GETTING STARTED WITH AI FOR .NET DEVS

LET'S GET STARTED

GOALS OF THIS TALK

- I made this talk for me

- Explaining terms/jargon

- High level picture of AI landscape

- 50% theory, 50% practice

- Be able to speak about AI

- Semantic Kernel
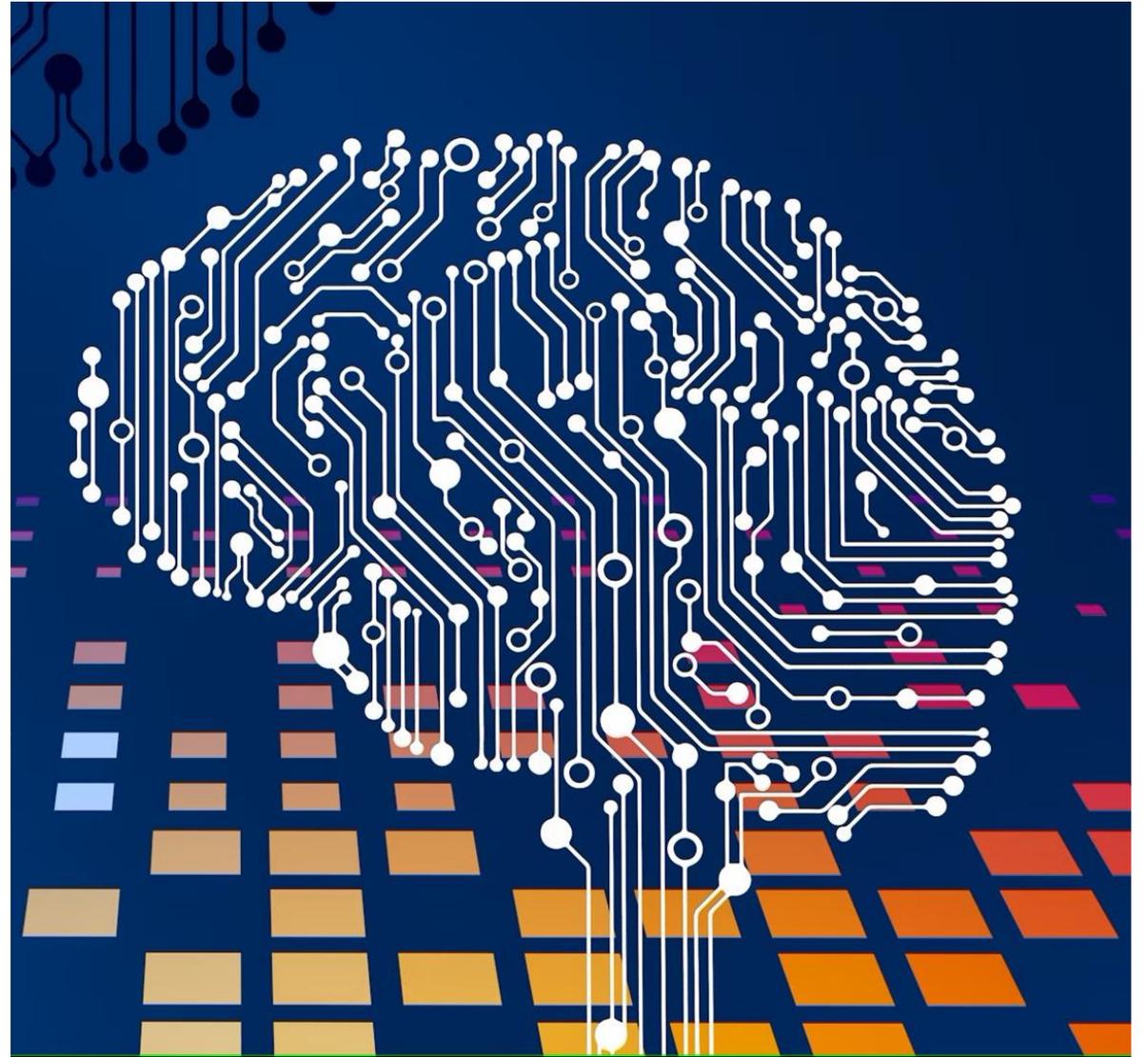
WHAT THIS TALK ISN'T

- In-depth
- Philosophy
- Morals/ethics

WHO AM I?

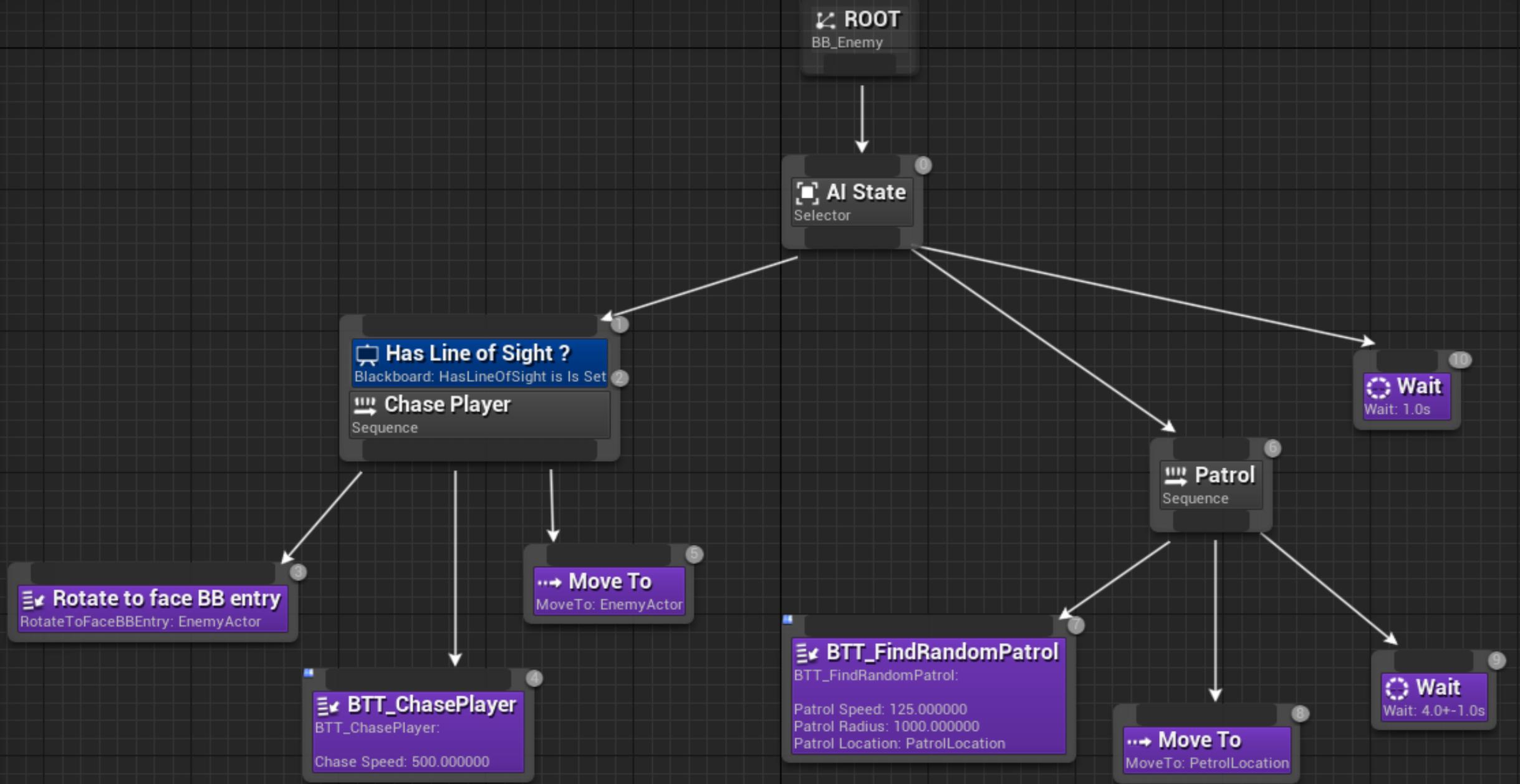- Software developer, consultant
- Microsoft .NET MVP
- Co-organizer of the San Antonio/Austin .NET User Group
- 🌐 daninacan.com
- 𝕏 @danielwarddev
- 💼 daniel-ward-dev
- 🦋 danielwarddev.bsky.social

