



# Leveraging .NET Expertise for AI with Semantic Kernel



# Andreas Wänqvist

AI Architect

**Voyado**

Microsoft MVP,  
Aurelia Core Team,  
ex Azure

@mobilemancer

[github.com/mobilemancer](https://github.com/mobilemancer)

[linkedin.com/in/awanqvist/](https://linkedin.com/in/awanqvist/)





# The AI Hype Train





**PYTHON**

**PM**


**.NET**

# This session

- Overview of what's available in Azure
- From ML to Gen AI
- Learn Semantic Kernel

# Azure Services

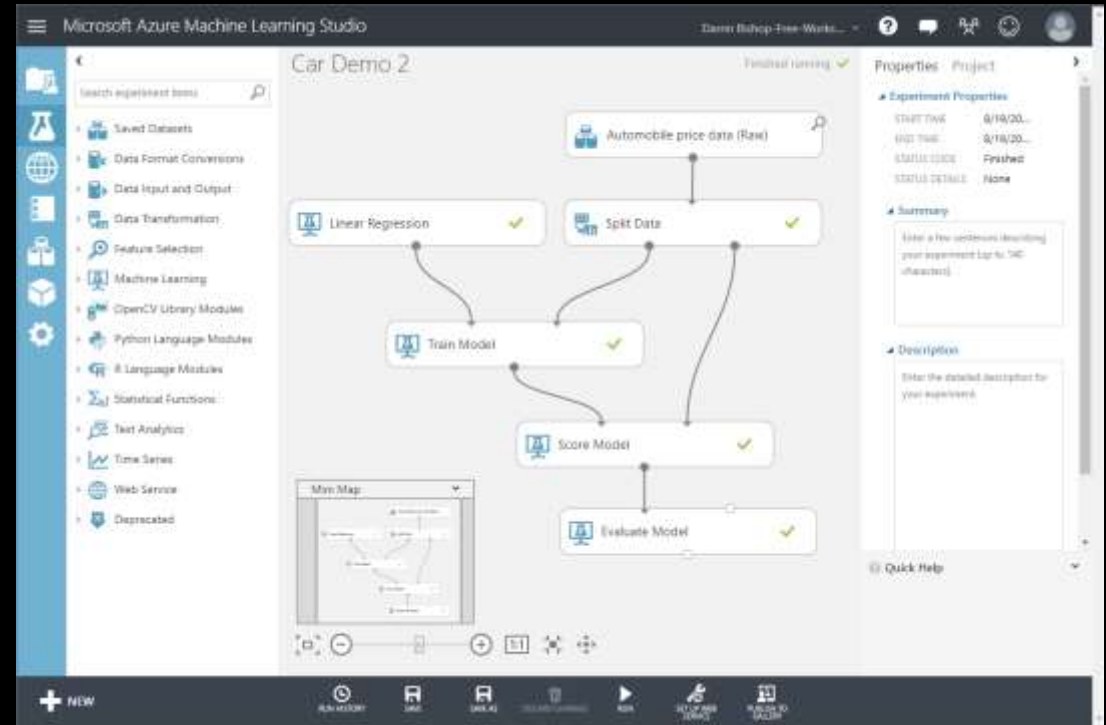
# What's Available in Azure?

- Azure Machine Learning
- Azure AI Services
- Azure AI Search
- Azure AI Studio 



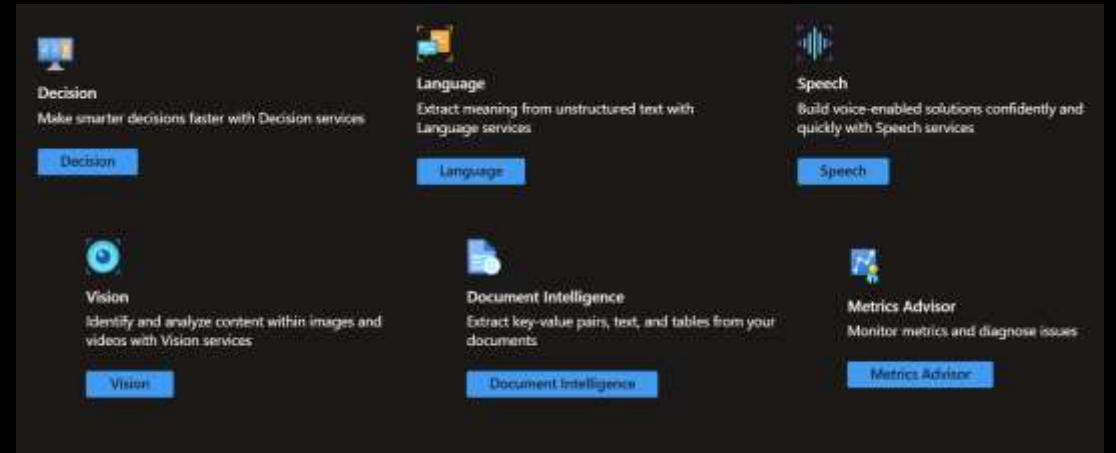
# Azure Machine Learning (2018)

