(Daytime, In-person, Hands-On)
AI Workshop in BURLINGTON MA

Fri Dec 6 @ 9:30-4:00

Workshop Learning Objectives

1. Understanding role and purpose of Semantic Kernel
2. Basics of setting up and using Semantic Kernel
3. Prompting techniques for your application to interact with LLMs
4. How to implement RAG and use memory
5. How to create and manage single and multi-agent systems

Workshop Outline

Understanding Semantic Kernel and how to use it

- Overview of the Semantic Kernel project
- Add it to an application
- Chat with an LLM
- Add chat history for better context

Retrieval Augmented Generation (RAG)

- Overview of RAG
- What is function calling
- Create functions/plugins in Sematic Kernel
- Add memory using a persistent data store
- Ways to improve RAG

Agent framework

- Overview of the Semantic Kernel Agent framework
- Creating your first agent
- Multiple agent orchestration

Examples and Labs ~~mostly~~ in C# and using Azure OpenAI.

WELCOME & AGENDA

1. Starting with the Prompt – AI basics with Semantic Kernel (lab instructor: Bill Wilder)
2. Applying RAG & Vector patterns (lab instructor: Jason Haley)
3. Creating AI Agents (lab instructor: Juan Pablo Garcia Gonzalez)

```
gh repo clone
  bostonazure/rag-vector-agent-semantic-kernel
```

WELCOME

About the day

About the building

About the food

TODAY’s CREDENTIALS
Azure OpenAI, SQL DB, and Bing Search

Mac, Linux, WSL users update their appsettings.Local.json file thusly:

```
curl -L -o appsettings.Local.json https://bit.ly/workshop-banana-2024
```

And the PowerShell crowd can do:

```
Invoke-WebRequest -Uri "https://bit.ly/workshop-banana-2024" -OutFile "appsettings.Local.json"
```

All creds (incl .NET secret manager). But there are steps!

AI Workshop

Getting hands on with Semantic Kernel in C#

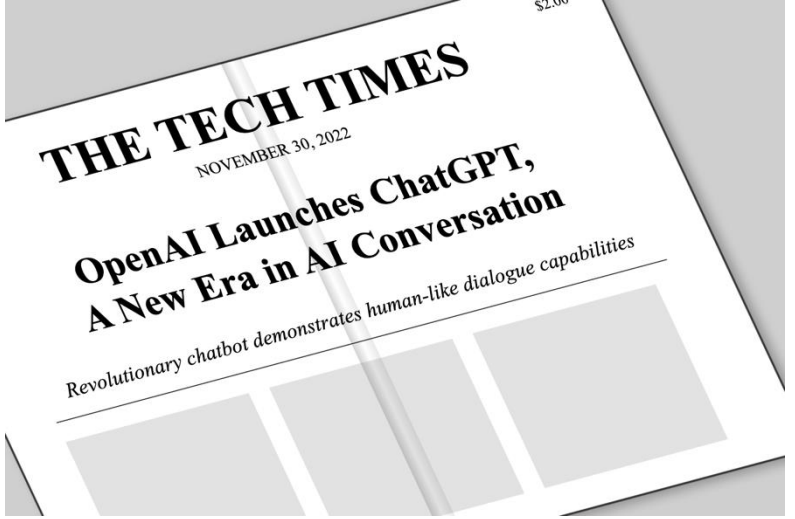
06-Dec-2024

Bill Wilder
(one of three co-presenters)

linkedin.com/in/billwilder
blog.codingoutloud.com
@codingoutloud
bill@semantickernel.dev

billw@devpartners.com

New superpowers: Nov 2022



We all now have at our command capabilities that have not been possible before in the history of computers

Today we'll look at ways to put it to work.

Semantic What?

- Semantic Kernel is a set of SDKs for interacting with AI (LLM) services
- More portable/flexible than the LLM- or vendor-specific SDKs (like OpenAI)
- Orchestrates your interation
- Supports Plugins & Functions (units of functionality made up of your prompts, your code)
- Supports “chatbot” style calling
- Building towards some more advanced topics today – RAG, Vectors, and Agent – that we’ll come to later

https://www.nuget.org/packages/Microsoft.SemanticKernel/

https://github.com/microsoft/semantic-kernel

Enterprise Scale

.NET 1.0

Python 1.0

Java 1.0

Python: the language for ML and AI modeling.

