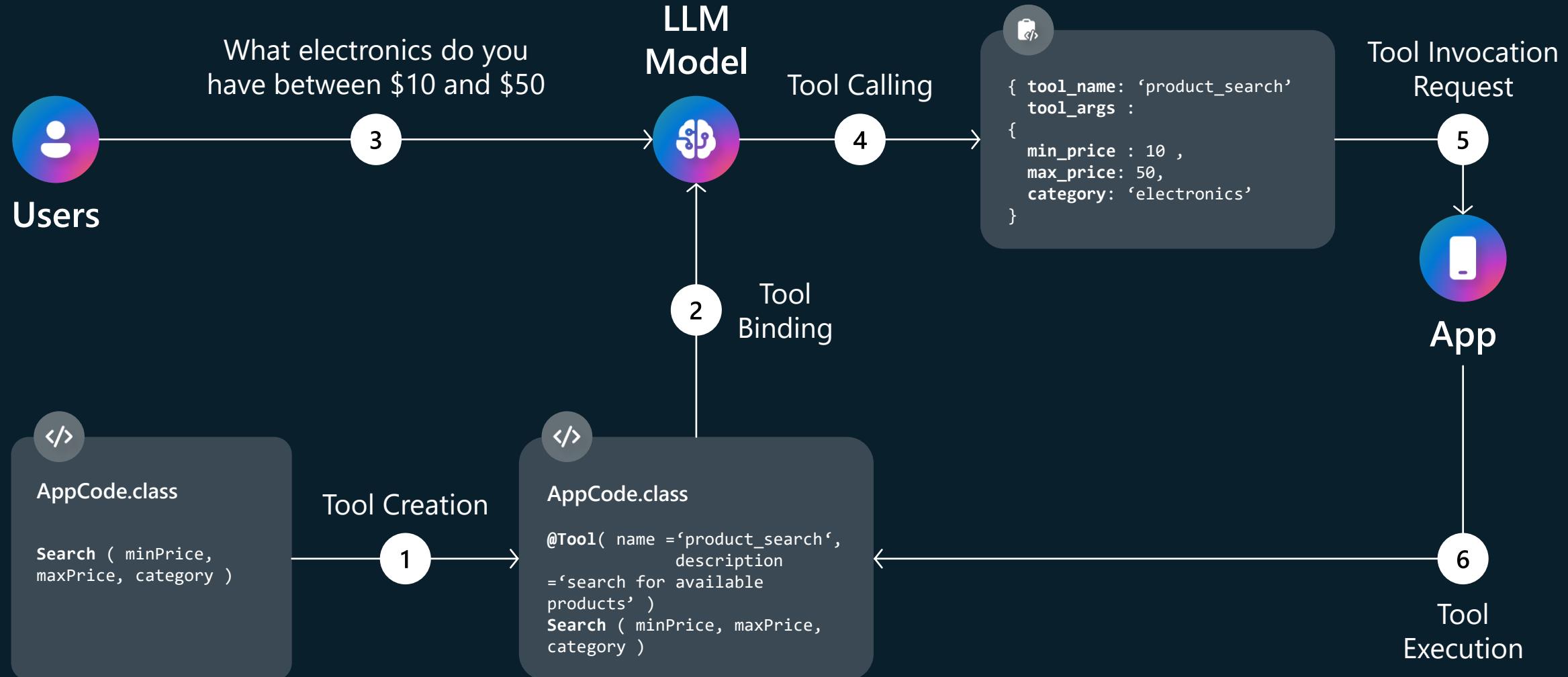


Agent Chat as layer for collaboration

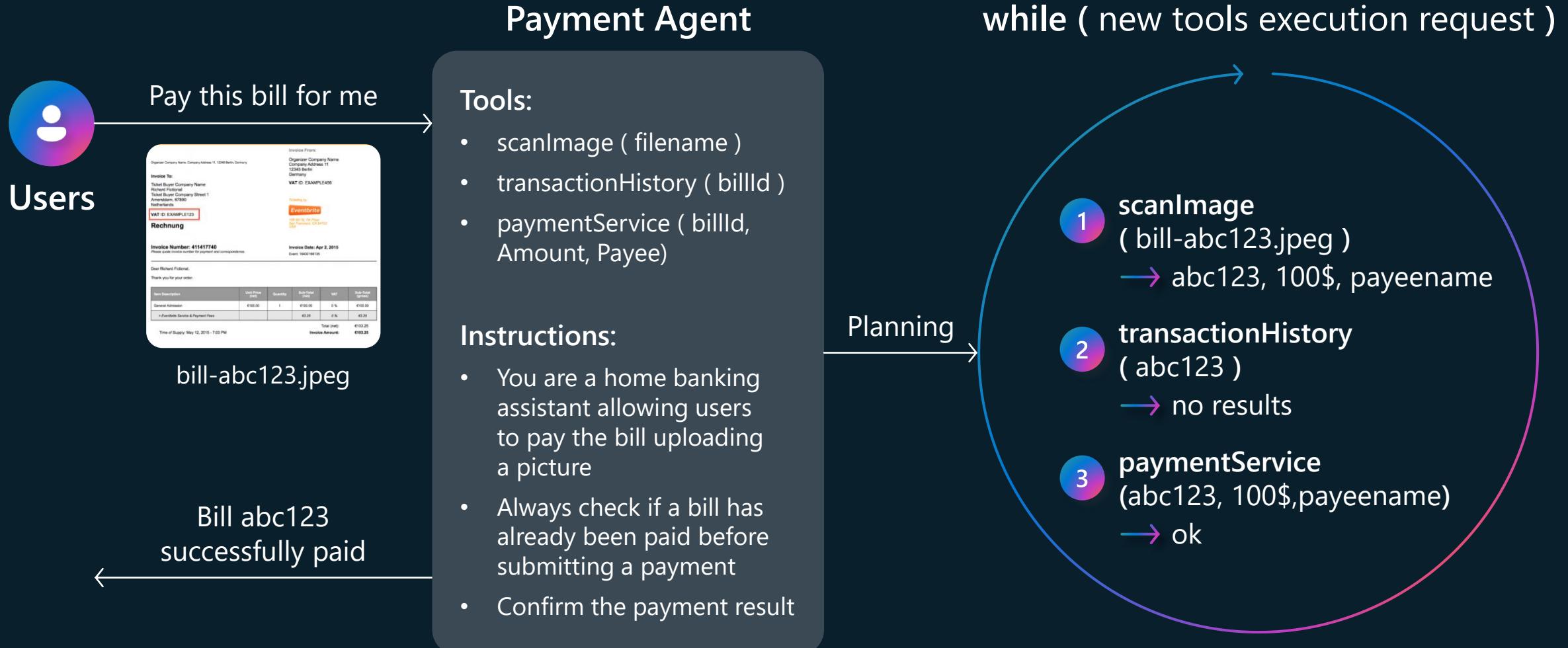
- Multiple agents can engage with each other
- Enables multi-turn or parallel execution



Agentic Pattern - Tools Calling



Agentic Pattern - ReAct Planning with Tools Calling



Agentic Pattern - Memory

Short Term

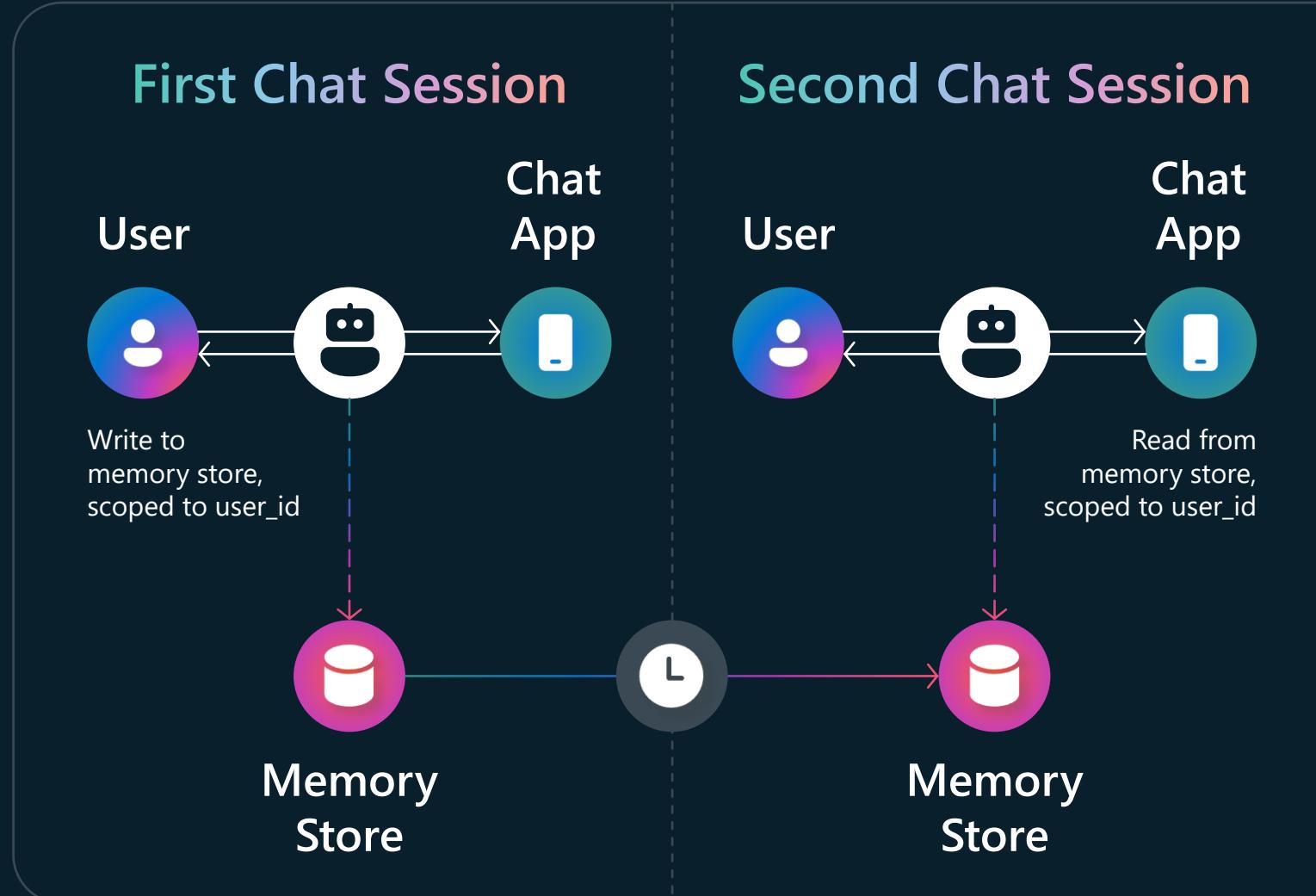
- Access steps info in one loop iteration
- Shared state context
- Chat history

