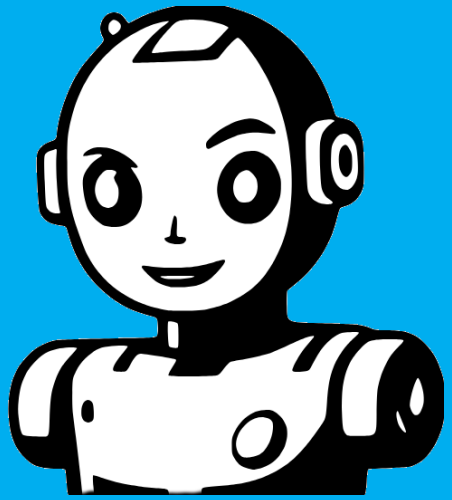
Warwick Business School

# Data Science & Generative AI

**Dr Michael Mortenson**

Associate Professor (Reader)
*michael.mortenson@wbs.ac.uk*



## Session 9: AI Applications and Agentic AI

# 1.1 Full Transformer Architecture (DeepSeek)

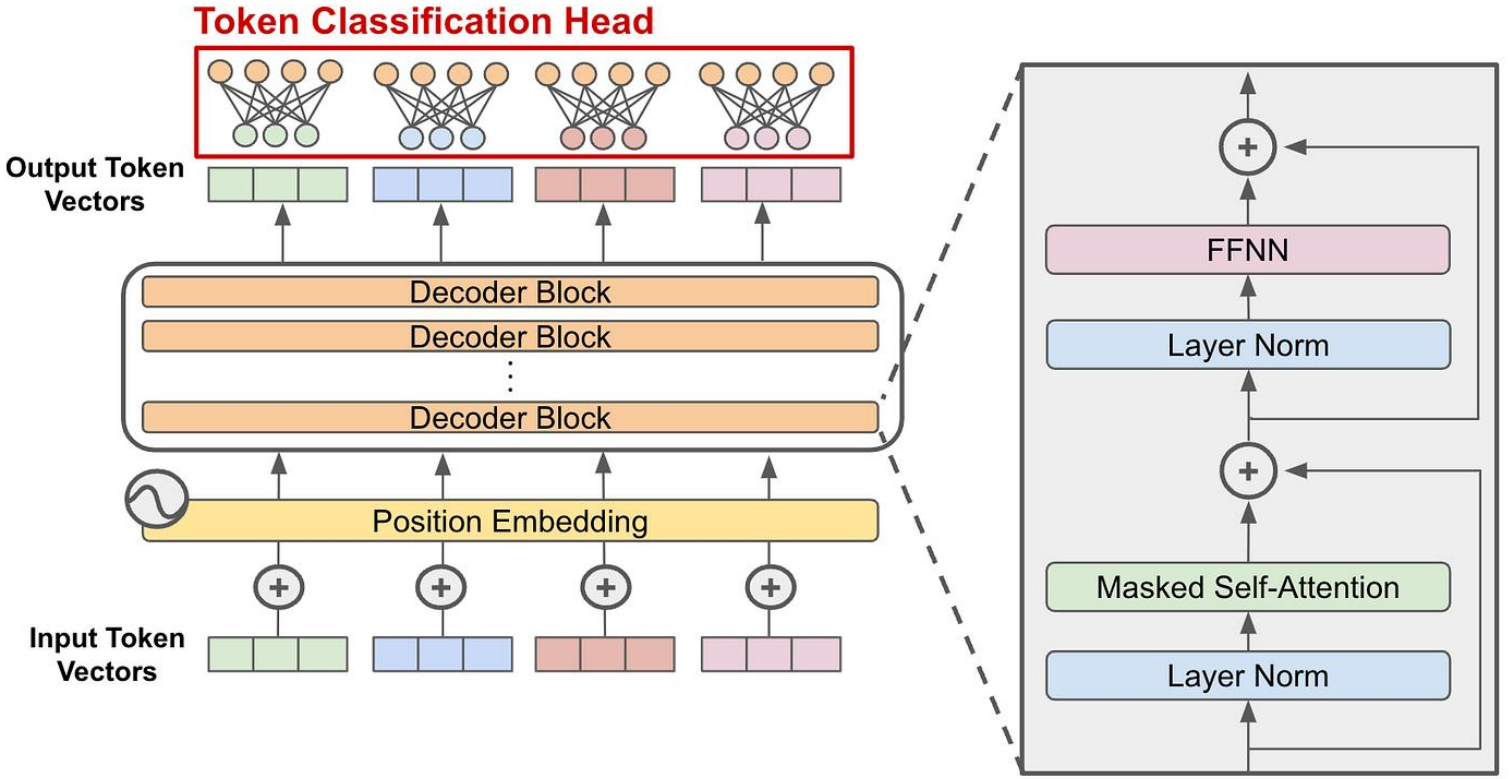





*Image Credit: https://medium.com/@tanalpha-aditya/deepseek-coder-when-the-llm-meets-programming-better-than-gpt-3-5-054cf85e3493*