C# and Java: the languages for enterprise software. And enterprise AI.

Why C# & .NET in a Pythonic world?

- Why **NOT** C#/.NET (and Java/JVM)?

- Great platforms and ecosystems…
- **Idiomatic .NET**, DI, Aspire-ready, .NET tools, …
- Observable including OpenTelemetry (“OTel”) support
- Azure OpenAI, OpenAI, most other LLMs on Azure or available via network from SK

- SK very much *reduces* incentives to leave C# & .NET

Basic features are not so hard

Around 10 lines of meaningful code needed to interact with the AI service.

NOT so complex.

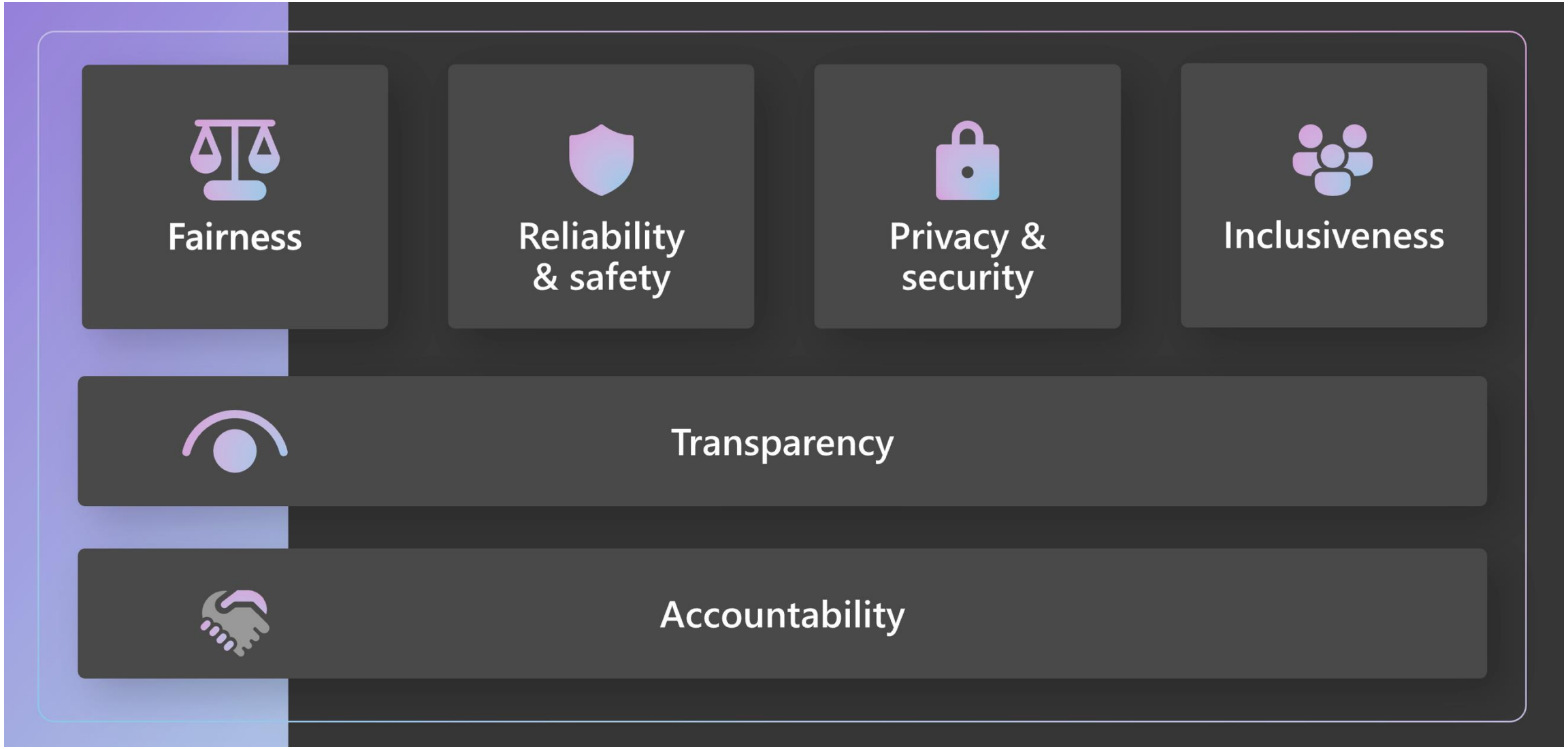
Today we'll look also look at more complex scenarios.

