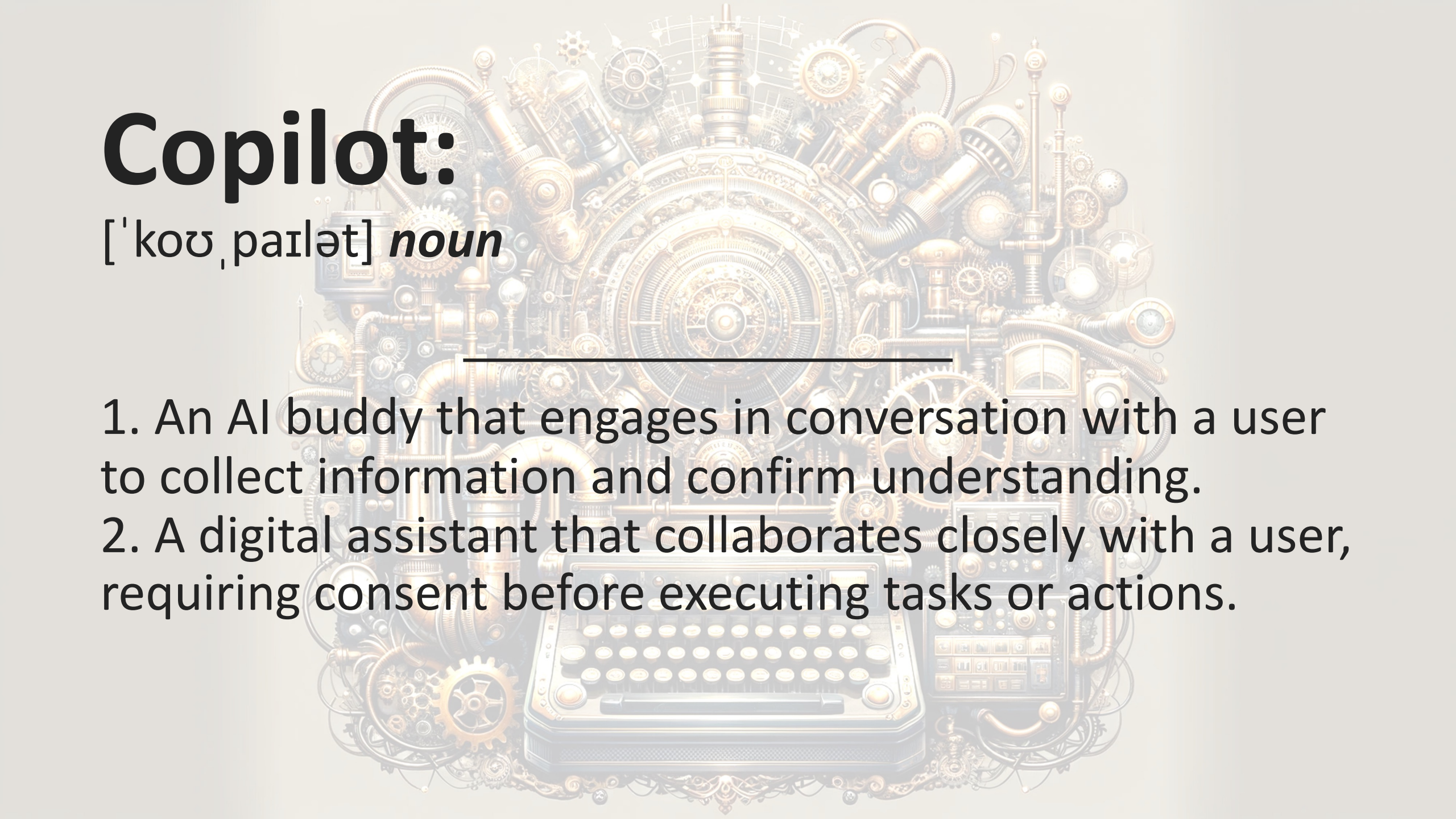

Building a Copilot for Your Own Application with Semantic Kernel

Pieter Nijs

- Consultant, Mobile Dev. Expert & AI @ Xebia
- The MVVM Pattern in .NET MAUI (Packt)
- Active Community Member
- Blog.PieEatingNinjas.be
- Microsoft MVP



Copilot:

[ˈkoʊˌpaɪlət] ***noun***

---

1. An AI buddy that engages in conversation with a user to collect information and confirm understanding.
2. A digital assistant that collaborates closely with a user, requiring consent before executing tasks or actions.

