

Unlocking LLM Potential: A Journey with APIs



Ricardo Covo

Principal Consultant – App Dev & Modernization
@ Centrilogic Inc.

<https://www.linkedin.com/in/ricardocovo/>

CONF42

Agenda

Environment Overview

Basics

- Demo

Chat Completion with Tools

- Demo

Assistants Overview

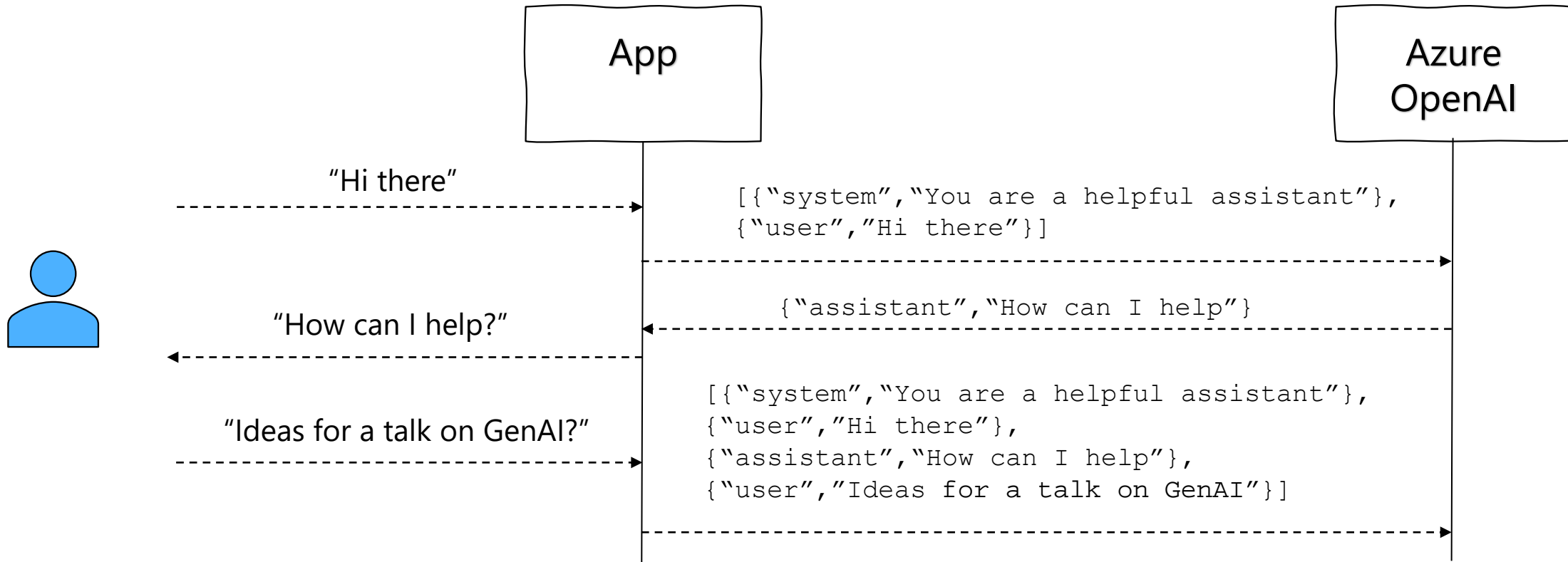
Environment

- AzureOpenAI
- Web Api
- VS Project

Basics

- Service
 - AzureOpenAI, OpenAI
 - Client (Options)
- Chat
- Role
 - System
 - User
 - Assistant
 - Tool
 - Functions
- Request Message
 - Include Role, Content
 - Must include all context (including chat history)
- Response Message
 - Response Content
 - Other indications