WHAT IS AI, ANYWAY?
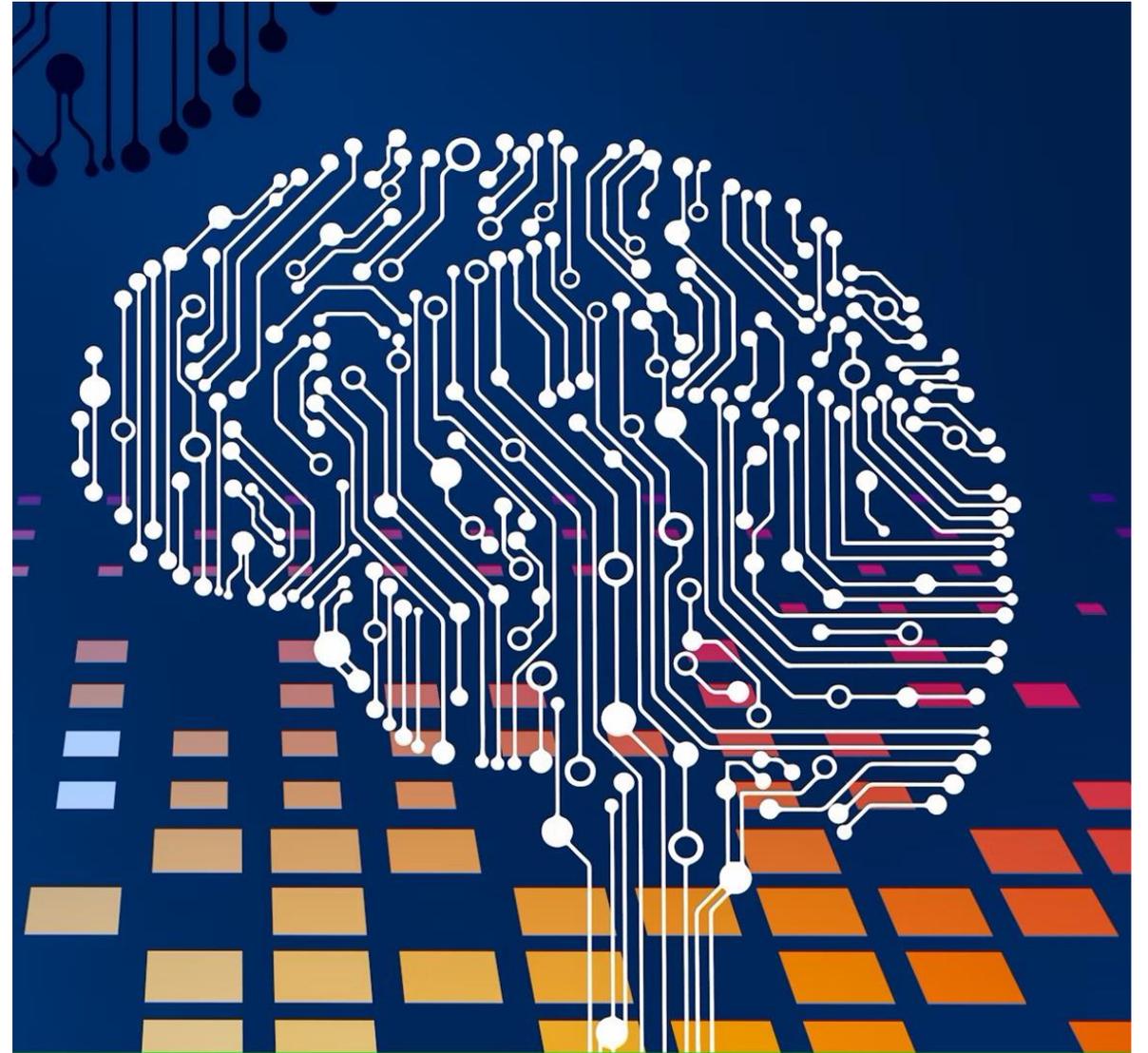
- Simulating human intelligence
- Not new

WHAT IS AI, ANYWAY?

- Simulating human intelligence
- Not new
- ⭐1950: Turing Test
- 1950s: checkers; algebra word problems; English; neural networks
- 2010s – machine learning (ML), deep learning
- 2010s, 20s – transformers, large language models (LLMs), generative AI
  - Copilot, ChatGPT, etc.

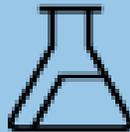# AI VS ML VS DEEP LEARNING VS GENERATIVE AI

Artificial Intelligence

Any technique that enables computers to mimic human intelligence. It includes *machine learning*

Machine Learning

A subset of AI that includes techniques that enable machines to improve at tasks with experience. It includes *deep learning*

Deep Learning

A subset of machine learning based on neural networks that permit a machine to train itself to perform a task.

https://learn.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning?view=azureml-api-2

NEURAL NETWORK

- Loosely based on biological neurons/synapses
- Central part of modern ML
- 3 layers: input, hidden, output
- Each node is a math function
- The more layers, the deeper it is
- output = inputs * weights + bias