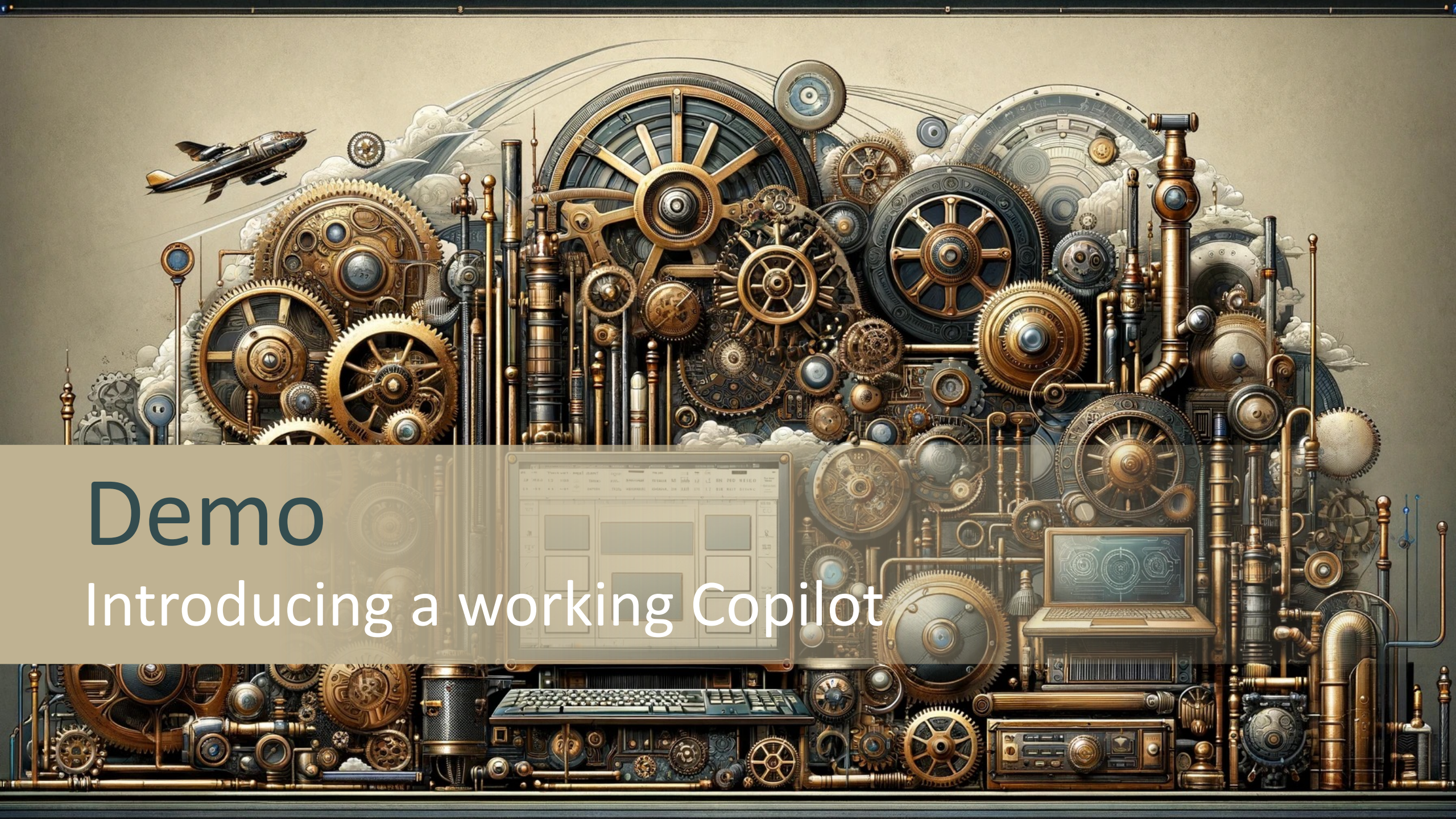
Going Beyond
RAG

RAG = enhancing responses with information retrieval

RAG can be part of a Copilot

Our Copilot focusses on

- Conversational AI, engaging with user
- Gather information
- Clarify intents
- Seek permission before taking action