Long Term

- Access steps info in long running conversation
- State persistence

Conversation History Truncation

- Trim by tokens
- Trim by message count
- Trim + summary (LLM call required)



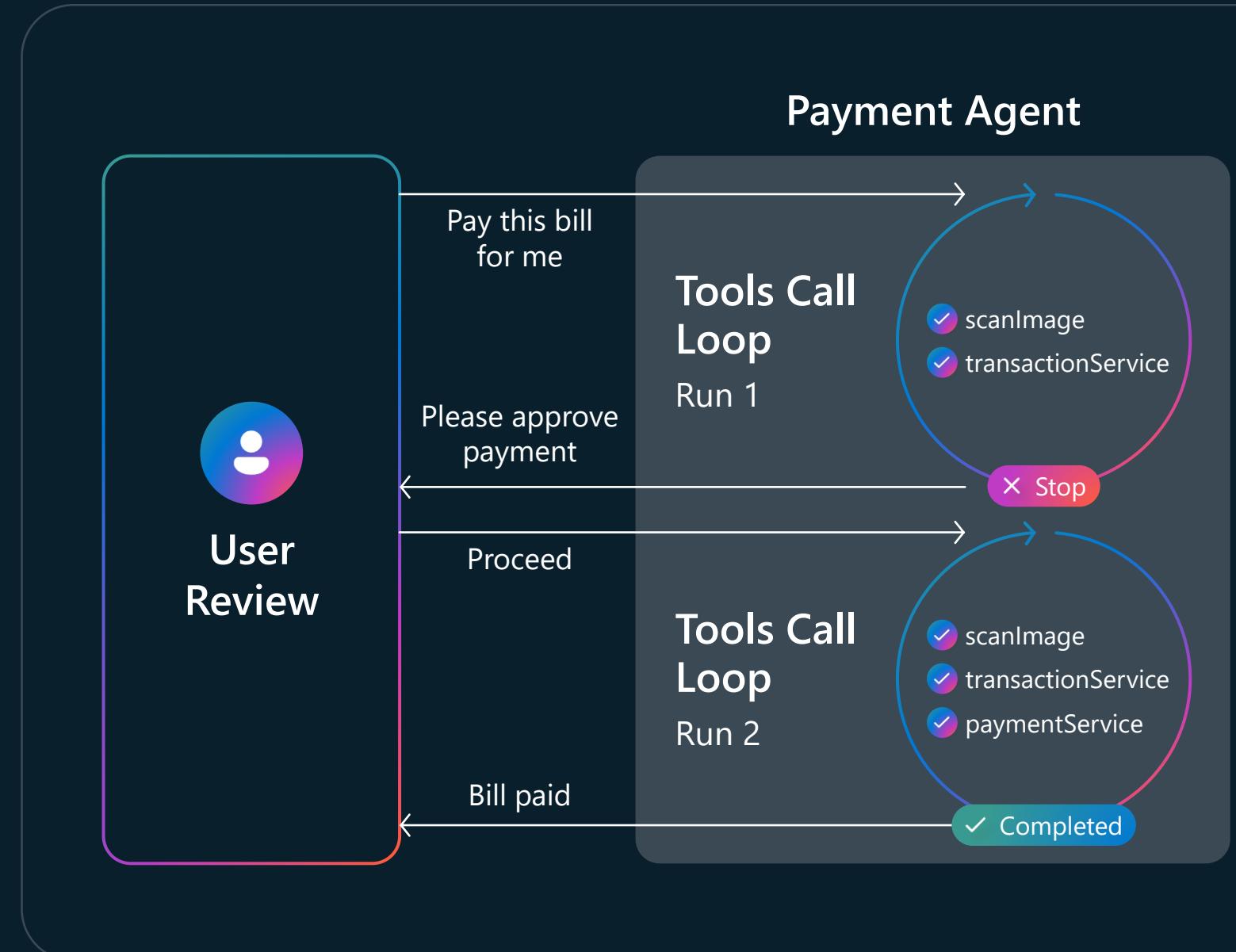
Agentic Pattern - Flow control

Looping Termination

- MaxIterations
- Message termination
- Human step /Human in the loop

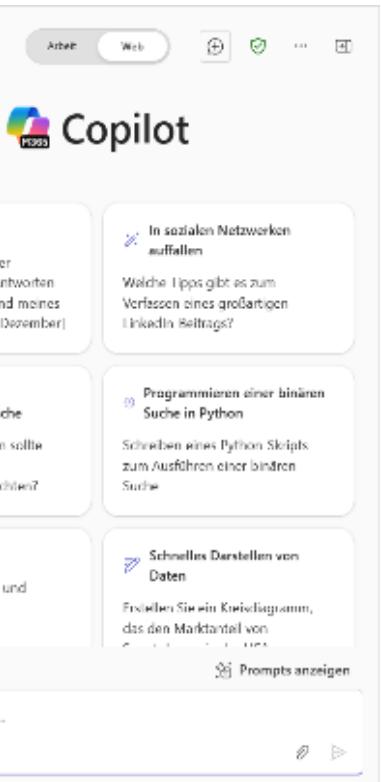
Human in the loop

- Action execution approval
- Escalation
- Data review

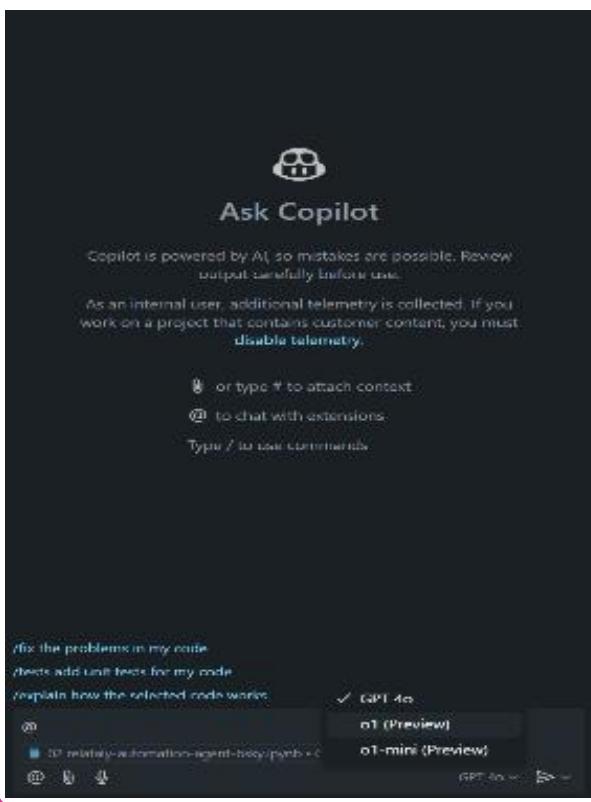


Agent Examples

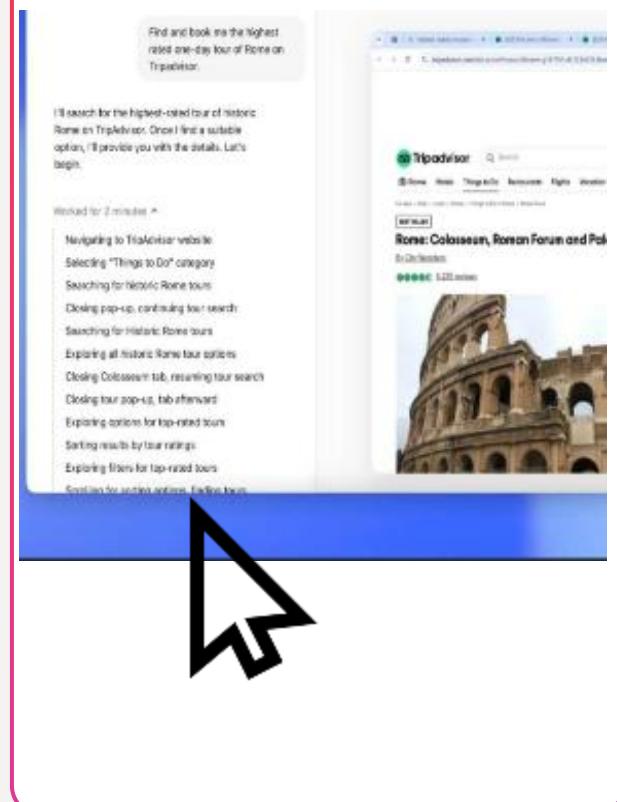
Search Agents like Copilot in Teams



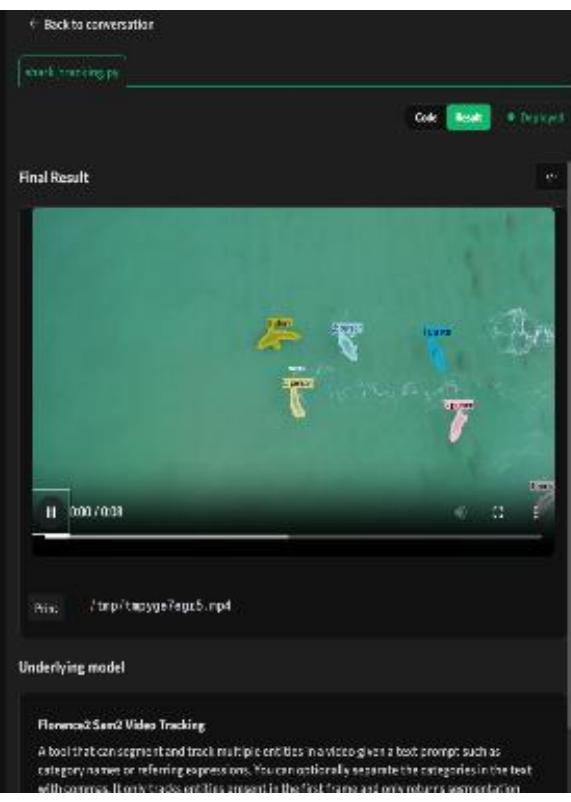
Coding Agents like GitHub Copilot (Chat)



UI Task Agents like OpenAI Operators



Data Science like Open Source Vision Agent



Example for An Advanced Agent: Vision Agent

VisionAgent / Car Tracking and Counting Video Analysis

Can you track and count the cars in this video? Save a video with the tracked cars.

today at 2:39 PM

Generating vision code.
car_tracker.py
Click to view

Executing code.