https://learn.microsoft.com/en-us/archive/msdn-magazine/2012/may/test-run-dive-into-neural-networks
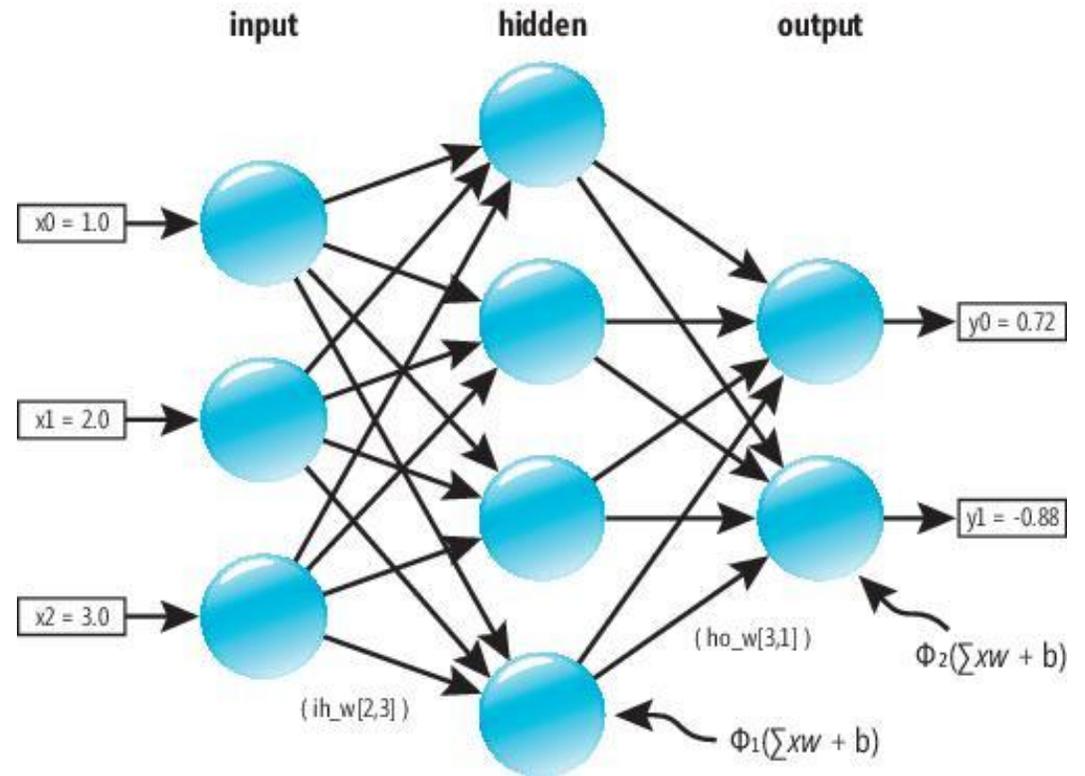
NEURAL NETWORK - WEIGHTS AND BIASES
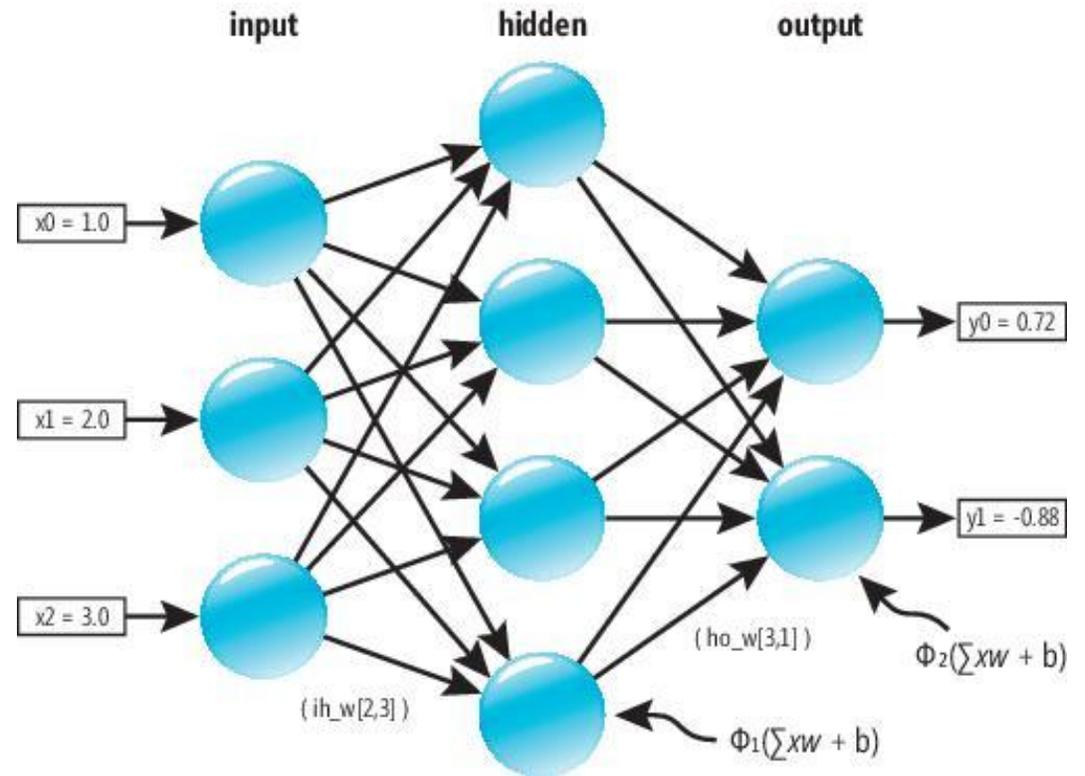
- output = inputs * weights + bias
- Both weights and biases are changed during training
- Weight - strength of a connection between nodes; scales input
- Bias - how "sensitive" a node is - helps it activate even with low inputs

input   hidden   output

x0 = 1.0

x1 = 2.0

x2 = 3.0

y0 = 0.72

y1 = -0.88

(ih_w[2,3])

(ho_w[3,1])

$\Phi_1(\sum xw + b)$

$\Phi_2(\sum xw + b)$

https://learn.microsoft.com/en-us/archive/msdn-magazine/2012/may/test-run-dive-into-neural-networks

MACHINE LEARNING (ML) VS DEEP LEARNING

- Machine learning
  - Subset of AI
  - Specific kind of AI that allows machines to improve over time in their tasks with experience
  - Instead of hardcoding rules, train models to recognize patterns
- Deep learning
  - Subset of ML
  - Uses neural networks
  - Difference is in scale
  - Lots of hidden layers, lots of processing

HOW IS IT POSSIBLE AI CREATES NEW STUFF?

- BBC. (2024, March 20). *NHS AI test spots tiny cancers missed by doctors.* https://www.bbc.com/news/technology-68607059