For your further exploration

Prompty.ai

1. https://prompty.ai/
2. https://github.com/microsoft/prompty
3. Extension
4. And more…

Microsoft's Responsible AI Principles



Safety first!

1. Prompt Shields

https://learn.microsoft.com/en-us/azure/ai-services/content-safety/concepts/jailbreak-detection

2. OWASP TOP 10 for LLMs

https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

GitHub Copilot

1. See **High-Level Summary of Labs**

https://github.com/bostonazure/rag-vector-agent-semantic-kernel?tab=readme-ov-file#high-level-summary-of-labs

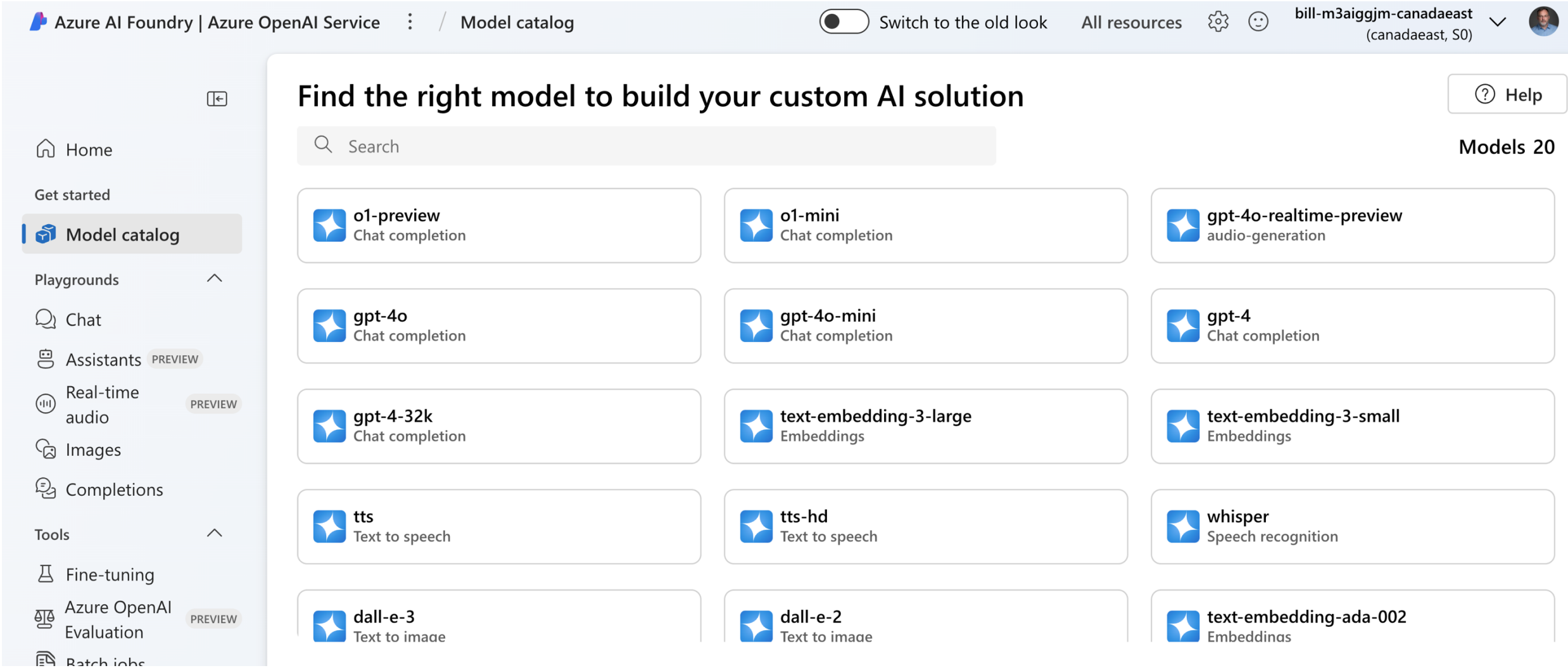
2. Install VS Code / VS extension

Dramatis Personae

ACTORS in this AI PRODUCTION

1. Large Language Model (LLM) – hosted services
   - API key, model name, endpoint (domain name)
   - Stateless (how is chat possible?)
   - Non-deterministic (can Temperature and Top-p fix that?)

Azure AI Foundry – our MaaS host!



ACTORS in this AI PRODUCTION

2. Tokens
   - “a syllable”
   - Prediction unit
   - Billing unit

Tokens

Tokens	Characters
11	43

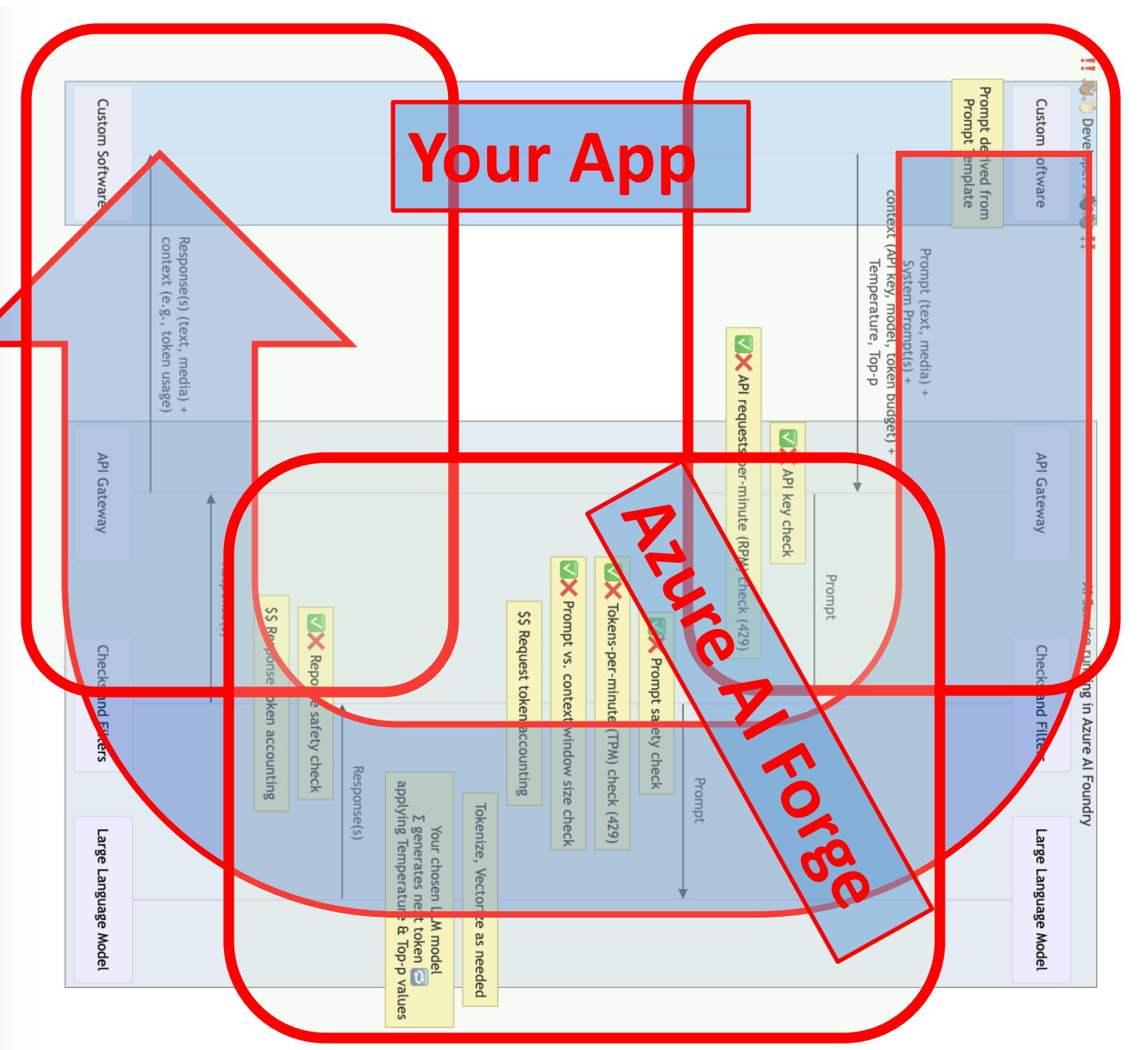
We need to stop anthropomorphizing ChatGPT.