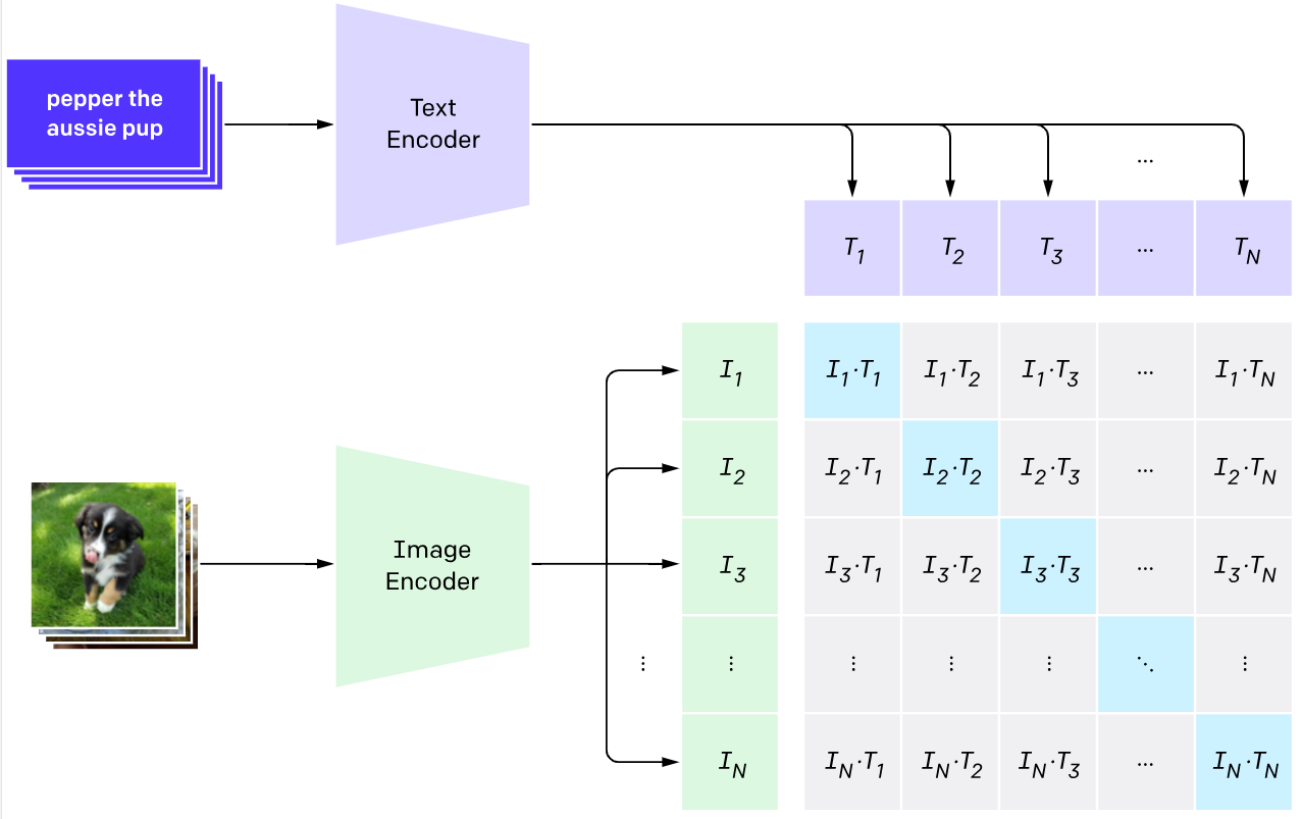
# 1.2 Prompting

| Prompt Element | Purpose |
| --- | --- |
| **Role** (“You are a...”) | Sets perspective and voice |
| **Task** (“Summarise...”) | Gives the main instruction |
| **Focus** (“Emphasise X”) | Directs attention to what matters |
| **Format** (“Bullet points, 3 sentences”) | Shapes the output |
| **Tone** (“Plain English, formal”) | Affects language and style |
| **Contraints** (“Avoid quotes, use 3–4 sentences”) | Reduces ambiguity |