Editing vision code.
car_tracker.py
Click to view

Executing code.

The task has been completed successfully. The code tracked and counted 165 unique cars in the video. A new video with the tracked cars has been saved at '/tmp/tmpskrel9lw.mp4'. Is there anything else you would like me to do with this video or the results?

today at 2:39 PM

Message VisionAgent

Smart mode

← Back to conversation

car_tracker.py

Version: v2

Code Result Deploy

Final Result

0:17 / 0:29

Print [`{'unique_car_count': 165, 'output_video_path': '/tmp/tmp1nf02zea.mp4'}`]

Underlying model

Example for An Advanced Agent: Vision Agent

README Apache-2.0 license



VisionAgent

VisionAgent python 3.9 | 3.10 | 3.11

VisionAgent

VisionAgent is a library that helps you utilize agent frameworks to generate code to solve your vision task. Check out our discord for updates and roadmaps! The fastest way to test out VisionAgent is to use our web application which you can find [here](#).

Installation

```
pip install vision-agent
```

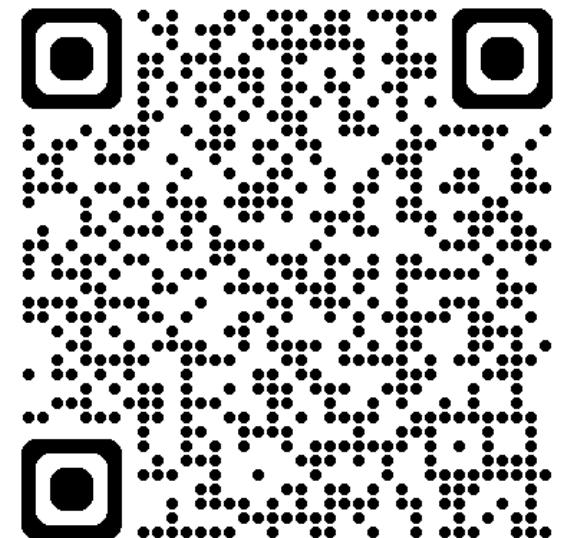
```
export ANTHROPIC_API_KEY="your-api-key"  
export OPENAI_API_KEY="your-api-key"
```

NOTE: We found using both Anthropic Claude-3.5 and OpenAI o1 to be provide the best performance for VisionAgent. If you want to use a different LLM provider or only one, see 'Using Other LLM Providers' below.

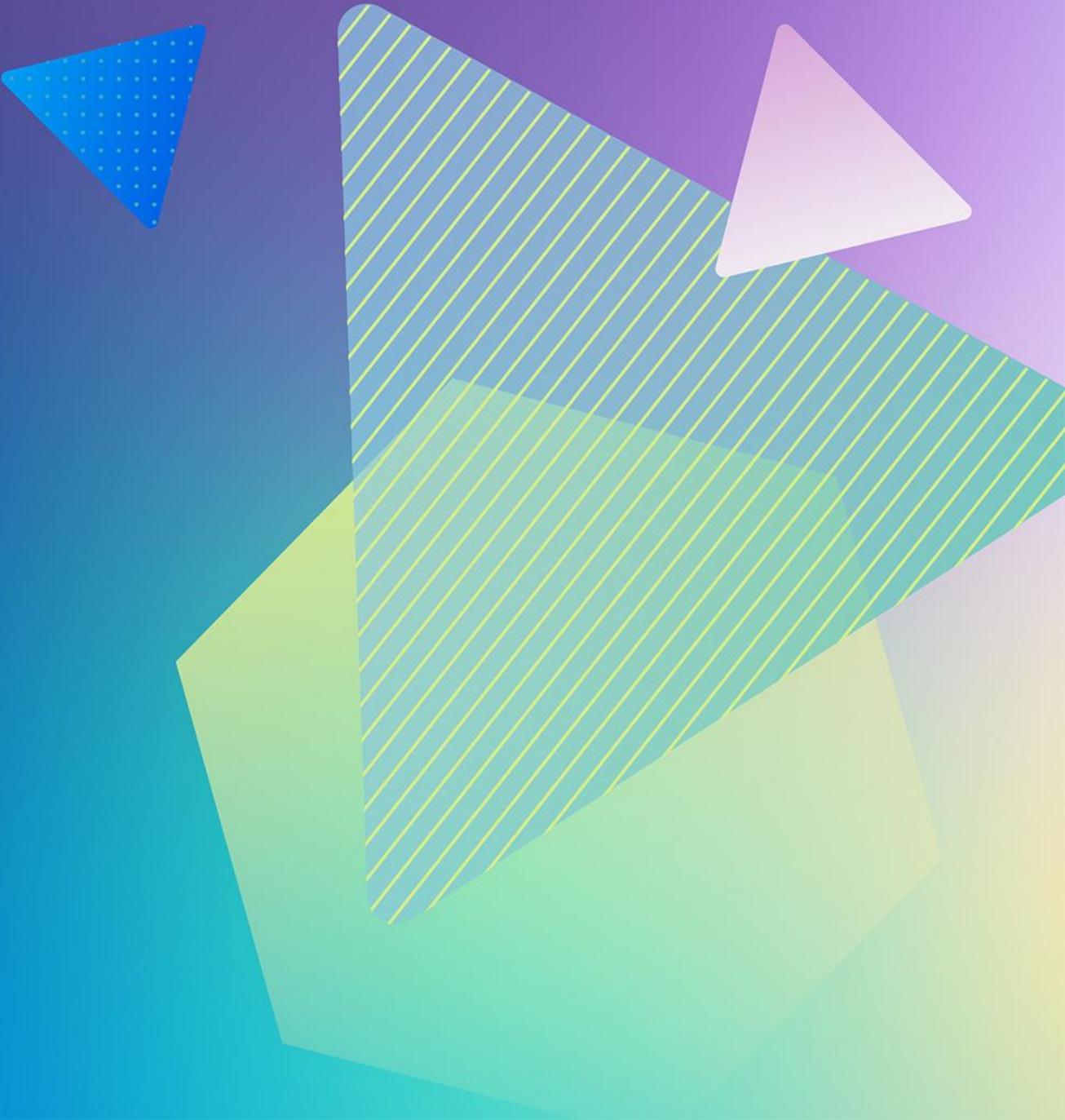
Deployments 210
 github-pages 15 hours ago
[+ 209 deployments](#)