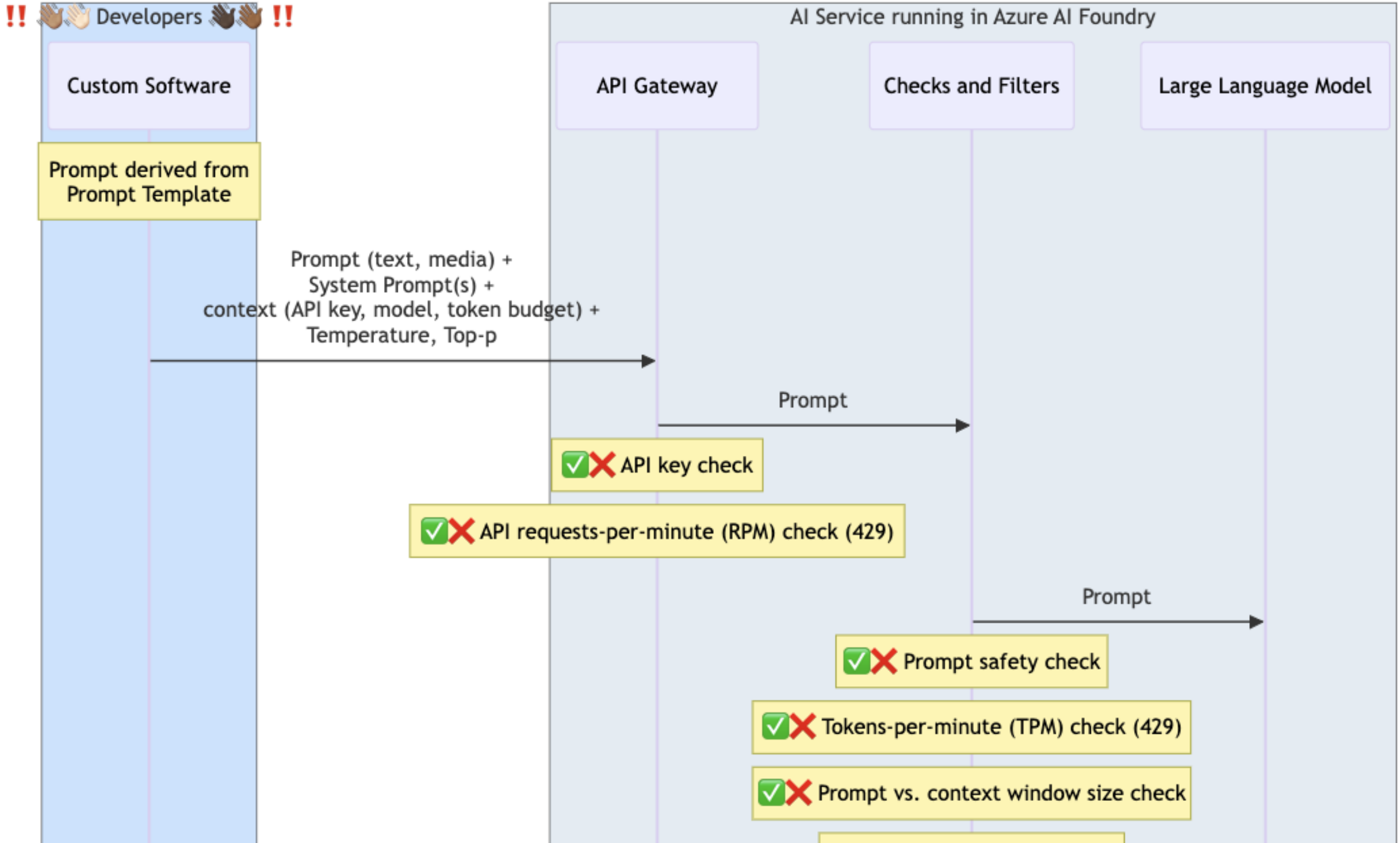

in
out
We need to stop
We need to stop anthrop
We need to stop anthropomorph
We need to stop anthropomorphizing
We need to stop anthropomorphizing Chat
We need to stop anthropomorphizing ChatG
We need to stop anthropomorphizing ChatGPT
We need to stop anthropomorphizing ChatGPT.