- MIT Technology Review (2023, December 14). *Google DeepMind used a large language model to solve an unsolved math problem.* https://www.technologyreview.com/2023/12/14/1085318/google-deepmind-large-language-model-solve-unsolvable-math-problem-cap-set/
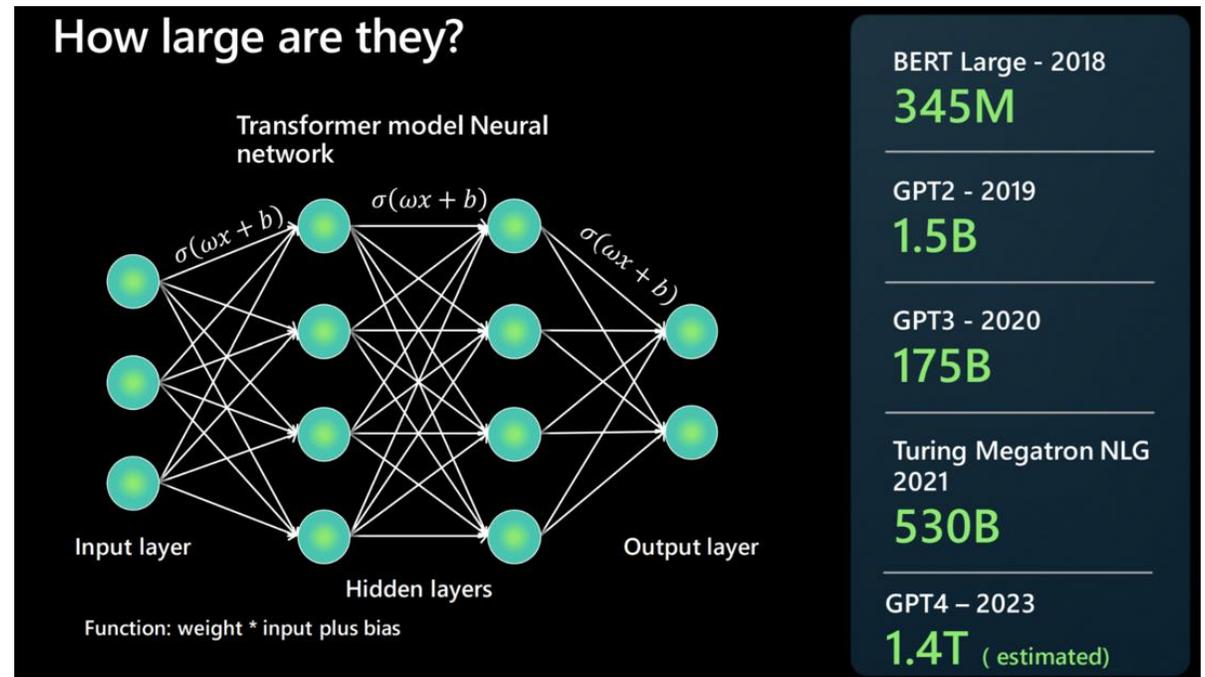
GENERATIVE AI

- Subset of AI
- Uses techniques like deep learning to generate new content
- Depends on a huge amount of pre-trained knowledge
- ChatGPT, GitHub Copilot, etc.

LARGE LANGUAGE MODELS (LLMS)

- Type of ML model designed for natural language processing tasks

https://microsoft.github.io/Workshop-Interact-with-OpenAI-models/llms/

GENERATIVE PRE-TRAINED TRANSFORMER (GPT)

- Generative - generates new content
- Pre-trained - goes through pretraining
- Transformer - specific type of neural network

- GPTs are a kind of LLM

- Takes n tokens as input and produces one as output

1. How does it understand what we ask it and respond?

2. How does it refer back to its own knowledge?
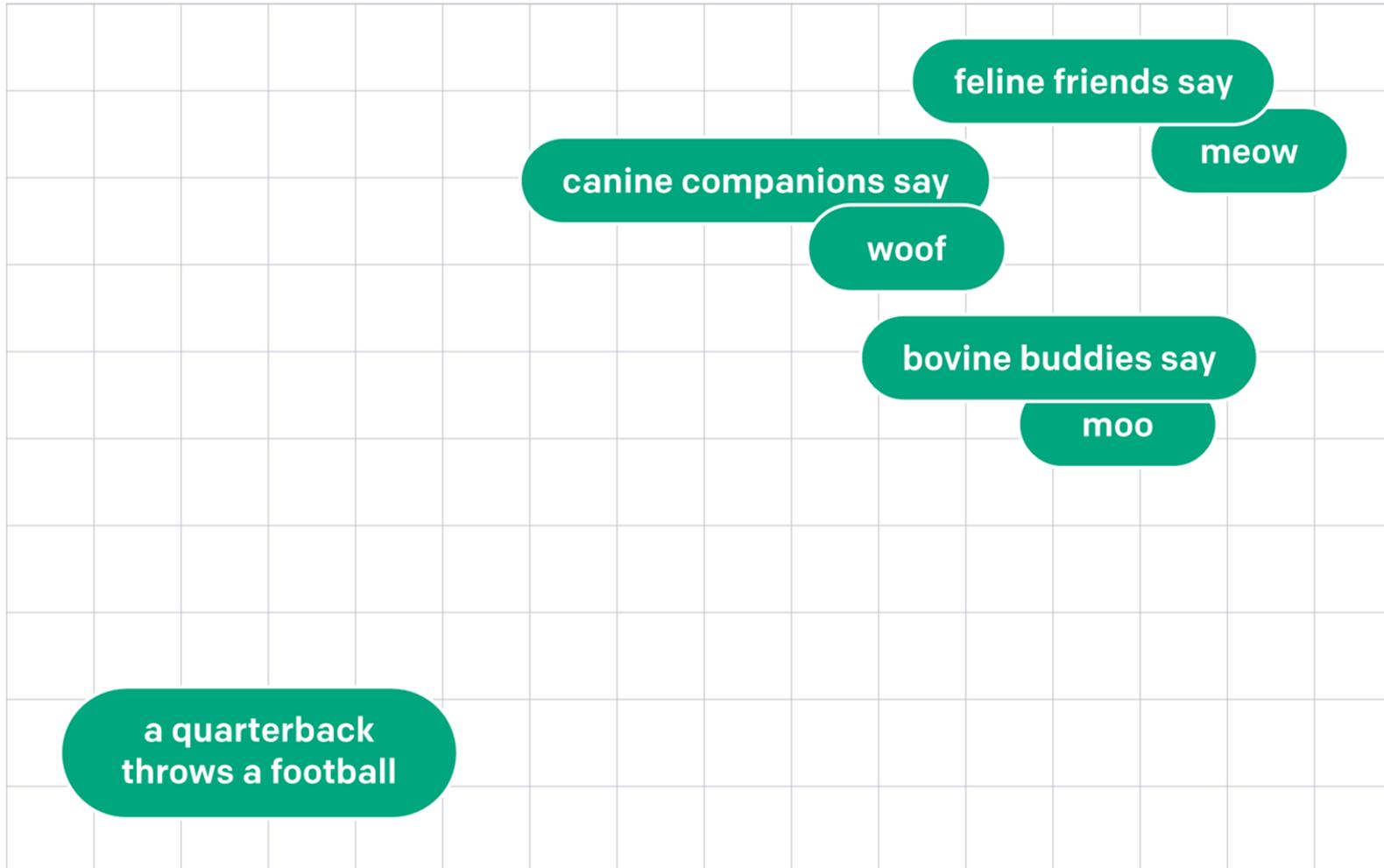
TOKENS

- "Building blocks" of input
- Middle ground between characters and words
- Model keeps a complete list of all possible tokens