Demo

Introducing a working Copilot

Semantic Kernel

Microsoft SDK

- Open-Source
- C#, Python, Java
- Extensible through plugins

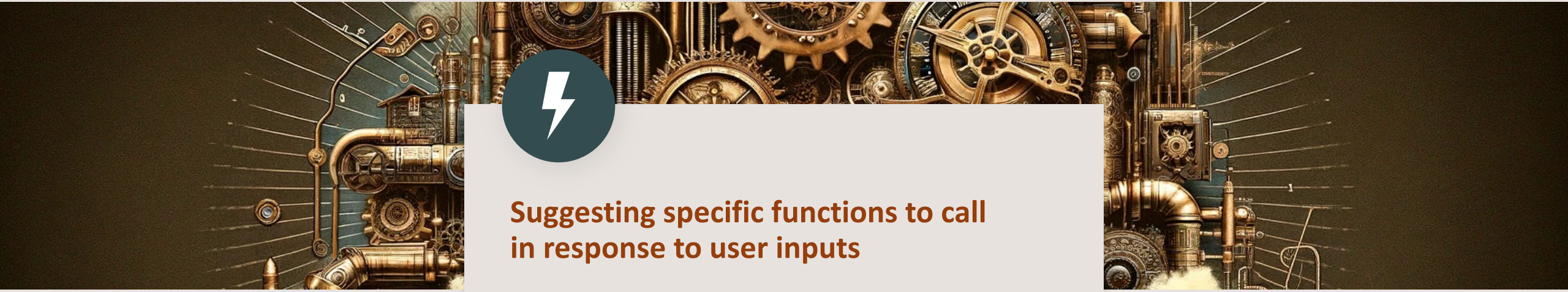
Integration with Leading AI Models

- OpenAI GPT
- Azure OpenAI
- Hugging Face
- Streamline AI capabilities into existing systems