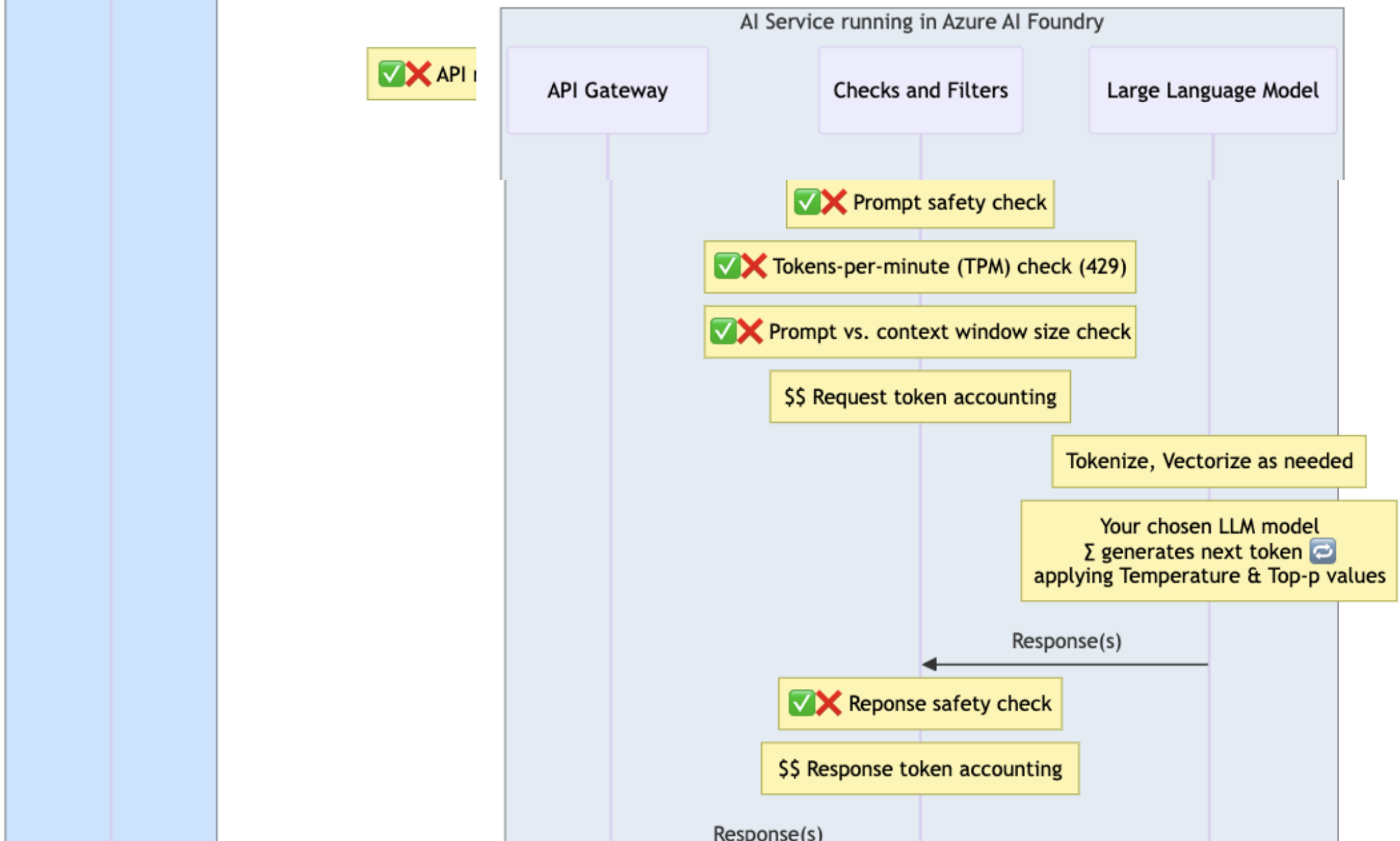
ACTORS in this AI PRODUCTION

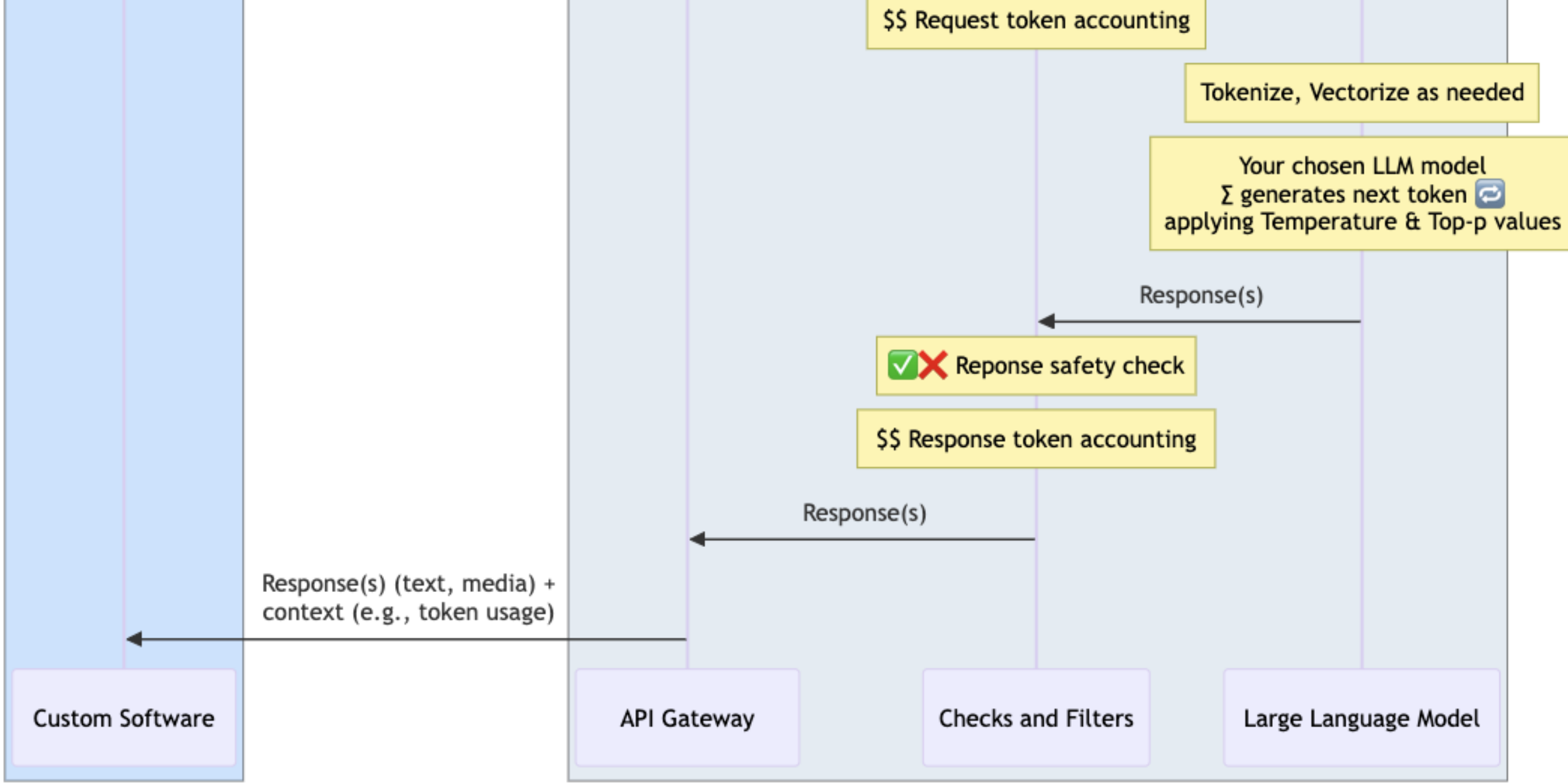
3. Prompts
   - Text
   - Media
   - Extra Grounding or Contextual Data (RAG, Native Functions)