Chat Conversation



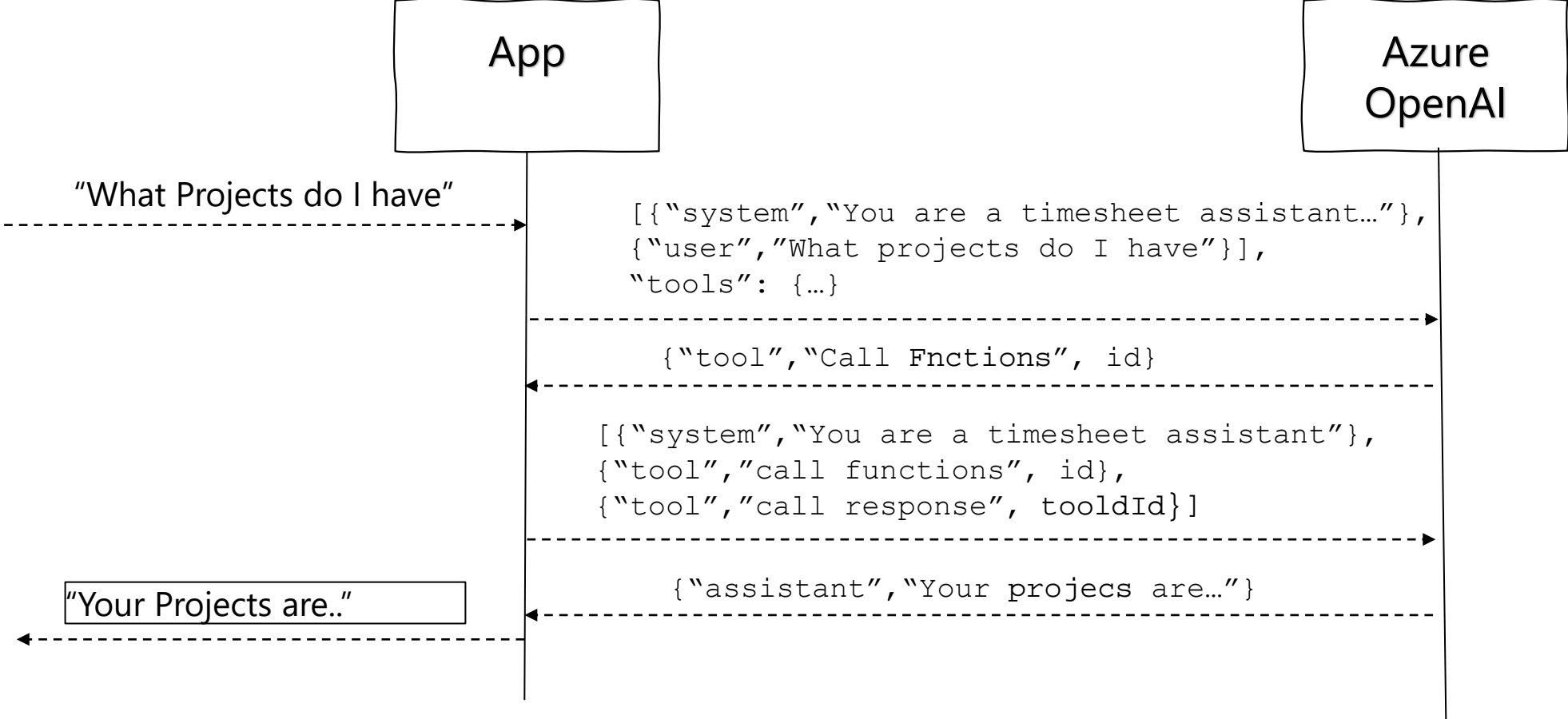


Demo!

Tools

- Give Assistants access to tools like Code Interpreter and Knowledge Retrieval
- Allows you to build your own tools using Function calling

Chat Conversation





Demo!

Lessons Learned

- API and SDKs are changing quickly
- SDK and Model Versions make a difference
- Optimizing system prompts and function definition is crucial. It will save you \$!
- Phyton > C# (at least for this)

References

- <https://github.com/Azure/azure-sdk-for-net/tree/main/sdk/openai/Azure.AI.OpenAI>
- <https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/function-calling>
- <https://learn.microsoft.com/en-gb/azure/ai-services/openai/concepts/assistants>



Thanks!