Empowering Goal-Oriented AI Interactions

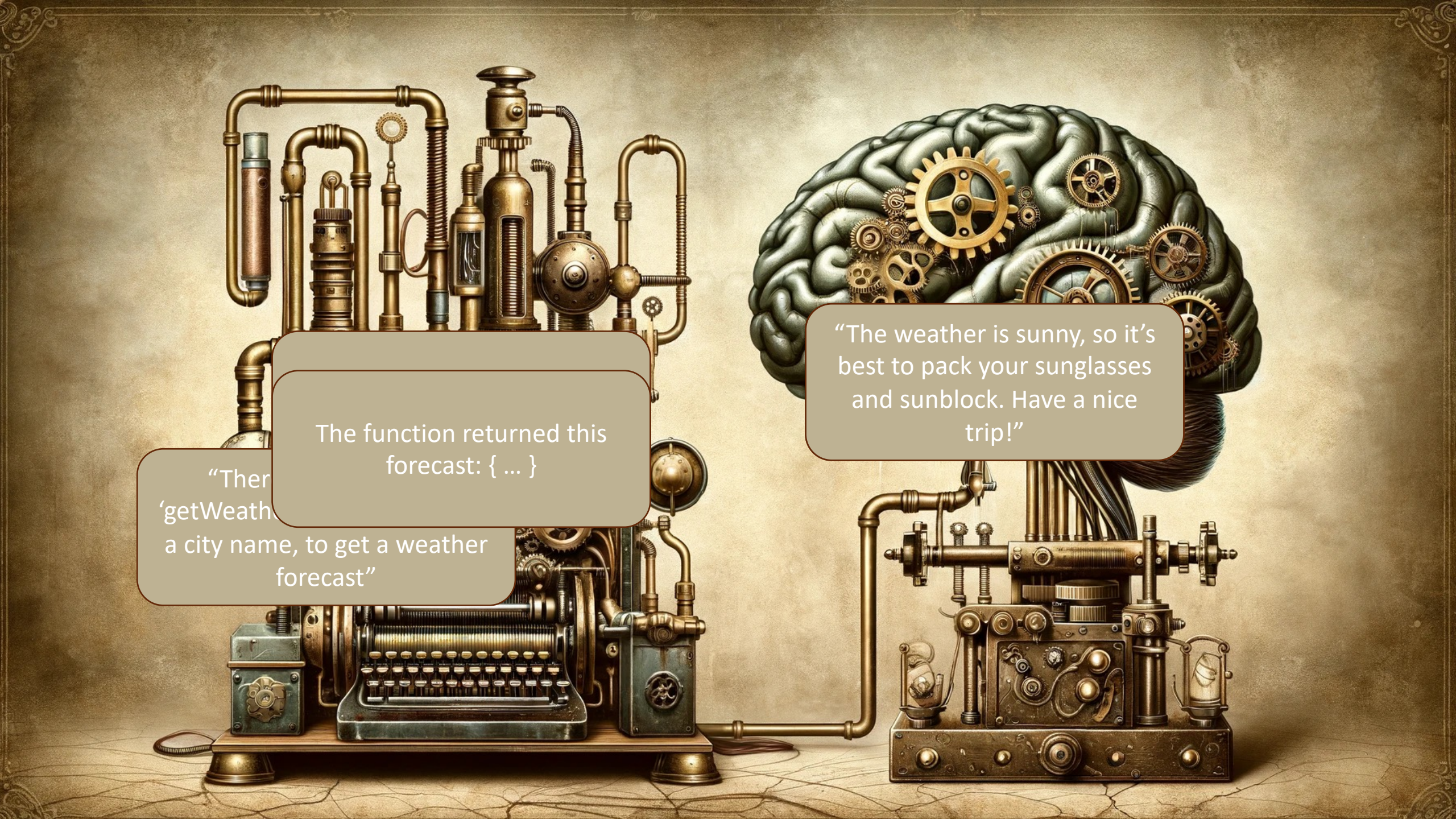
- Orchestrate AI capabilities
- Build intelligent Agents that understand and execute complex workflows

OpenAI Function Calling



Suggesting specific functions to call in response to user inputs

- **Define function:** describe purpose and parameters to the API of the LLM
- **LLM suggests action:** LLM assesses input and returns what function to call with arguments
- **Execute & Respond:** Invoke function and send result back to API of LLM
- >= GPT 3.5 Turbo
- Seemless with Semantic Kernel


“Ther
‘getWeath
a city name, to get a weather forecast”
The function returned this forecast: { … }
“The weather is sunny, so it’s best to pack your sunglasses and sunblock. Have a nice trip!”