~~Hero's~~ The Prompt's Journey




!! Developers !!
Custom Software
AI Service running in Azure AI Foundry
API Gateway
Checks and Filters
Large Language Model
Prompt derived from Prompt Template
Prompt (text, media) +
System Prompt(s) +
context (API key, model, token budget) +
Temperature, Top-p
Prompt
✅❌ API key check
✅❌ API requests-per-minute (RPM) check (429)
Prompt
✅❌ Prompt safety check
✅❌ Tokens-per-minute (TPM) check (429)
✅❌ Prompt vs. context window size check


AI Service running in Azure AI Foundry
API
API Gateway
Checks and Filters
Large Language Model
Prompt safety check
Tokens-per-minute (TPM) check (429)
Prompt vs. context window size check
$$ Request token accounting
Tokenize, Vectorize as needed
Your chosen LLM model
Σ generates next token
applying Temperature & Top-p values
Response(s)
Reponse safety check
$$ Response token accounting
Response(s)


$$ Request token accounting
Tokenize, Vectorize as needed
Your chosen LLM model
Σ generates next token 🔁
applying Temperature & Top-p values
Response(s)
✅❌ Reponse safety check
$$ Response token accounting
Response(s)
Response(s) (text, media) +
context (e.g., token usage)
Custom Software
API Gateway
Checks and Filters
Large Language Model

Let the Labs Begin!

Find us at **github.com/bostonazure**

```
gh repo clone
bostonazure/rag-vector-
agent-semantic-kernel
```

Lab 0, Lab 1, then break

Lab 0: Can we just access the dang API?

- Focus: Accessing APIs and running a simple SK console app.
- Objectives: Get local copies of API keys, run a simple SK console app.

Lab 1: Getting Started with Semantic Kernel

- Focus: Adding Semantic Kernel to an application, using Azure OpenAI, and creating prompt functions.
- Objectives: Demonstrate how to add Semantic Kernel to an existing application, use Semantic Kernel to chat with the Azure OpenAI LLM, define a prompt function and use it in an application, recognize the need for chat history and how to add it.
- Additional Exercises: Experiment with different Temperature values to see their influence.

Lab 2

Lab 2: Creating Semantic Kernel Plugins

- Focus: Creating native plugins and using web search plugins.
- Objectives: Implement a plugin with native C# code, use a plugin to give an LLM additional information, create a plugin that uses an LLM to rewrite a user query, utilize a Semantic Kernel plugin to perform a web search.
- Additional Exercises: Experiment with different plugin functions.

Like in Hollywood,
don’t call us, we’ll call you!

https://bit.ly/autoinvokefunc

Questions?

See you at Boston Azure
bostonazure.org
@bostonazure

Bill Wilder
@codingoutloud
Dec 6:semantickernel.dev
blog.codingoutloud.com
linkedin.com/in/billwilder