- Develop ML models
- Data Preparation
- Model training and evaluation
- Set up managed endpoints





# Azure AI Services (2019? / 2023)

- Contains the classical Azure AI services
  - Azure AI Translator
  - Azure AI Speech
  - Azure AI Vision
  - Azure AI Search
  - Etc...
- Pre-trained models
- Prompt flow



# Azure AI Search

- Rebranded  *Azure Cognitive Search*
- Similarity search, Multimodal search, Hybrid search (vector + keyword), Multilingual search
- Vector Search, Filtered vector search
-  RAG

# Azure AI Search – Supported Data Sources

- Azure Blob storage
- Azure Table storage
- Azure Data Lake Storage Gen2
- Azure Files (preview)
- Azure Cosmos DB
- Azure SQL Database, SQL Managed Instance, and SQL Server on Azure VMs
- SharePoint in Microsoft 365 (preview)

Gen AI



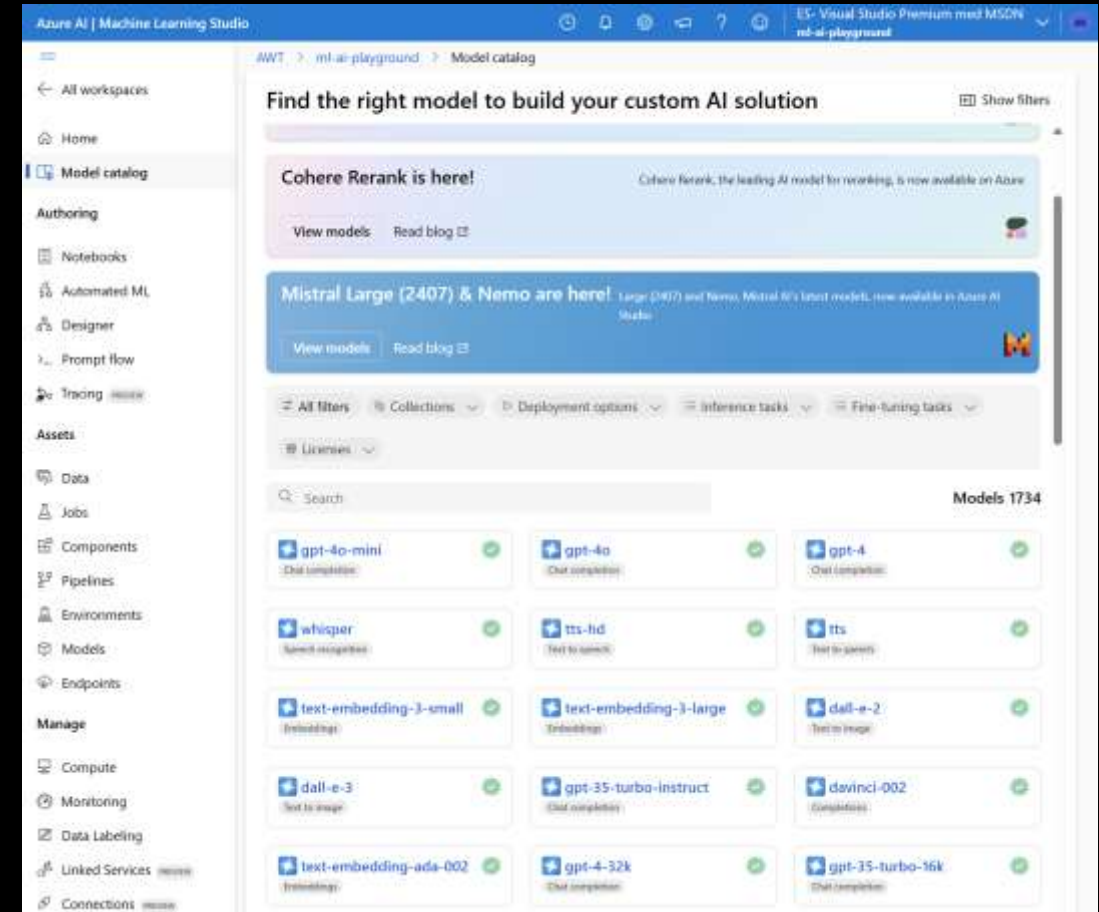
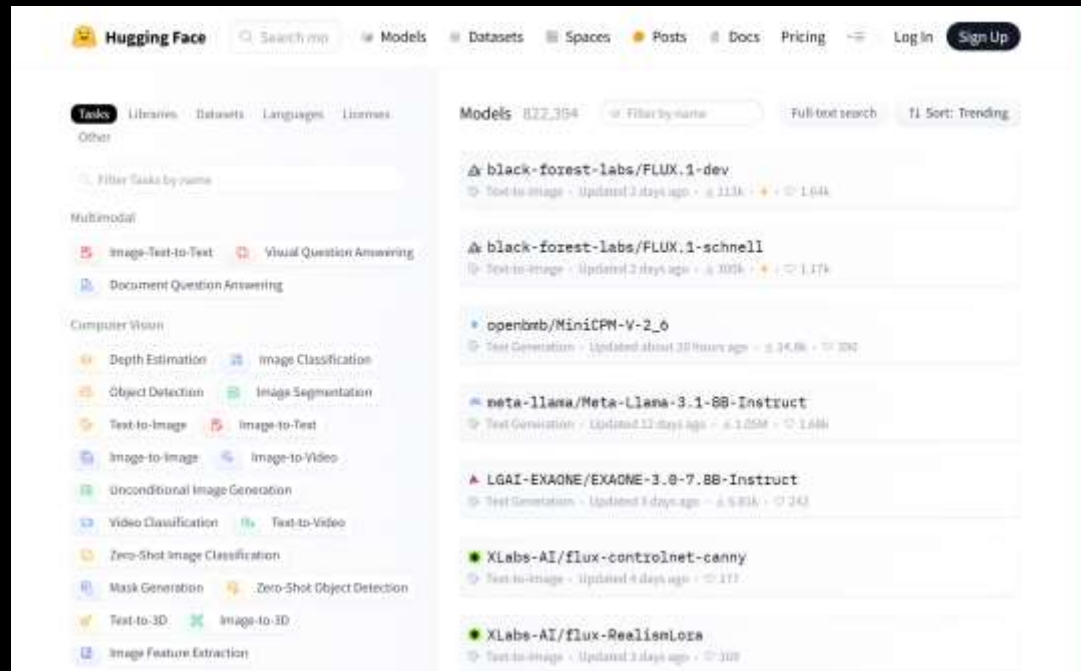
# Generative AI

- 2014 – GAN's
- 2017 – Transformer models
- 2018 – OpenAI produces the first GPT
- 2020 Nov – OpenAI releases ChatGPT (GPT 3.5)

# Models – Open vs Closed

- Llama 3.2 11b, 90b (Meta)
- Mistral (Large, Small, Nemo), Pixtral 12B (Mistral AI)
- GPT 4o, 4o-mini, o1-preview, o1-mini (OpenAI)
- Gemini 1.5 Ultra, Pro, Flash (Google)
- Phi-3 Medium, Small, Mini (Microsoft)

# Models – Open vs Closed



# What is an LLM?



# “Engineering” & Predictability





# Semantic Kernel



# Semantic Kernel

- Lightweight SDK
- Bridge between GPT and other code
- Supports C#, Python, Java
- Reached v1.0 🥳 🎉

⚠️ ...experimentation is still going on  
Some APIs are under experimental flag

# Semantic Kernel

Personas

Plugins

Planners

# Personas

- The prompt that decides how the agent should respond
  - "Meta prompt", "Instruction", "System prompt"
  - Can be inserted into the `ChatHistory` constructor
  - Can be modified with `AddSystemMessage`
- 
- `Use Assistant Tool System`

# Planners

Plans the execution when having multiple plugins

Was prompt based, in the beginning

Handlebars planner, Stepwise planner 

 Function calling (Tools)



# Plugins



Create with

- Native code
- OpenAPI specification
- Logic App

# Plugins

Semantic Kernel leverages function calling, a native feature of the latest LLMs

With function calling, LLMs request particular functions

- Semantic Kernel then marshals those requests

Must be properly semantically described!!!