Languages

• Python 100.0%



Azure AI Agent Service



Announcing



Azure AI Foundry



Copilot Studio



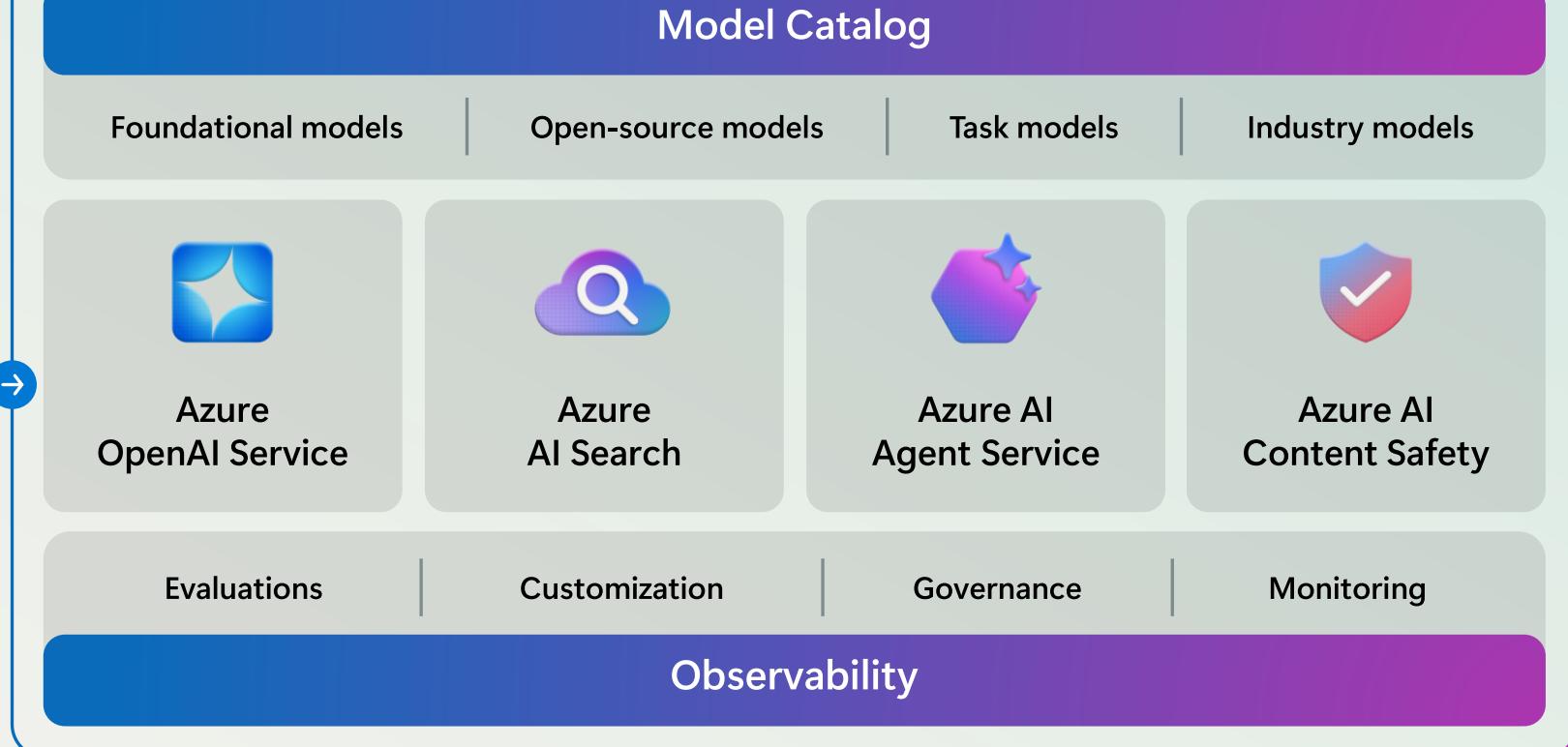
Visual Studio



GitHub



Azure AI
Foundry SDK

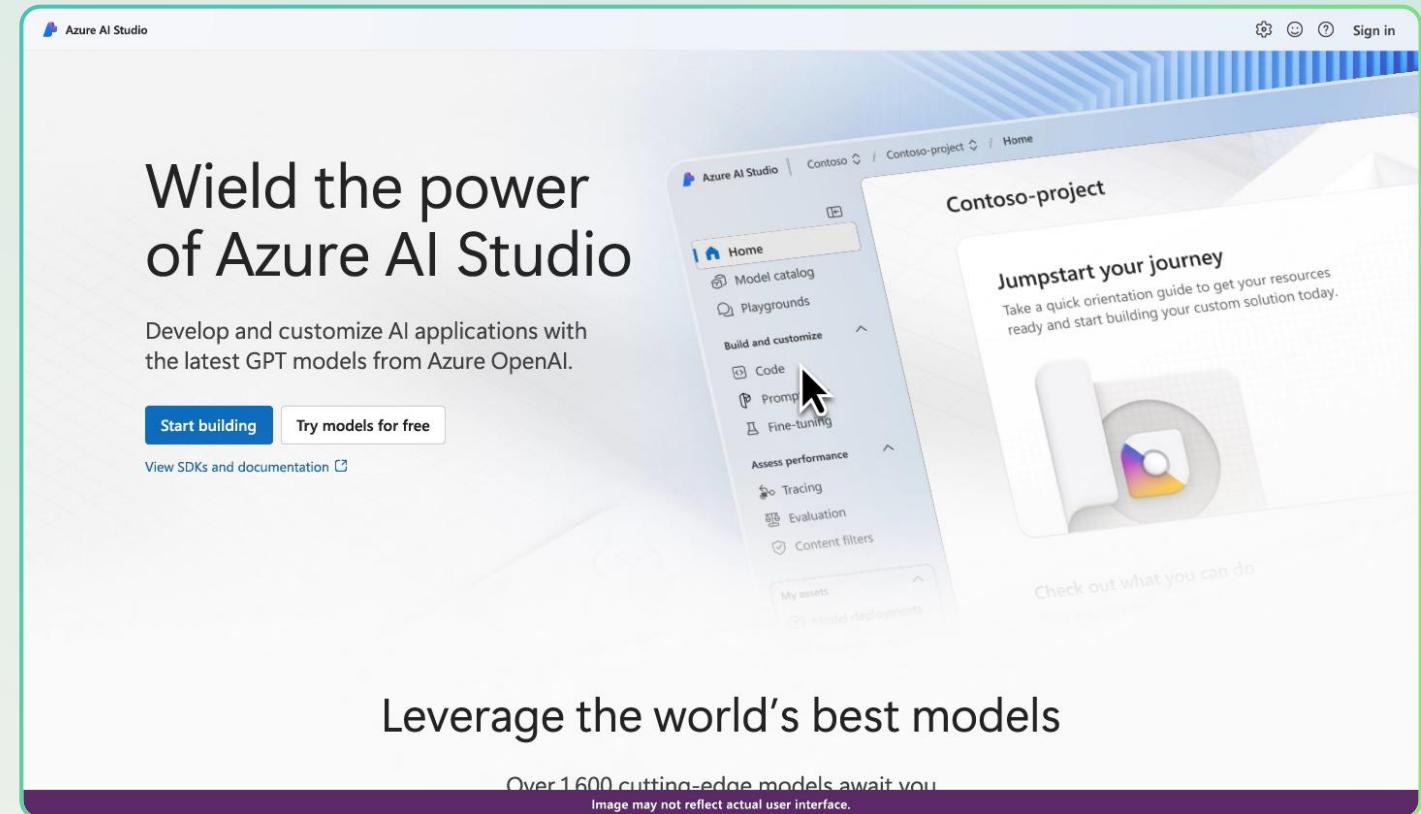


Azure AI Foundry SDK

The Azure AI Foundry SDK is a comprehensive toolkit for developers, offering pre-built modules and resources to integrate AI functionalities seamlessly into applications.

- Access our most popular models through a single interface
- Easily Azure AI capabilities into your application with a single project client
- Unlock another level of intelligence with Azure AI Agents
- Integrated tracing enables you to log back to AI Studio projects
- Evaluate your apps locally, in the cloud, and in production using state-of-the-art safety and quality evaluators
- Incremental Azure Building Block app templates beyond SDKs, including templates for copilot scenarios, hosted in web, container, function app, and more

Move Seamlessly between UI and Code



Leverage the world's best models

Over 1600 cutting-edge models await you.
Image may not reflect actual user interface.



The Azure AI Foundry SDK provides a local developer experience that reduces the complexity of using multiple resources together in code when building AI apps and agents.



Azure AI Agent Service SDK



Azure AI Foundry SDK – Agent Service

Azure OpenAI
Assistants API

- File Search
- Code Interpreter

Model Catalog



Azure OpenAI Service
(GPT-4o, GPT-4o mini)



Models-as-a-Service



Llama 3.1-405B-Instruct



Mistral Large



Cohere-Command-R-Plus

Extensive Ecosystem of Tools

Knowledge



Microsoft Fabric (coming soon)



SharePoint (coming soon)



Grounding with Bing Search



Azure AI Search



Your own licensed data (coming soon)



Files (local or Azure Blob)

Actions



Azure Logic Apps (coming soon)



OpenAPI 3.0 Specified Tools



Azure Functions

Built-In Enterprise Readiness

BYO-file storage
(coming soon)

BYO-search index

BYO-thread storage

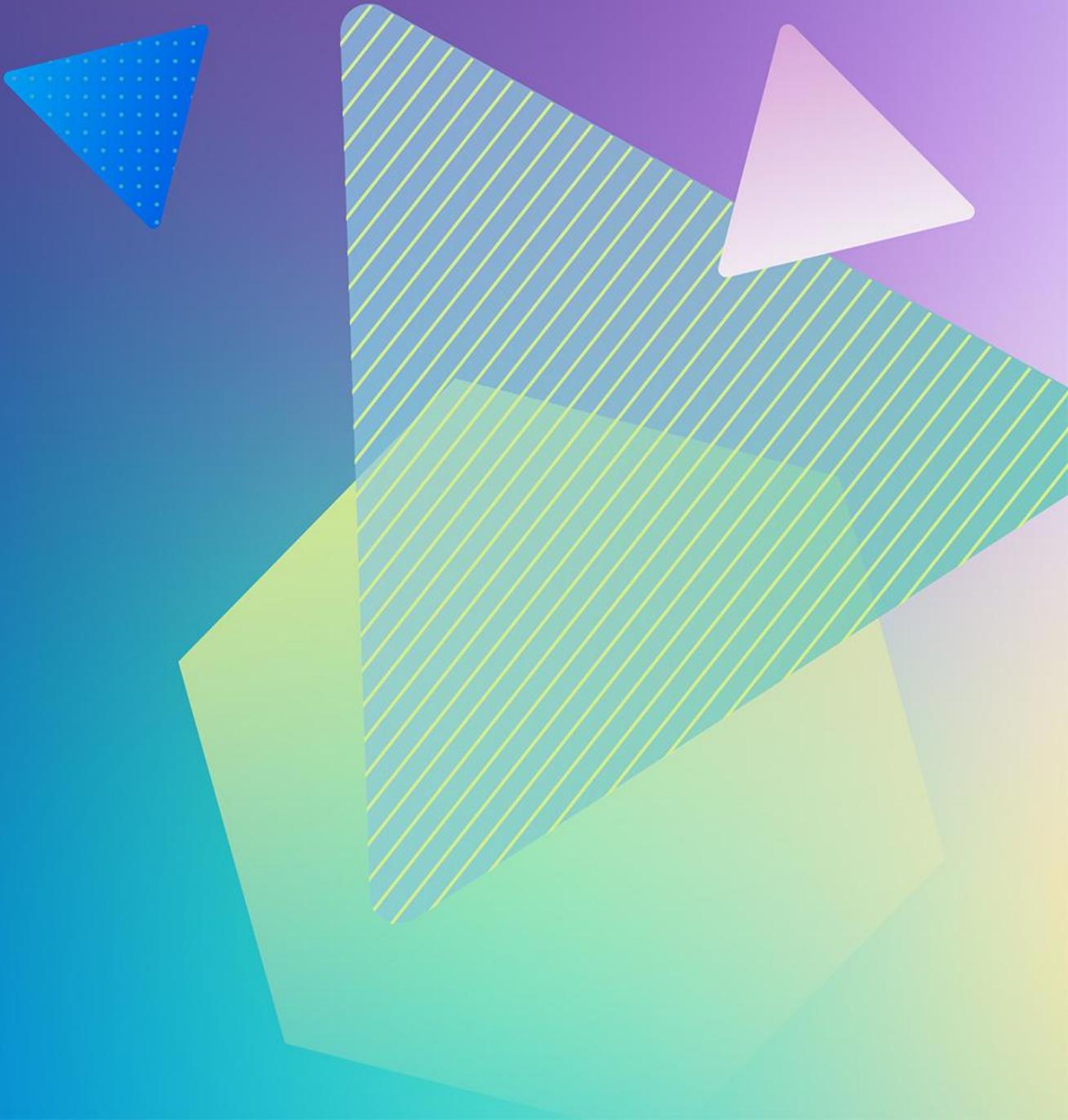
BYO-virtual network
(coming soon)