# 1.3 Vision Transformers (ViT)

- Whilst designed specifically for language tasks, transformers can be used in a wide range of applications, most notably computer vision (image processing).
- Ultimately, the same processes that make transformers work for text, also can be applied to the image domain:
  - Embeddings of image patches (e.g. 16x16px squares) are created and combined with positional encoding (e.g. top left of the image).
  - Self-attention (context) is used to “nudge” the embeddings of the patch based on the context of the picture.
- ViT has very high benchmark scores on standard vision tasks. However, they are very, very data hungry compared to Conv Nets.

# 1.4 Combing Text and Images



# 1.4 Combing Text and Images

- CLIP (Contrastive Language–Image Pre-training) takes an input a set of image and text pairs – web images that have *meta descriptions* (written summaries of the images).
- An image and a text encoder are trained separately for each.
- We then train a joint model that predicts the correct text description pair for the provided image. E.g. we get an image of a puppy with a text description of “pepper the aussie pup” and try to predict the description from the image.
- Both encoding models are fine-tuned in this way so that the embedding spaces of each are tuned to find shared concepts (i.e. the text “cowboy” can be accessed from an image of a cowboy).

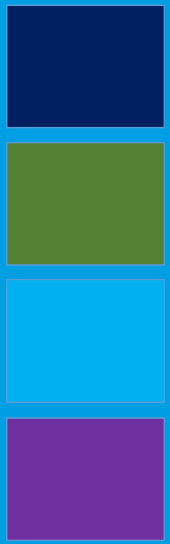
# Session Aims

Introduction

**Retrieval Augmented Generation (RAG)**

AI Agents and Agentic AI

The Path to Artificial General Intelligence



# 2.1 Knowledge Bases (KBs)

- While the training sets for LLMs are very large, that does not mean they contain all the information we might need. In many businesses we have specialised knowledge we would want the AI to be aware of.
- Equally, business will have protected information (for confidentiality and/or commerciality reasons) they won't want to be part of a general training set of information.
- The goal, therefore, is often to combine the generative and intelligence capabilities of an LLM, with the specific knowledge a business may store.

# 2.2 One-shot and Few-shot Prompting

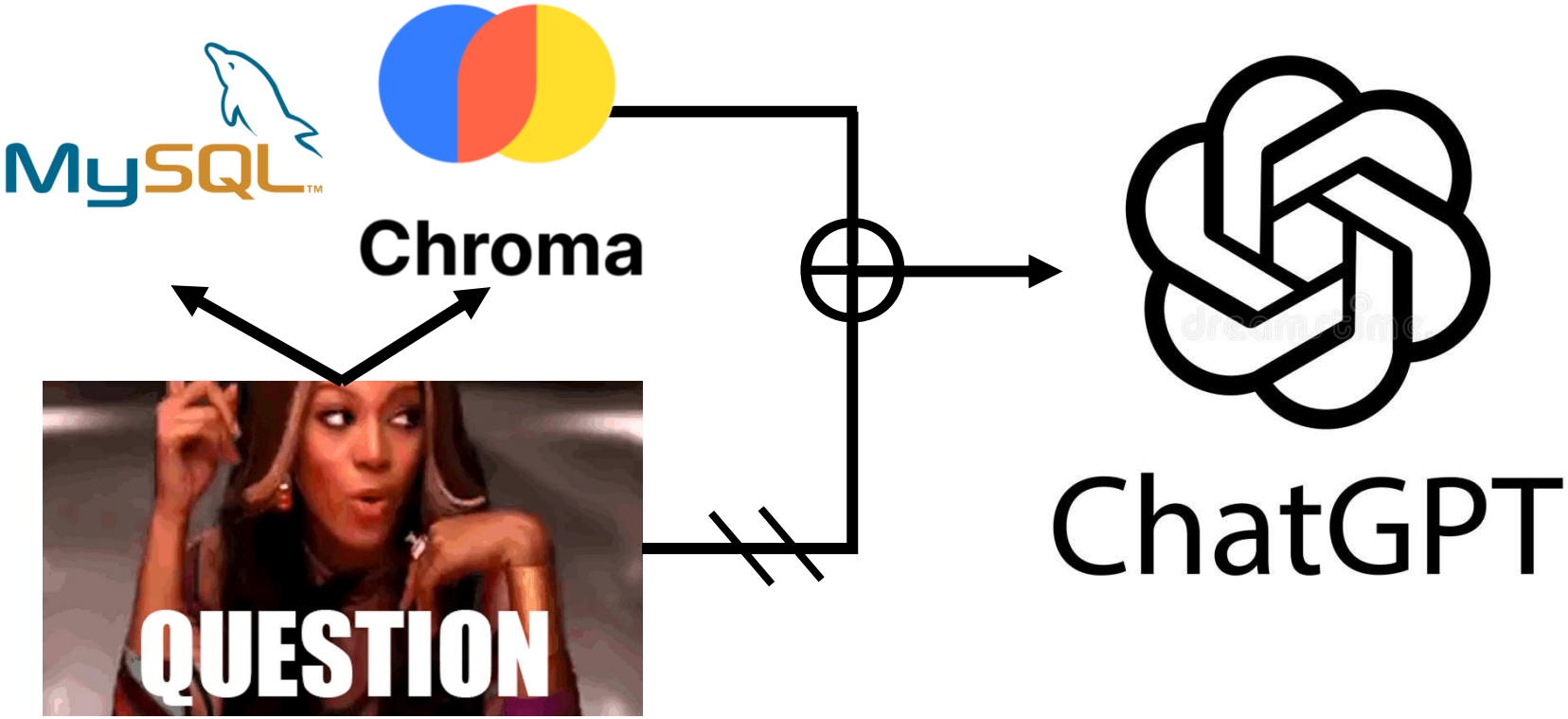
Sentence: "my dog is asleep on the sofa".
Answer: this sentence has neutral sentiment.
Sentence: "my dog has upset everyone by eating the food in the kitchen"
Answer: ?