# Plugins – Native code Attributes



```
[KernelFunction("get_customers")]  
[Description("Gets a list of customers")]  
[return: Description("An array of customers")]  
public async Task<List<Customer>> GetCustomersAsync() { ... }
```

# Plugins – Adding Native Code as Plugins

- Kernel uses builder pattern – inject services & utilize DI
- Add your plugins to the Kernel builder with
  1. `.AddFromObject()` – if you need control over creation
  2. `.AddFromType<>` – uses DI

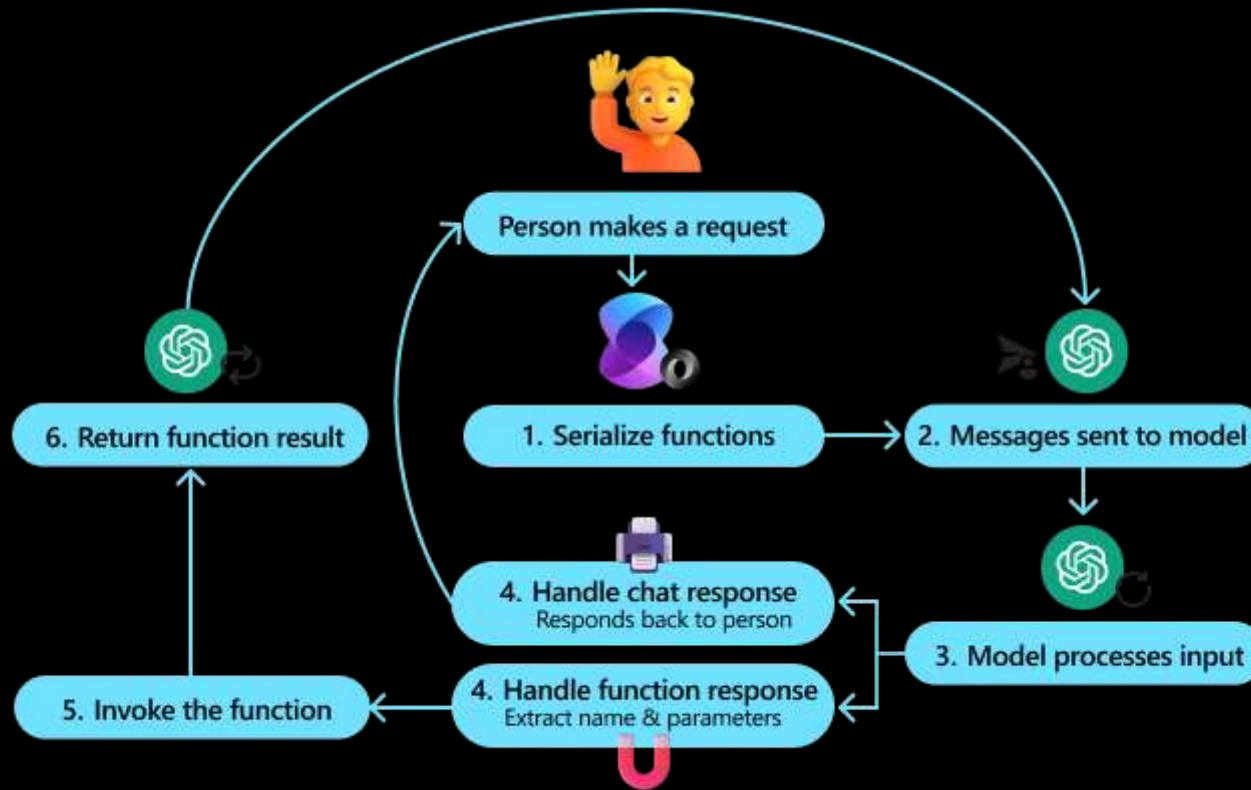
# Plugins – Add to KernelBuilder



```
var builder = new KernelBuilder();  
builder.Plugins.AddFromType<CustomersPlugin>("Customers")  
Kernel kernel = builder.Build();
```



# How Function Calling Works



Agents

# Agents

- What is an agent?