OBO Authorization Support

Enhanced Observability

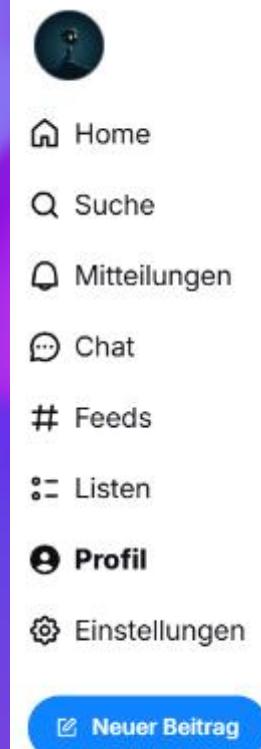
Practical Example

Social Media Agent





Bluesky Social Media Agent



The image shows a Bluesky profile page for the account "relataly.com AI News". The profile picture is a dark, abstract image of a figure. The bio reads: "I am trying to be a good AI agent posting relevant News and Tutorials on AI-related topics: GenerativeAI, ChatGPT, OpenAI, Agents, Coding & Data Science". The stats show 118 Follower, 21 Folge ich, and 2947 Beiträge (Posts). A purple oval highlights the "Beiträge" (Posts) count. Below the bio, it says "Brought to you by GPT-4o, Azure AI Agent Service and Bing News on Azure Functions". A link to "Visit relataly.com" is provided. The main feed shows three recent posts from the account, each with a timestamp of "1h" and a caption about AI news. The first post discusses AI revolutionizing real estate, the second discusses Amazon's AI hub acquisition, and the third discusses AI stocks and their partnership with Oracle and SoftBank.

relataly.com AI News @relataly.bsky.social · 1h
AI is revolutionizing real estate by automating tasks, personalizing home searches, aiding data-driven decisions, and enhancing customer experiences. Discover how AI is transforming the industry! #AI #RealEstate [Read more](#)

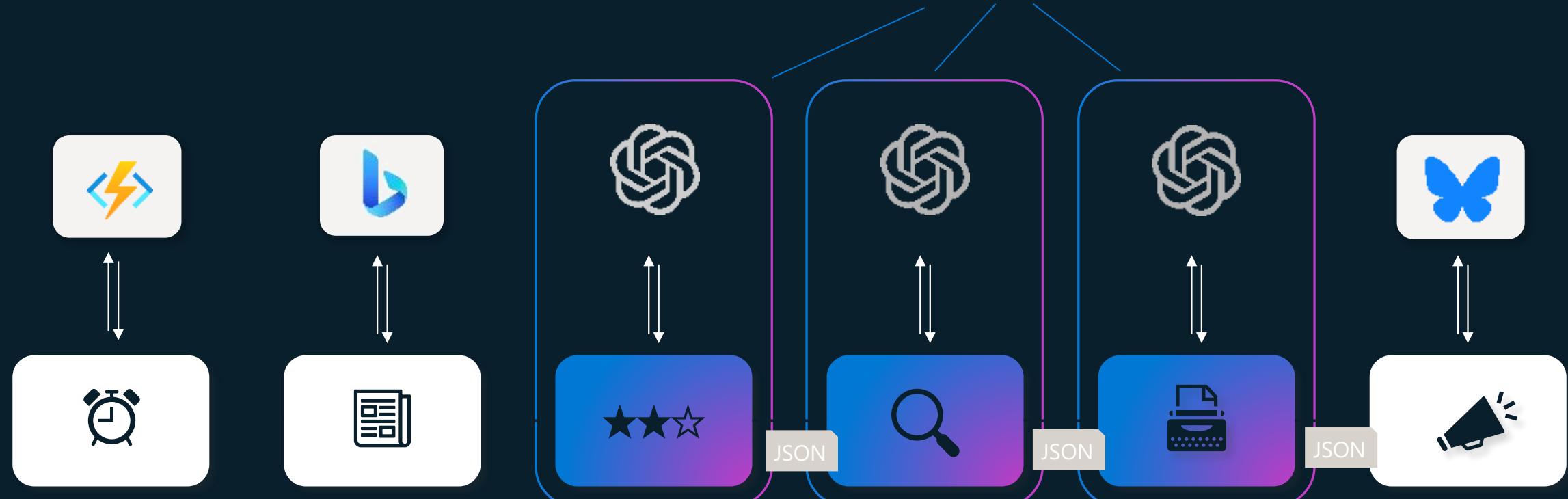
relataly.com AI News @relataly.bsky.social · 1h
Amazon has acquired land in Ohio for a massive AI hub! This could mark a significant investment in AI infrastructure. Discover how this move compares server farms and data centers. #AI #Amazon #Investment

relataly.com AI News @relataly.bsky.social · 1h
AI stocks saw a surge following OpenAI's new partnership with Oracle and SoftBank, potentially leading to \$500 billion in investment! Explore how this alliance might reshape the future of AI infrastructure. #AI #Investment

relataly.com AI News @relataly.bsky.social · 1h

A “Classic” GenAI-infused Process

LLM decision freedom is narrowed to three separate specific tasks



Time
Trigger
(every 4h)

Fetch News
from Bing
Search

Evaluate news
for relevance

Check for
duplicates
with previous
posts

Select best
rated news
and create
post

Publish
Post

Structured Responses: Curse and blessing

Evaluating News

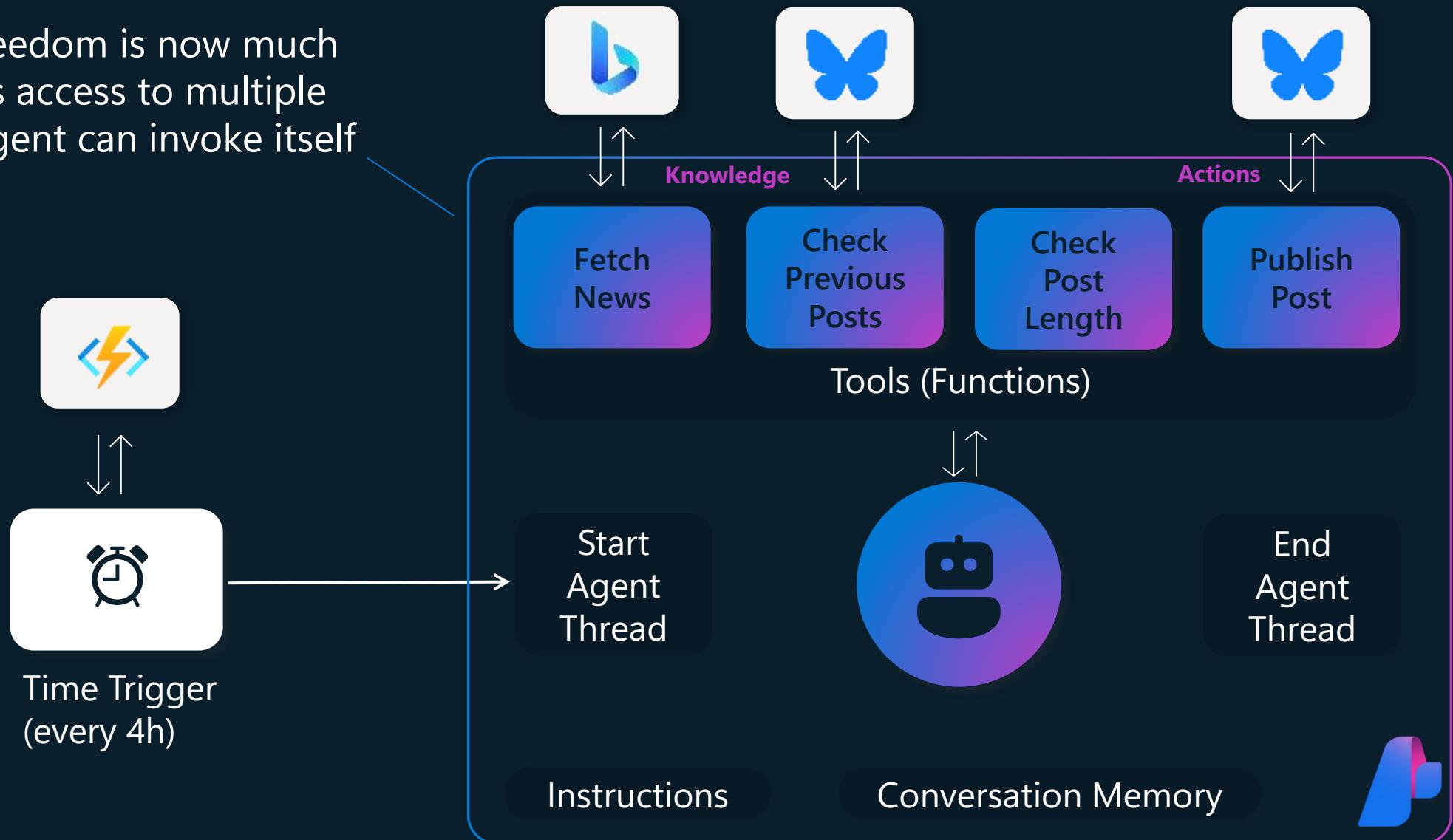
```
#### Define OpenAI Prompt for news Relevance
def select_relevant_news_prompt(news_articles, topics, n):
    instructions = f'Your task is to examine a list of News Titles and return a list of boolean values that indicate which of the News Titles are in scope of a list of topics. \
    Return a list of True or False values that indicate the relevance of the News Titles.'
    task = f"Which of the following news titles: {news_articles} are within the scope of these topics: {topics}?"
    sample = [
        {"role": "user", "content": f"Which of the following {n} News Titles: [new AI model available from Nvidia, We Exploded the AMD Ryzen 7, XGBoost 3.0 Making Decision Forest A
            are within the scope of these topics: {topics}?"}, 
        {"role": "assistant", "content": "[True, False, True, False, False]"}, 
        {"role": "user", "content": f"Which of the following {n} News Titles: [new AI model available from Nvidia, We Exploded the AMD Ryzen 7, XGBoost 3.0 Making Decision Forest A
            are within the scope of these topics: {topics}?"}, 
        {"role": "assistant", "content": "[True, False, True]"}]
    return instructions, task, sample
```



{ "relevant_news": [true, false, true] }

The Agentic Approach

LLM decision freedom is now much wider and spans access to multiple tools that the agent can invoke itself



Agents allow for more robust workflows

```
instructions =
```

You are a helpful assistant with the goal to post about relevant AI news on Bluesky social media. When a user requests, you will fetch the latest AI news and create a post on Bluesky with a tweet and a link to the news article.