Tokens	Characters
9	51

The English language is sometimes incomprehensible.
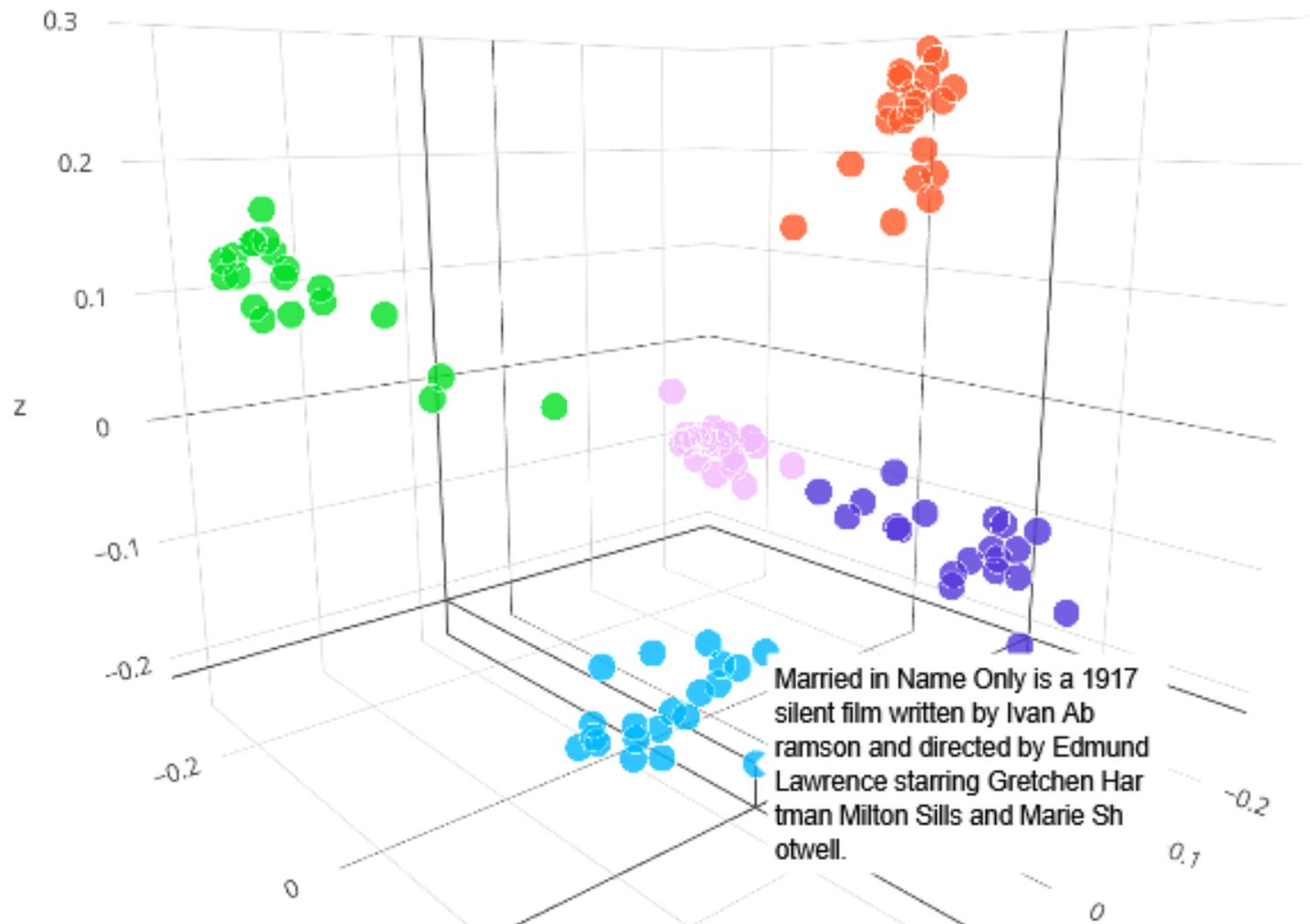
https://platform.openai.com/tokenizer

VECTORS

- Tokens are embedded into vectors
    - Mathematical representation of data
    - "king" -> [0.12, -0.98, 0.55, …, 1.06]
- Semantic similarity
    - "apple" and "orange" are closer than "apple" and "car"
    - Each dimension is a "feature"
- GPT-3 had 12288 dimensions

ATTENTION

- Attention is All You Need, Google, 2017
- "Attention block" – two primary things
  1. **Allowed each token to consider all other tokens**. Tokens can "attend" to any other token, not just ones that come before
  2. **Allowed for parallel calculation**, rather than waiting on all the tokens before it
- "What is the capital of France?"

https://learn.microsoft.com/en-us/azure/search/semantic-search-overview
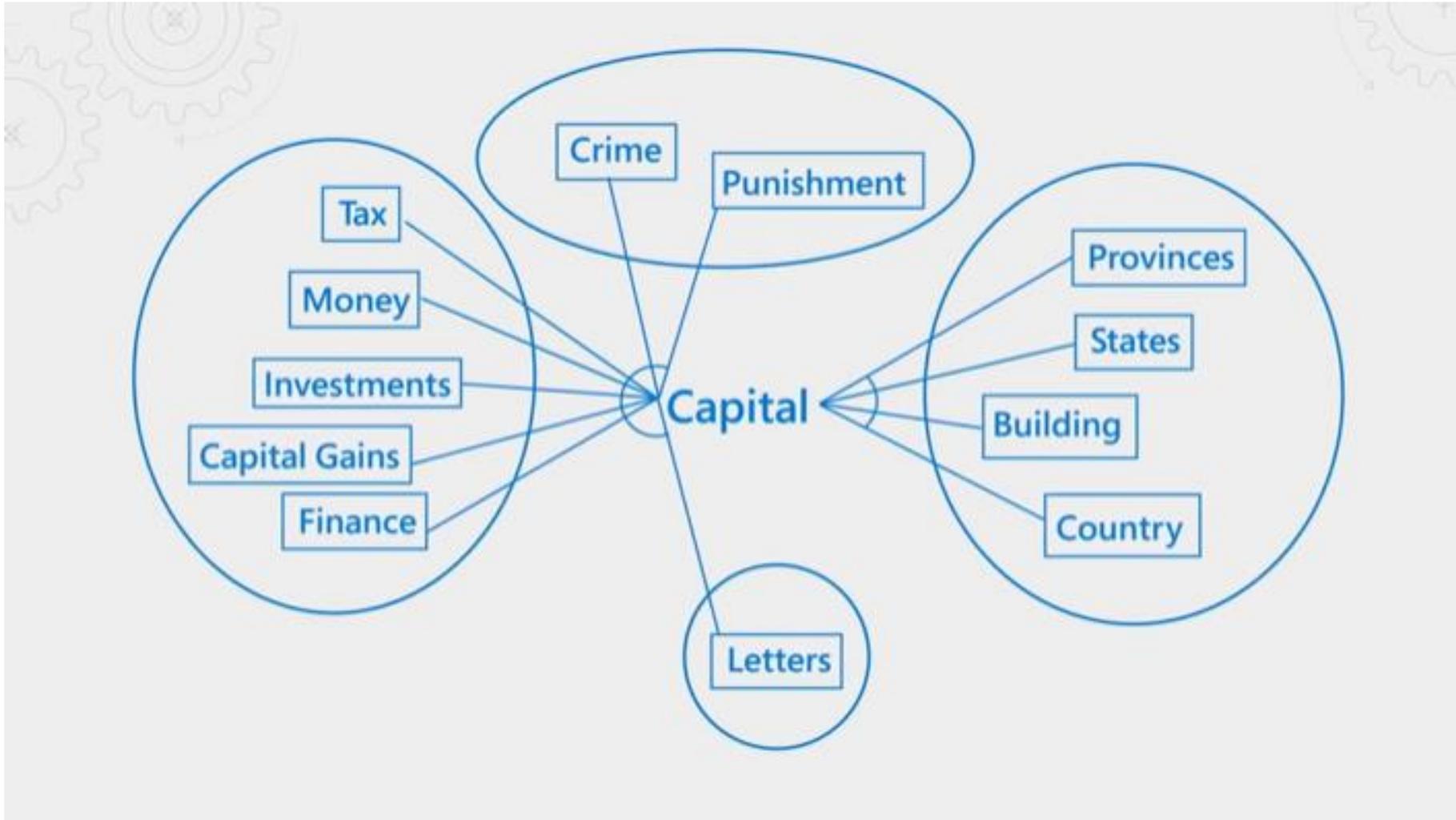
CONTEXT