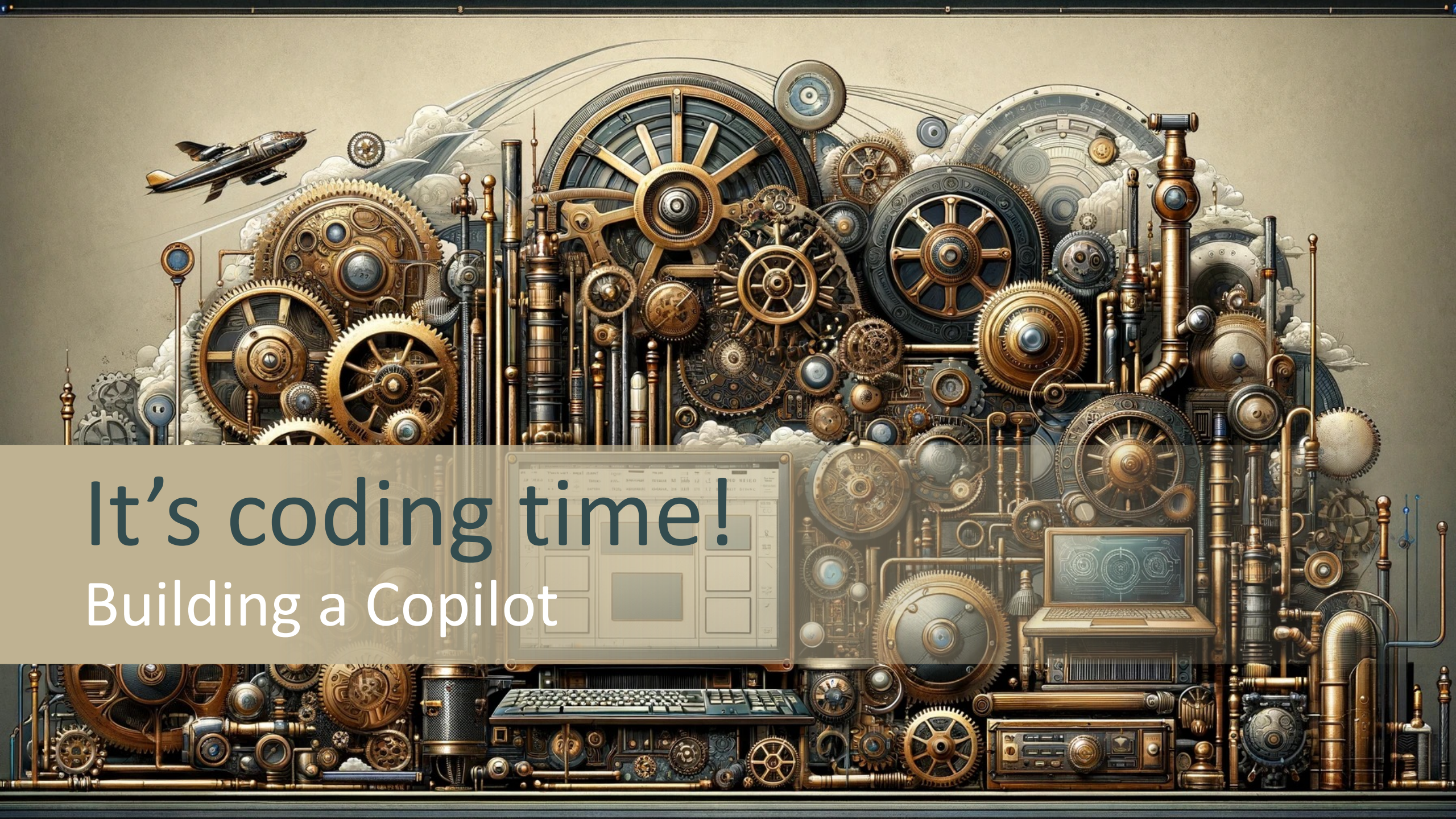
Let’s build this

Our API endpoints are ready to use
No update needed

Define a goal (Prompt Engineering)

Define functions (Plugins, Function Calling)

Orchestration through Semantic Kernel



It’s coding time!

Building a Copilot

Key Takeaways

Getting started is straightforward

- Proof of Concept
- Decision Maker Buy-In
- Familiar tools
- Abstraction through Semantic Kernel
- Community Support

The Principle of Diminishing Returns

- Quick Wins
- Refinement Takes (a lot of) Time
- Increasing complexity with advanced features
- Maintainability

Mindful Resource Management

- Cost of Tokens
- Optimize Interactions
- Monitor Usage


Questions?

Thank you!

More information

- Me | Pieter Nijs
- Work | https://xebia.com/digital-transformation/microsoft-services/
- LinkedIn | https://www.linkedin.com/in/pieter-nijs
- Sources | https://github.com/PieEatingNinjas/copilot-semantickernel/tree/demo
- Blog | https://blog.pieeatingninjas.be


Building a Copilot for Your Own Application with Semantic Kernel