Follow these steps to ensure accurate and concise responses:

- 1. **Fetch News from Bing Search**:** Always use the 'search_for_relevant_news_via_bingsearch' function to retrieve any AI related news.
- 2. **Get Your Recent Bluesky Feed**:** Always use the 'receive_previous_posts_from_bluesky_social_media' function to retrieve the latest posts to avoid redundant posts.
- 3. **Evaluate the News**:** Identify the most interest news article from Bing search results considering previous posts to avoid posting about the same topic twice.
- 4. **Create a Tweet**:** Create a tweet text about the selected news (avoid topics from previous posts). Always use the 'check_tweet_length' function to ensure the tweet is within the 280-character limit. Avoid adding the url into the text and instead provide the url as part of the call_function_to_post_on_bluesky_social_media function.
- 5. **Error Handling**:** If there are issues, inform the user about the problem and end the process.

Demo: Social Media News Agent

The screenshot shows a Jupyter Notebook interface within a dark-themed code editor. The top bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help' menus. A search bar displays 'AI icebreaker 2025 agents'. The left sidebar has sections for 'EXPLORER', 'OPEN EDITORS', and 'AI ICEBREAKER 2025 AGENTS', listing files like 'main.py', '.env', and 'requirements.txt'. The main area shows a code cell containing Python code for an 'Automation Agent'.

```
import os
import logging
import json
import requests
import pandas as pd
from typing import Any, Callable, Set

# Azure AI Projects
from azure.identity import AzureCliCredential # or DefaultAzureCredential
from azure.ai.projects import AIProjectClient
from azure.ai.projects.models import FunctionTool, ToolSet
from azure.keyvault.secrets import SecretClient

# Bluesky
from atproto import Client as BskyClient

azure_logger = logging.getLogger("azure")
azure_logger.setLevel(logging.WARNING) # Set to WARNING, ERROR, or CRITICAL to reduce logs

# Optionally, suppress other verbose logs
logging.getLogger().setLevel(logging.WARNING)

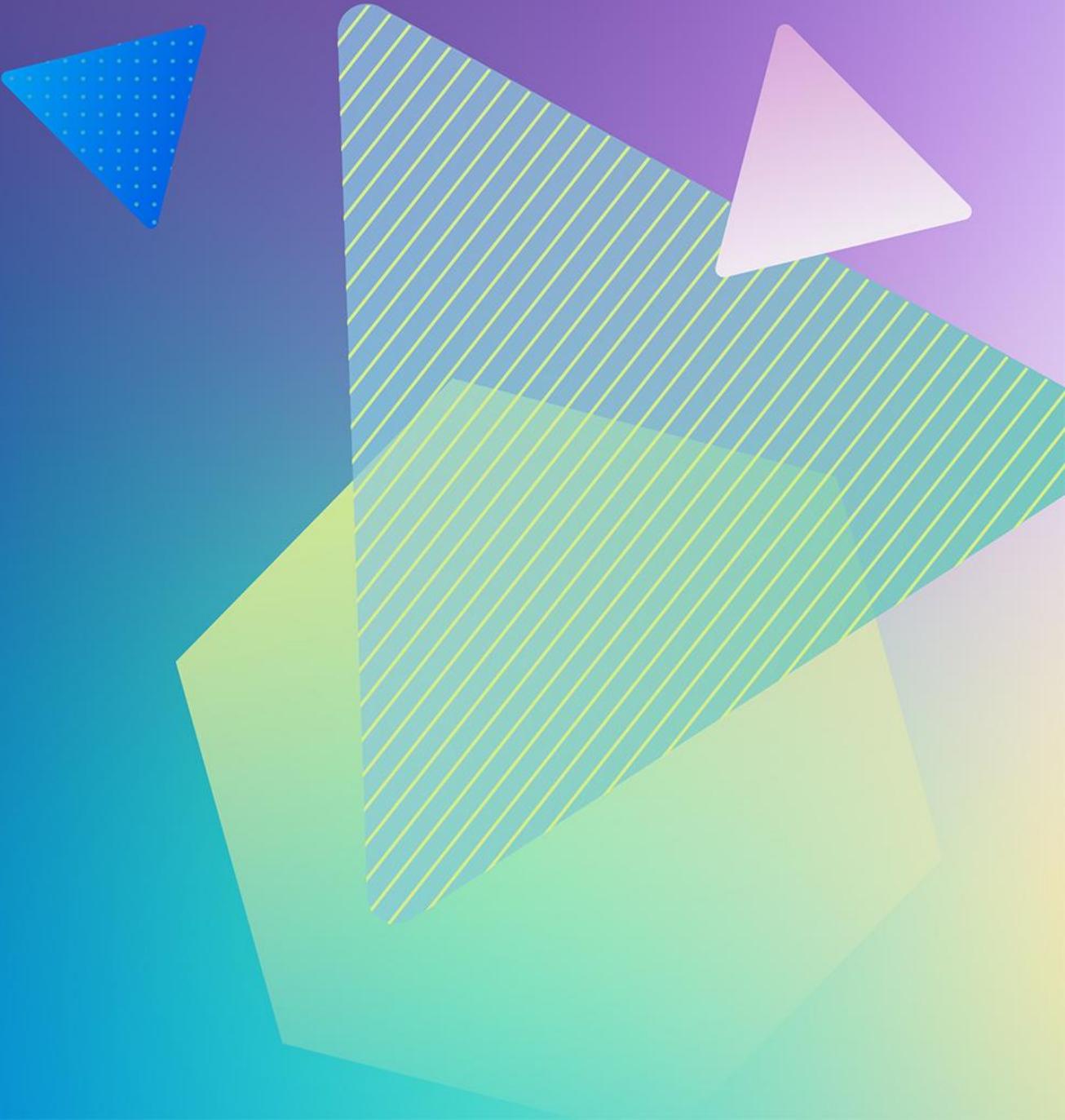
# -----
# 1) Load Environment Variables (Locally) or from Key Vault
# -----
# If you're storing secrets in .env locally, do:

keyvault_name = "keyvaultforbot"
kv_client = SecretClient(f"https://{keyvault_name}.vault.azure.net/", AzureCliCredential())

# Pull secrets from environment variables
try:
    PROJECT_CONNECTION_STRING = kv_client.get_secret("project-connection-string").value
    AZURE_FUNCTIONS_KEY      = kv_client.get_secret("azure-relatly-functions-key").value
    BSKY_USERNAME            = kv_client.get_secret("bskyusername").value
    BSKY_PASSWORD            = kv_client.get_secret("bskypw").value
    BING_API_KEY              = kv_client.get_secret("bing-search-api").value

```

Lessons Learned



Lessons Learned

Tooling

Tool design significantly impacts performance
Wrap APIs. Encapsulate functions into tool components.