**Answer:** This sentence has **negative sentiment**.

**Reason:**

- The phrase "**has upset everyone**" conveys emotional distress or dissatisfaction.
- It implies a negative consequence caused by the dog's action (eating the food).
- Overall, the sentence reflects an undesirable or problematic situation.

# 2.3 Augmented Prompts



## 2.3 Augmented Prompts

"""Use the following pieces of context to answer the question at the end. If you don't know the answer, just say that you don't know, don't try to make up an answer.

{context} # a variable that stores the retrieved knowledge

Question: {question} # the user’s original question

Answer:"""

# 2.4 Retrieval Augmented Generation

*"Retrieval-augmented generation is an AI framework for improving the quality of LLM-generated responses by* ***grounding the model on external sources of knowledge*** *to supplement the LLM's internal representation of information.*

*Implementing RAG […] has two main benefits: It ensures that the* ***model has access to the most current, reliable facts****, and that* ***users have access to the model's sources****, ensuring that its claims can be checked for accuracy and ultimately trusted."*

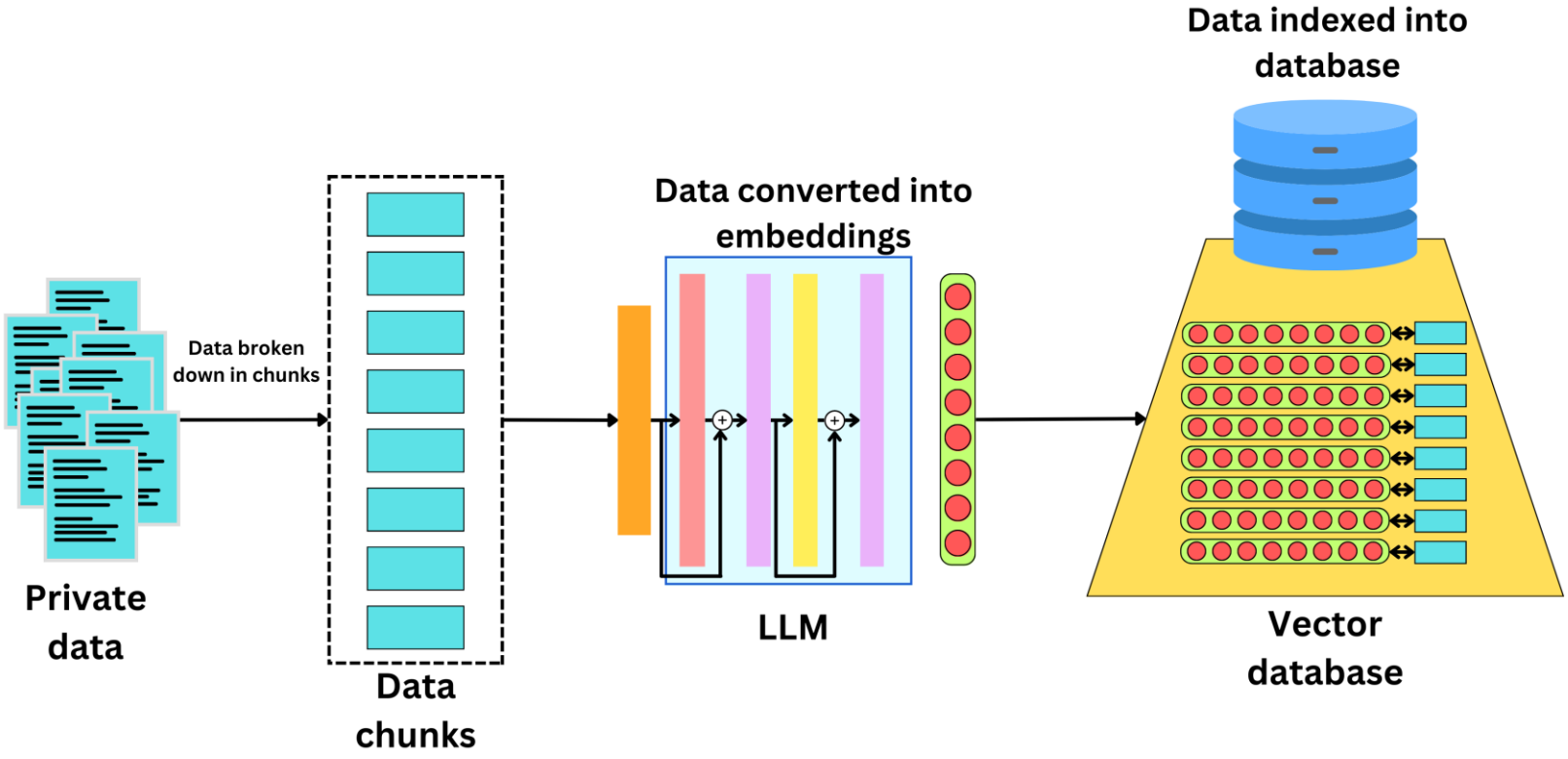


Martineau K (2023). *What is retrieval-augmented generation?* IBM. https://research.ibm.com/blog/retrieval-augmented-generation-RAG.

# 2.4 Retrieval Augmented Generation

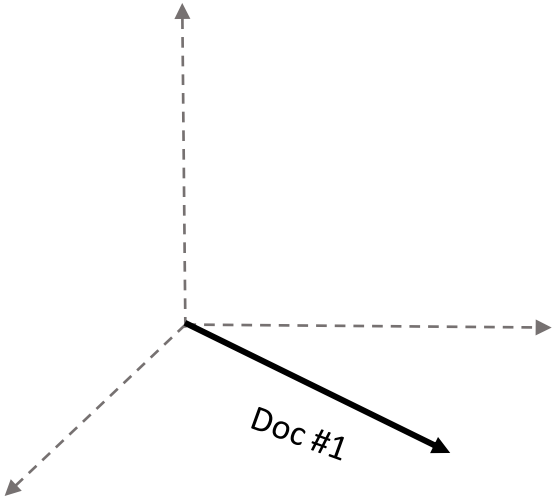
- RAG systems comprise of:
  - Some form of database(s) to store relevant context;
  - A search mechanism to retrieve context;
  - A prompt template(s) that can be augmented with the user query and the retrieved context;
  - (A) LLM(s) that are queried with the prompt;
  - An overall orchestration system that can manage all of these elements and return the generated content to the user.