- The tokens that the LLM is keeping in memory to generate new ones
- "Context window"
- Scales quadratically
- 1k tokens - 1000x1000 matrix - 1M vectors
- 10k tokens - 10000x10000 matrix - 100M vectors
- 100k tokens - 100000x100000 matrix - 10B vectors

TRAINING

- Two phases
  - 1. Pre-training. Give it a massive dataset, gets embedded, finds relationship. Get a **base model**
  - 2. Post-training/fine-tuning. Requires at least some manual intervention by humans. Swap out to a different dataset by humans. Get lots of people to write prompts, as well as an ideal response. Get an **assistant model**
- This makes up the **model**

EXAMPLE: OLLAMA 2:70B

Step 1 – pre-training

1. Get ~10TB of text from the internet
2. Use ~6,000 GPUs ~$2M, and ~12 days to embed this into vectors in a neural network with rough semantic meaning

Step 2 – post-training/fine-tuning

1. Write labeling instructions for humans
2. Hire lots of humans to write ~100k high quality idea Q&A
3. Use these to "test" the model and refine it

PARAMETERS

- Weights and biases = "parameters"
- GPT-4 (supposedly) has 1.7 trillion

- As the model trains, these change

EXAMPLE: CHATGPT

Write me a haiku about C#

`Write me a haiku about C#`

Brackets hold the flow,

LINQ dances with sharp logic—

Types bloom, strong and bold.

`System: You are a helpful assistant willing to help users with a variety of tasks.`

`User: Write me a haiku about C#`
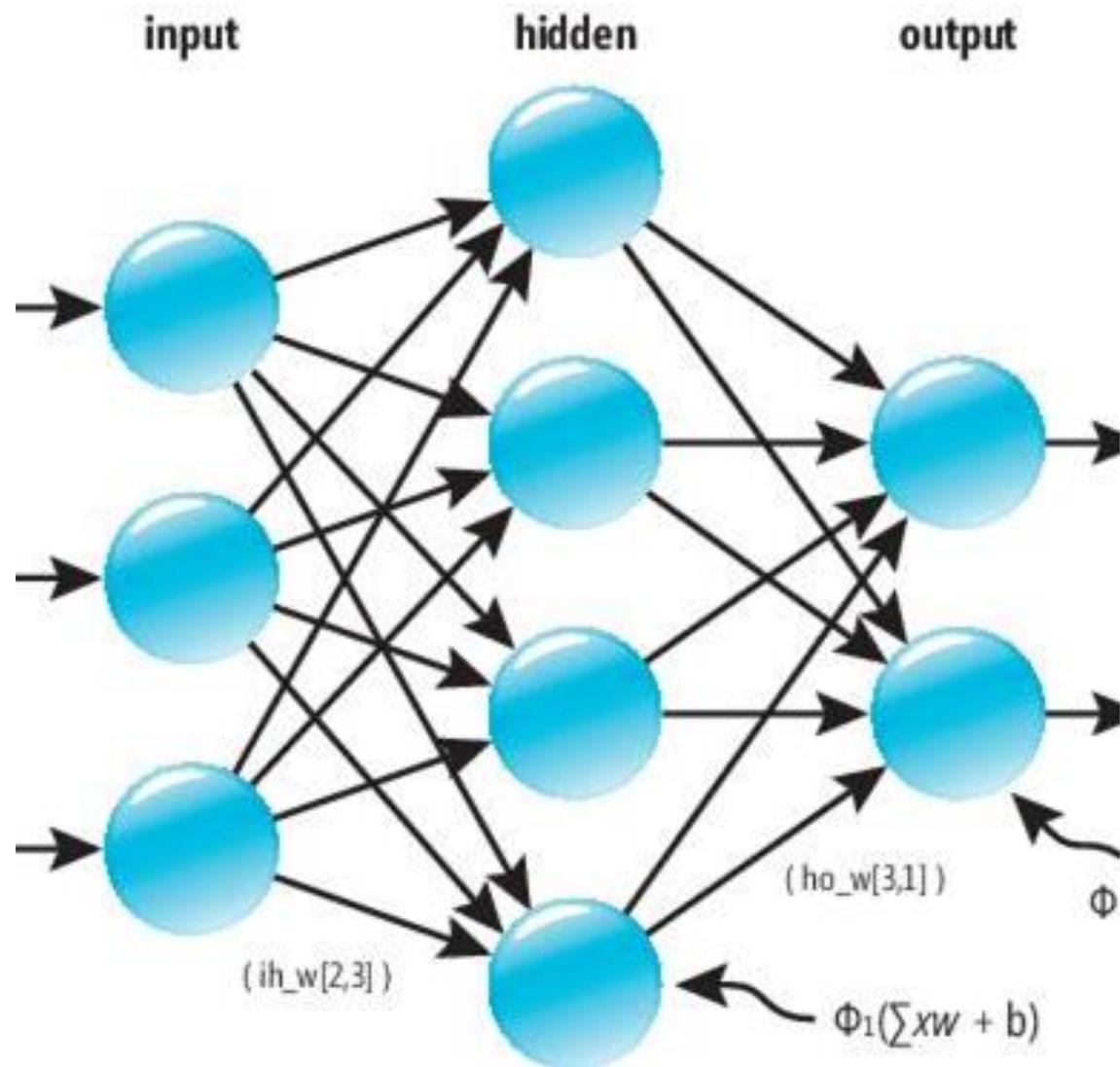
`System:`

EXAMPLE: CHATGPT

Brackets hold the flow,
LINQ dances with sharp logic—
Types bloom, strong and bold.

- Br
- Brackets
- Brackets hold
- Brackets hold the
- etc.

flow	0.8
world	0.05
code	0.02
...	...
dog	0

RETRIEVAL AUGMENTED GENERATION (RAG)

- Augment the response of an LLM with info retrieved from a data store

- Two main phases

1. Retrieval. User prompts something, system retrieves info from an external knowledge store

2. Generation. Retrieved info is used to augment user's prompt. AI model processes both user prompt and retrieved info to produce an enriched response

RAG BENEFITS

- Improved accuracy
- Reduced hallucinations
- Retrieve up-to-date info
- Get domain-specific knowledge

MODEL CONTEXT PROTOCOL (MCP)

- Problem: models are generalized and not connected to our tools

- Solution: MCP is a protocol to allow AIs to connect/interact with external tools, data sources, services, etc. in real time

- Databases, Jira, GitHub, internal APIs, etc.