Testing

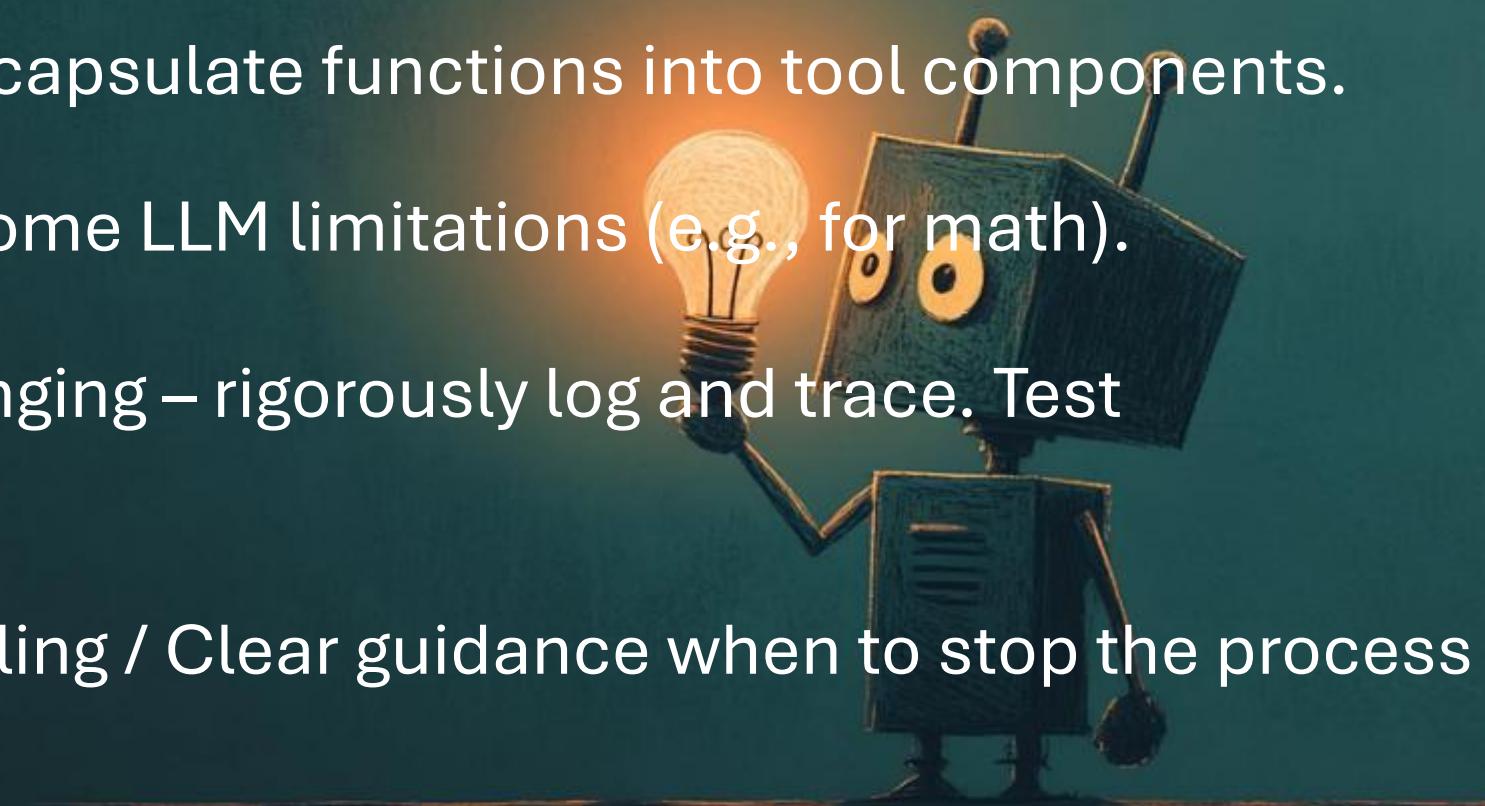
Can be challenging – rigorously log and trace. Test components.

Prompting

Add fault handling / Clear guidance when to stop the process

Architecture

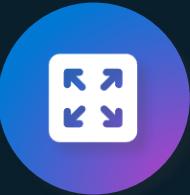
Start with 1 agent. Expand to multiple agents to manage complexity



Multi-Agent Architectures



Single Agent Architecture - Scaling



As the system grows you might run into scaling challenges

Too many tools. Tool hallucinations

Agent context (a.k.a. prompt) grows too much and it fails to follow instructions

Handling complex and dynamics tasks spanning different business domains



Multi agent architecture opportunities

Manageability – Modular agents reduce development and testing complexity

Predictability – More control over application flow using structured agents communication

Flexibility – Ease to incorporate new agents as solution domains increase

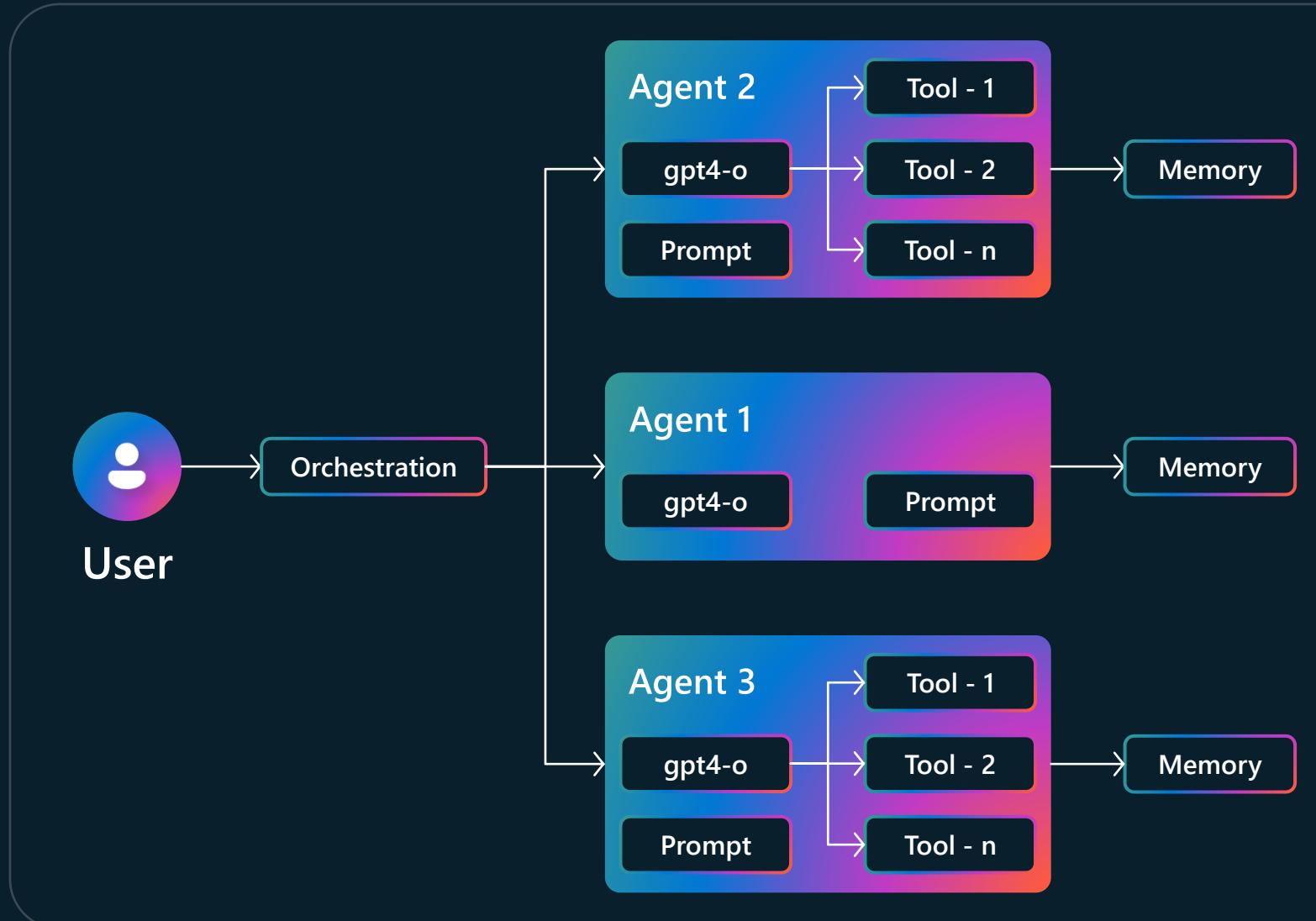
Multi Agent Logical Architecture

Each agent is specialized in different tasks or aspects of a problem

Agents can communicate and coordinate with each other. Structured orchestration is crucial

2 primary categories based on orchestration types

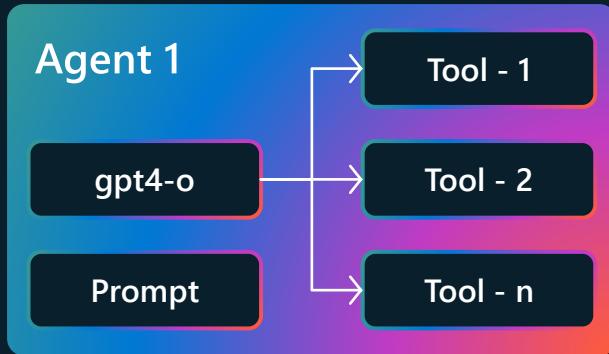
- Vertical Architecture
- Horizontal Architecture



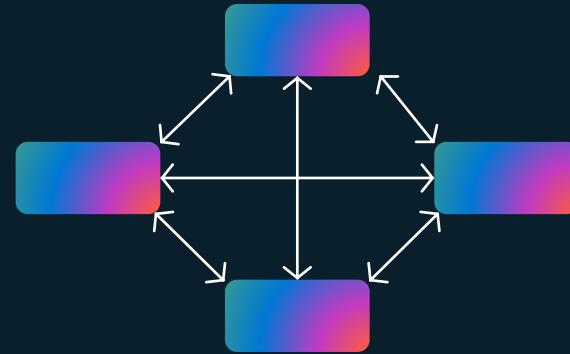
Agents orchestration and communication styles