*“Software-based entities that leverage AI and perform work”*

- Chatbot
- CoPilot
- Autonomous

# Kernel Memory

- Service for efficient indexing of datasets
- Supports RAG, synthetic memory, prompt engineering, and custom semantic memory processing
- Available as Web Service, Docker Container, Plugin for ChatGPT/CoPilot/SK, .NET lib for embedded apps

# Semantic Kernel Demo

Going to Production

# Testing – What?

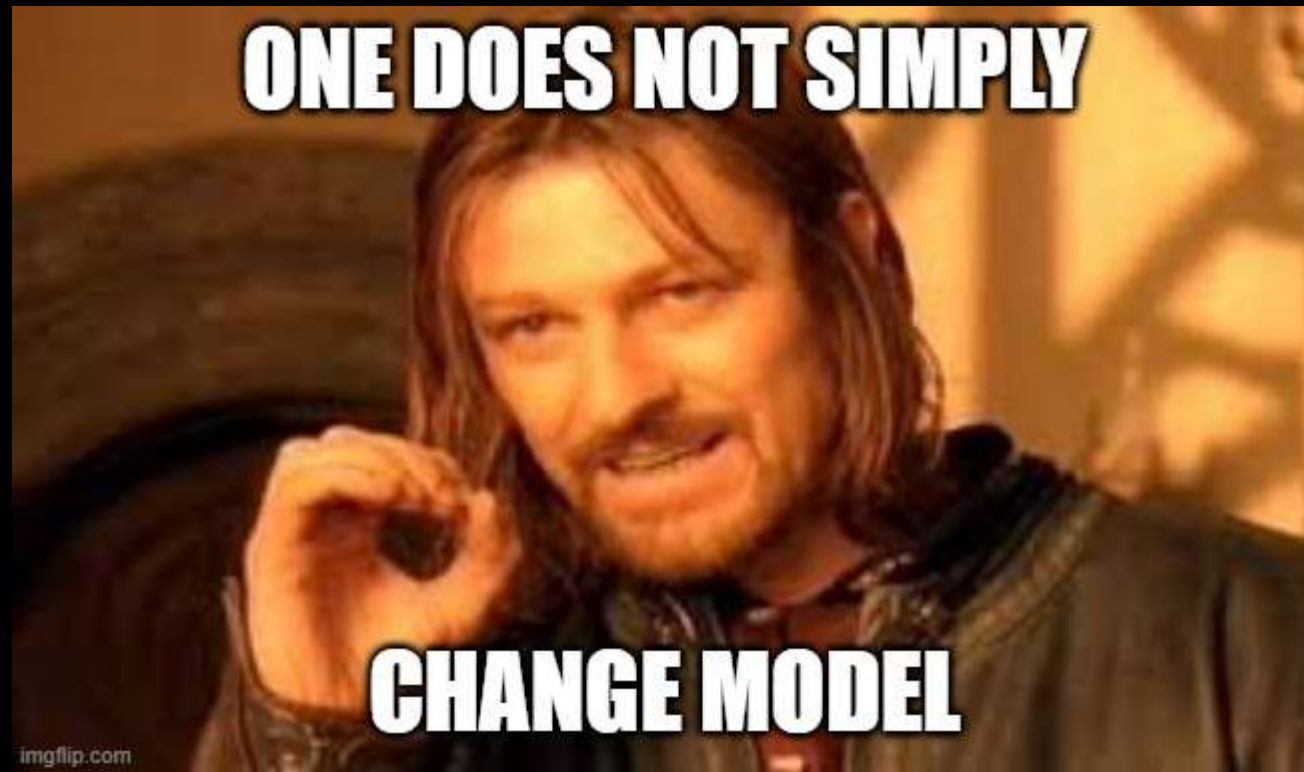
- Relevance
- Coherence
- Correctness

# Testing – How?





# Upgrading Gen AI Systems





[@mobilemancer](#)



[linkedin.com/in/awanqvist/](https://www.linkedin.com/in/awanqvist/)