AGENTS

- "Agentic"
- AI that doesn't just generate, but makes a plan and takes action
- Fulfill wide variety of tasks with some/minimal human intervention
- 3 parts
1. LLM – model used
2. State – context, helps guide decisions based on past results
3. Tools – like the functions before with the convos, a bridge between the model and external systems
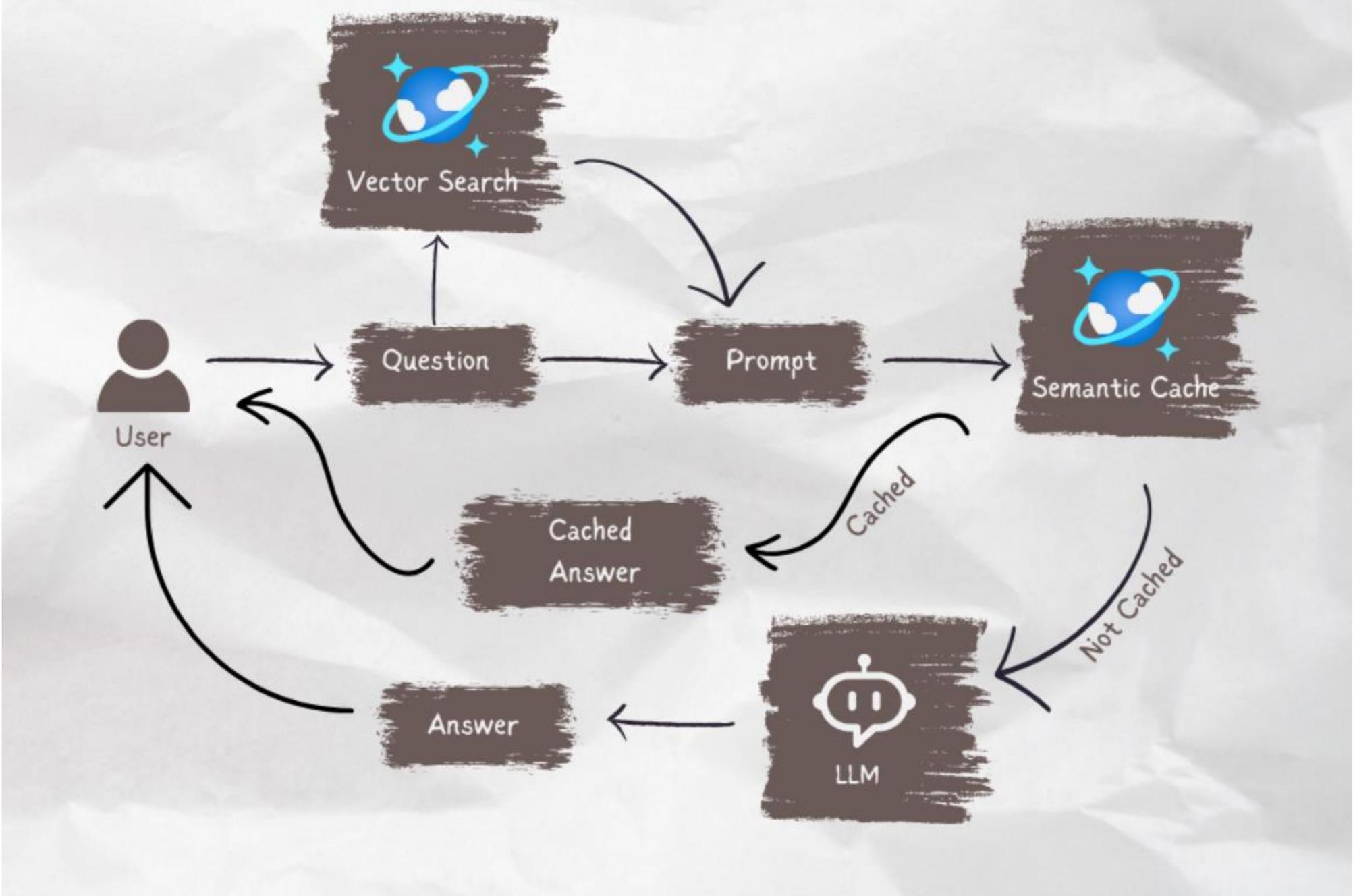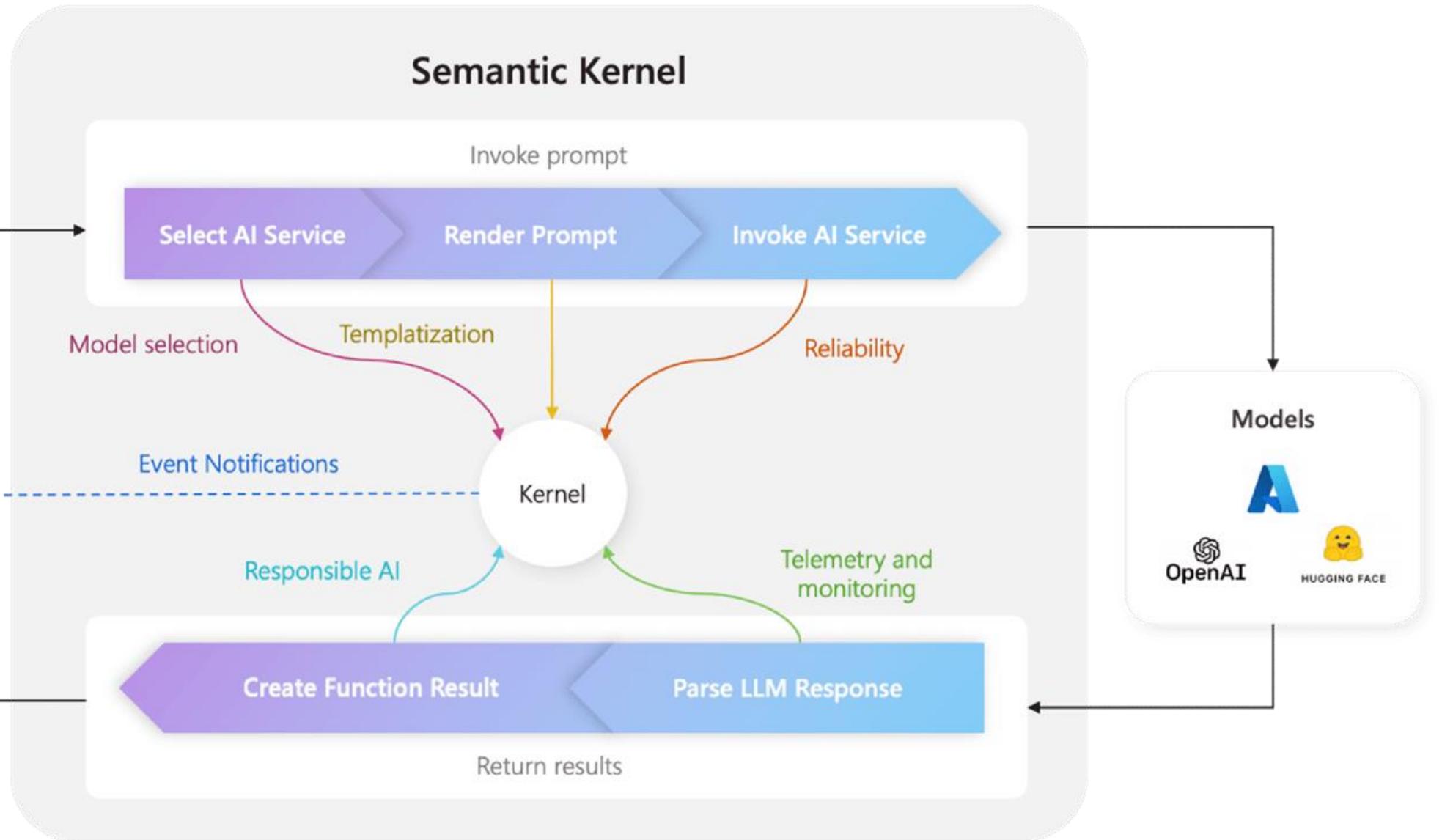
SEMANTIC KERNEL
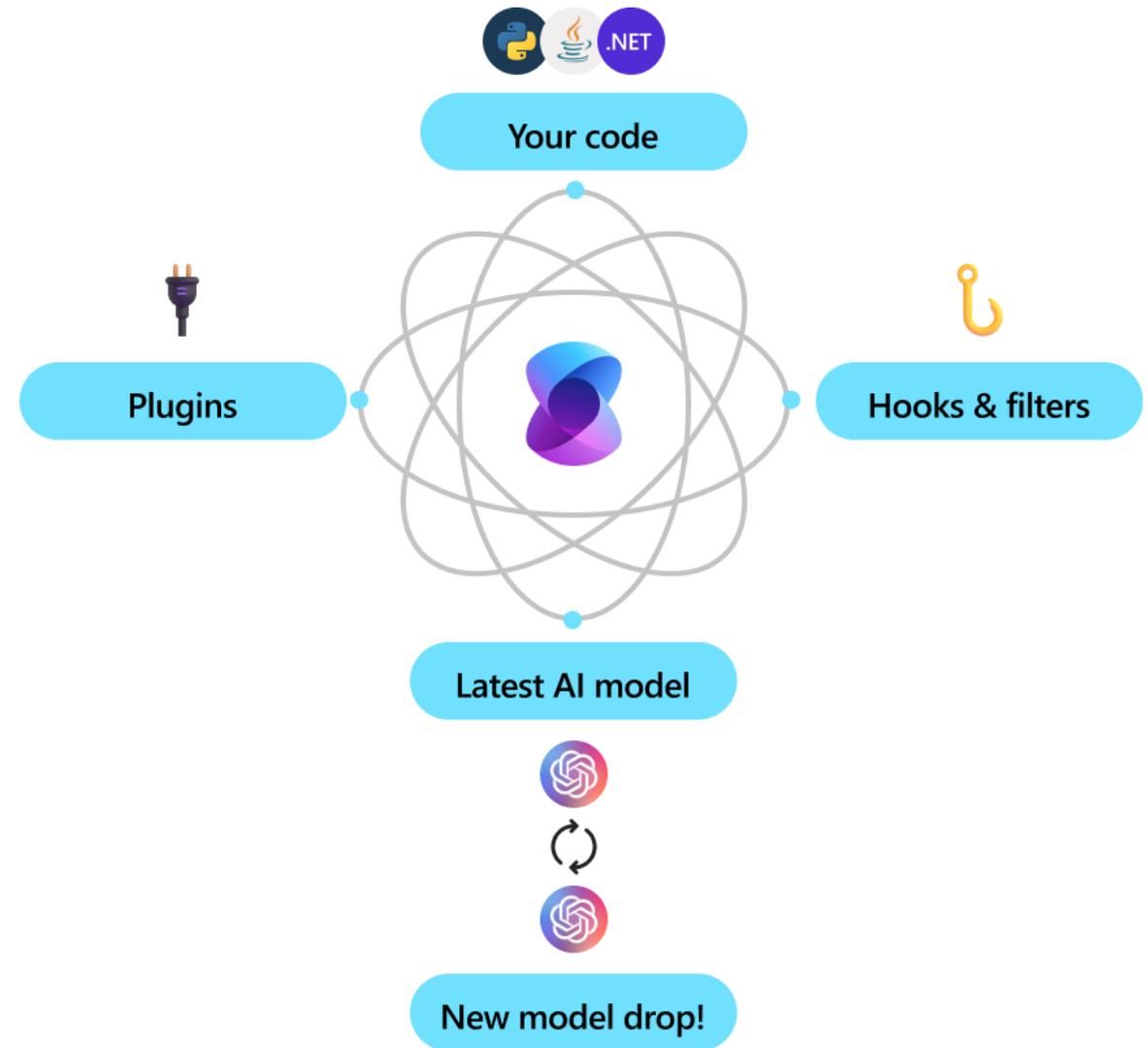
SEMANTIC KERNEL