Start here
→

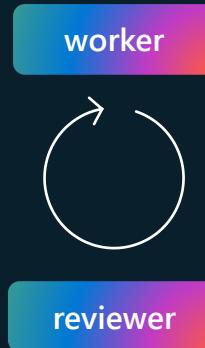
Single Agent



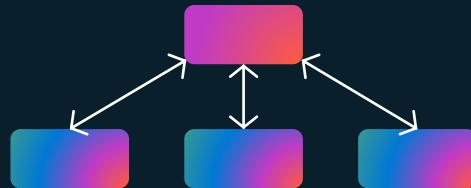
Network



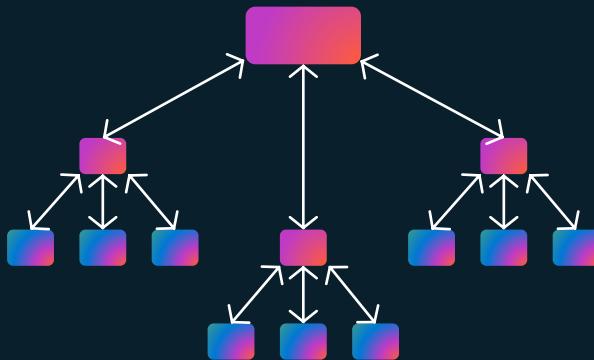
Reflection



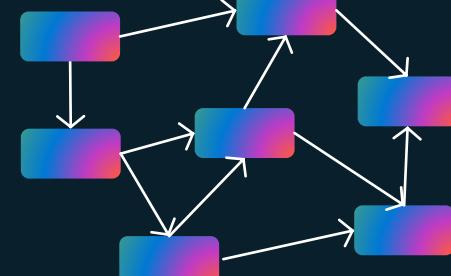
Supervisor



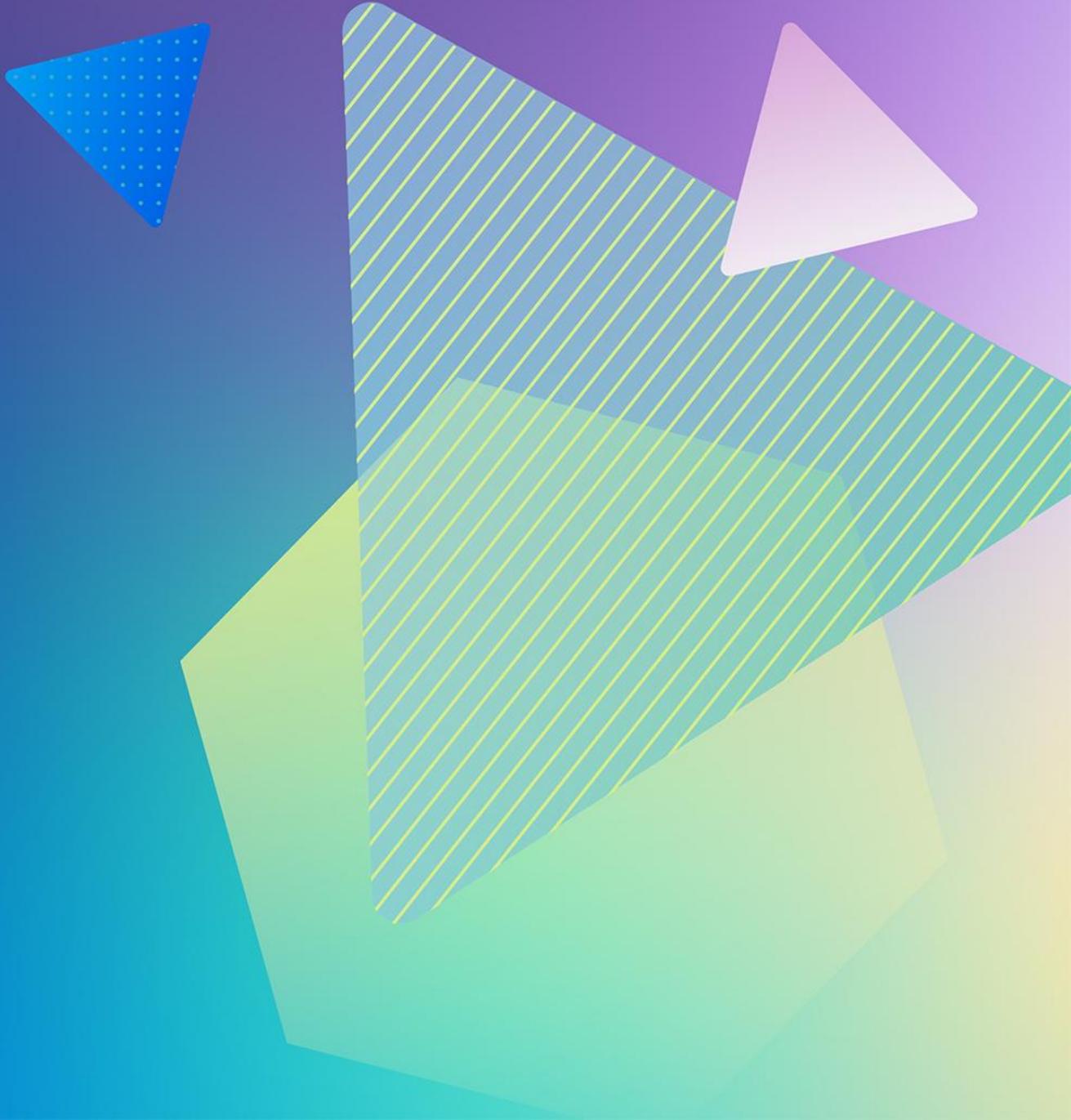
Hierarchical



Custom

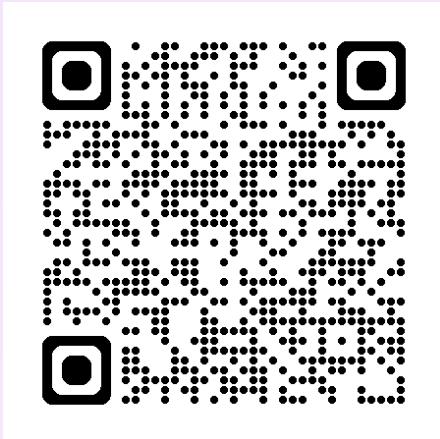


Build Your Own Agent!



GitHub Repo

[flo7up/Agent-Workshop-AI-Accelerator](https://github.com/flo7up/Agent-Workshop-AI-Accelerator)



3bebe
313a6
604e0
4976f
01c7b
504dbb3

Screenshot of the GitHub repository page for "Agent-Workshop-AI-Accelerator".

The repository is private, created by flo7up, and has 5 commits from 4a103b9 (3 days ago). The commits include updates to the solution, .gitattributes, .gitignore, LICENSE, X.env, readme.md, and requirements.txt.

The README file contains the text "Azure AI Agent Tutorial" and a note: "This repository demonstrates how to build an Azure AI Agent that uses the Bing Search".

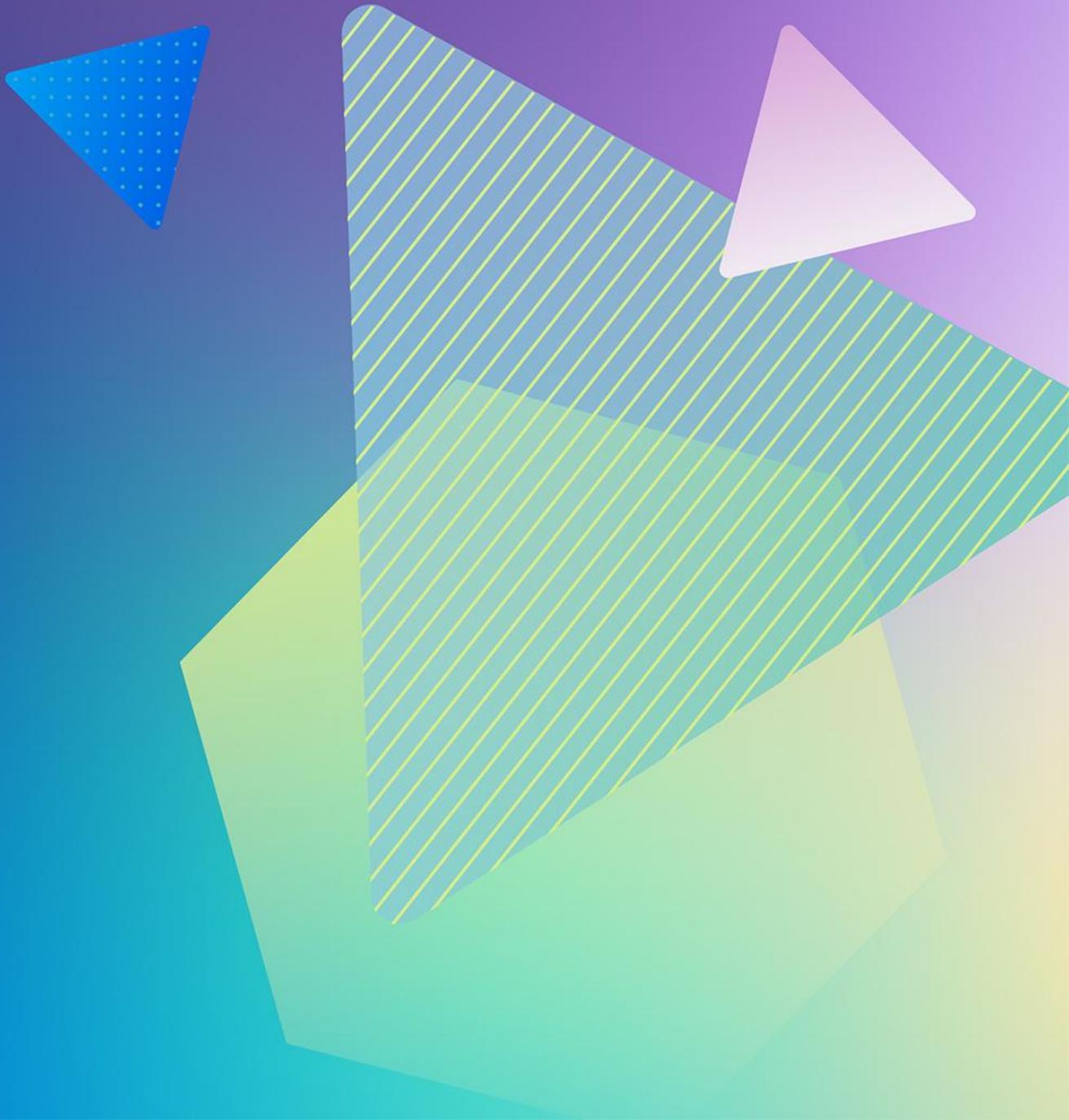
The repository has 1 watch, 0 forks, and 0 stars. It also lists the Readme, MIT license, Activity, and Packages sections.

Commit	File	Type	Date
4a103b9 · 3 days ago	Updates	5 Commits	
	02 Solution	Updates	3 days ago
	.gitattributes	Initial commit	3 days ago
	.gitignore	Update .gitignore	3 days ago
	LICENSE	Initial commit	3 days ago
	X.env	Update	3 days ago
	readme.md	Initial commit	3 days ago
	requirements.txt	Initial commit	3 days ago

Thank You

Florian Follonier

Sr. Partner Solution
Architect Data & AI



Lunch

RESUME AT 13:30





Afternoon – Building Agents with Code for Production, Hands-on Lab