- SDK for building AI agents and integrating AI models into your apps

- Microsoft, open-source

- C#, Java, Python

- Bridge between normal AI usage and your code, which helps simplify process of developing AI-powered apps

WHAT MAKES UP SEMANTIC KERNEL?

- Kernel itself is basically just a DI container
- Central location to configure/monitor agents
- Two types of components

1. **Services**. AI services like chat completion, but also logging, HTTP clients, etc. Modeled after existing service provider in .NET

2. **Plugins**. Components used by your AI services to perform work, like retrieving data from a database or calling an external API

WHY SEMANTIC KERNEL?

- Devs normally have to learn the different APIs/quirks of each service
- SK abstracts and unifies APIs
- Bridge between your code and LLM
- Model used must support function calling

CODING DEMO

Get weather   Send email   Search knowledge base

```javascript
Function calling example with get_weather function                          javascript ⇕  ⧉

1   import { OpenAI } from "openai";
2
3   const openai = new OpenAI();
4
5   const tools = [{
6       "type": "function",
7       "name": "get_weather",
8       "description": "Get current temperature for a given location.",
9       "parameters": {
10          "type": "object",
11          "properties": {
12              "location": {
13                  "type": "string",
14                  "description": "City and country e.g. Bogotá, Colombia"
15              }
16          },
17          "required": [
18              "location"
19          ],
20          "additionalProperties": false
21      }
22  }];
23
24  const response = await openai.responses.create({
25      model: "gpt-4.1",
26      input: [{ role: "user", content: "What is the weather like in Paris today?" }],
27      tools,
28  });
29
30  console.log(response.output);
```

https://platform.openai.com/docs/guides/function-calling?api-mode=responses&lang=javascript

RESPONSIBLE AI

1. **Fairness**: AI systems should treat all people fairly.
2. **Reliability and safety**: AI systems should perform reliably and safely.
3. **Privacy and security**: AI systems should be secure and respect privacy.
4. **Inclusiveness**: AI systems should empower everyone and engage people.
5. **Transparency**: AI systems should be understandable.
6. **Accountability**: People should be accountable for AI systems.

RESOURCES

- **3Blue1Brown Neural networks playlist**

- **Andrej Karpathy Intro to Large Language Models**

- **Microsoft Github samples - Getting Started**

- **Semantic Kernel Microsoft docs**

- **Semantic Kernel course on Microsoft Learn**

DANIEL WARD

PROFESSIONAL SOFTWARE CONSULTANT

dward@leantechniques.com

leantechniques.com

Q&A

THANK YOU!