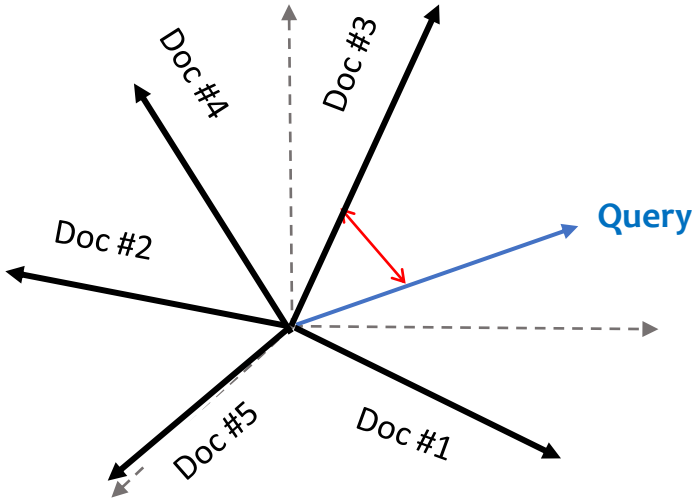
# 2.5 Vector Database



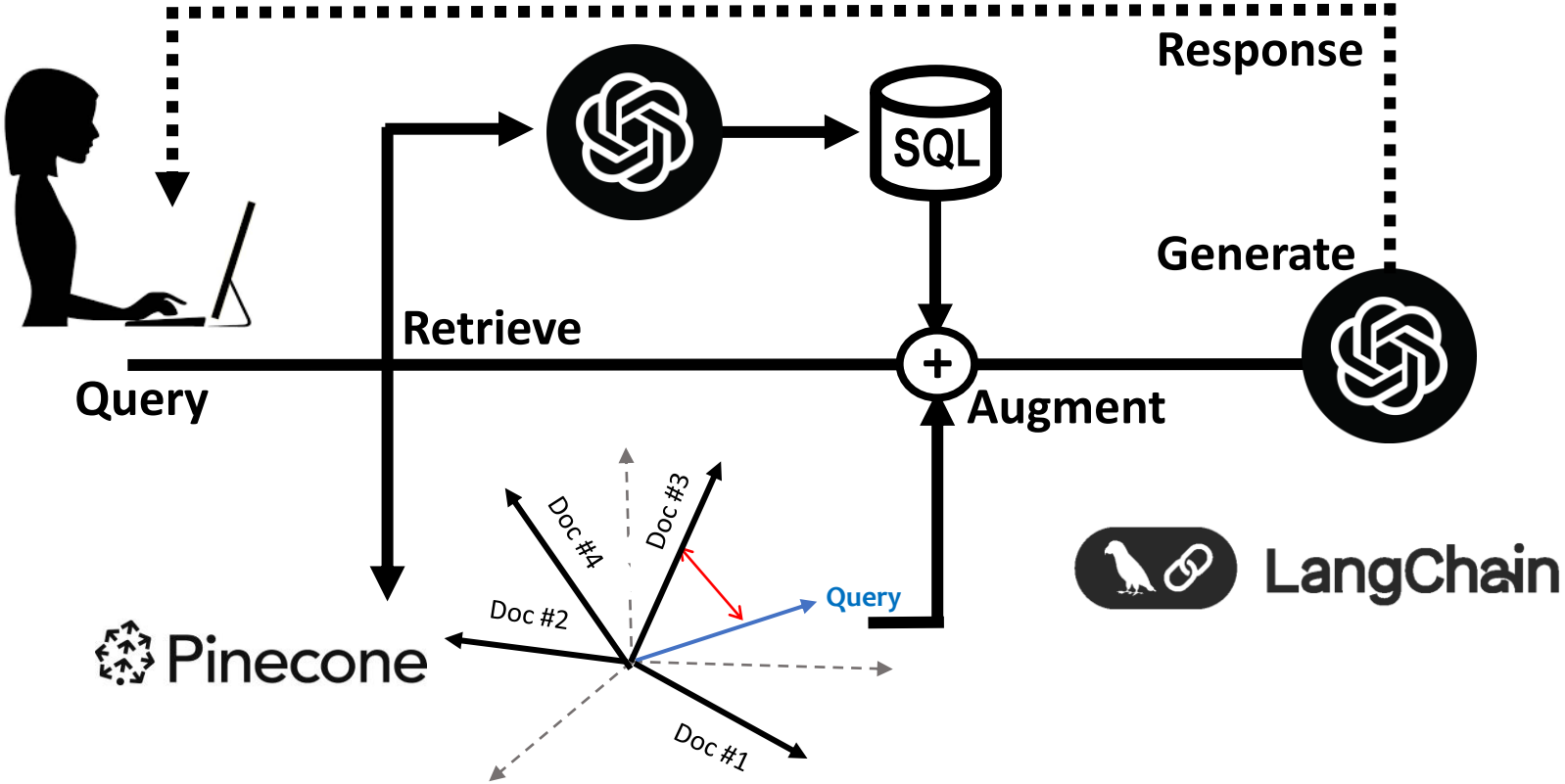
# 2.6 Semantic Search

- To retrieve context we need a search mechanism. In vector DBs we embed the users query (with the same model used to embed the knowledge base) and then find the documents that have the closest semantic similarity (shortest distance in embedding space).

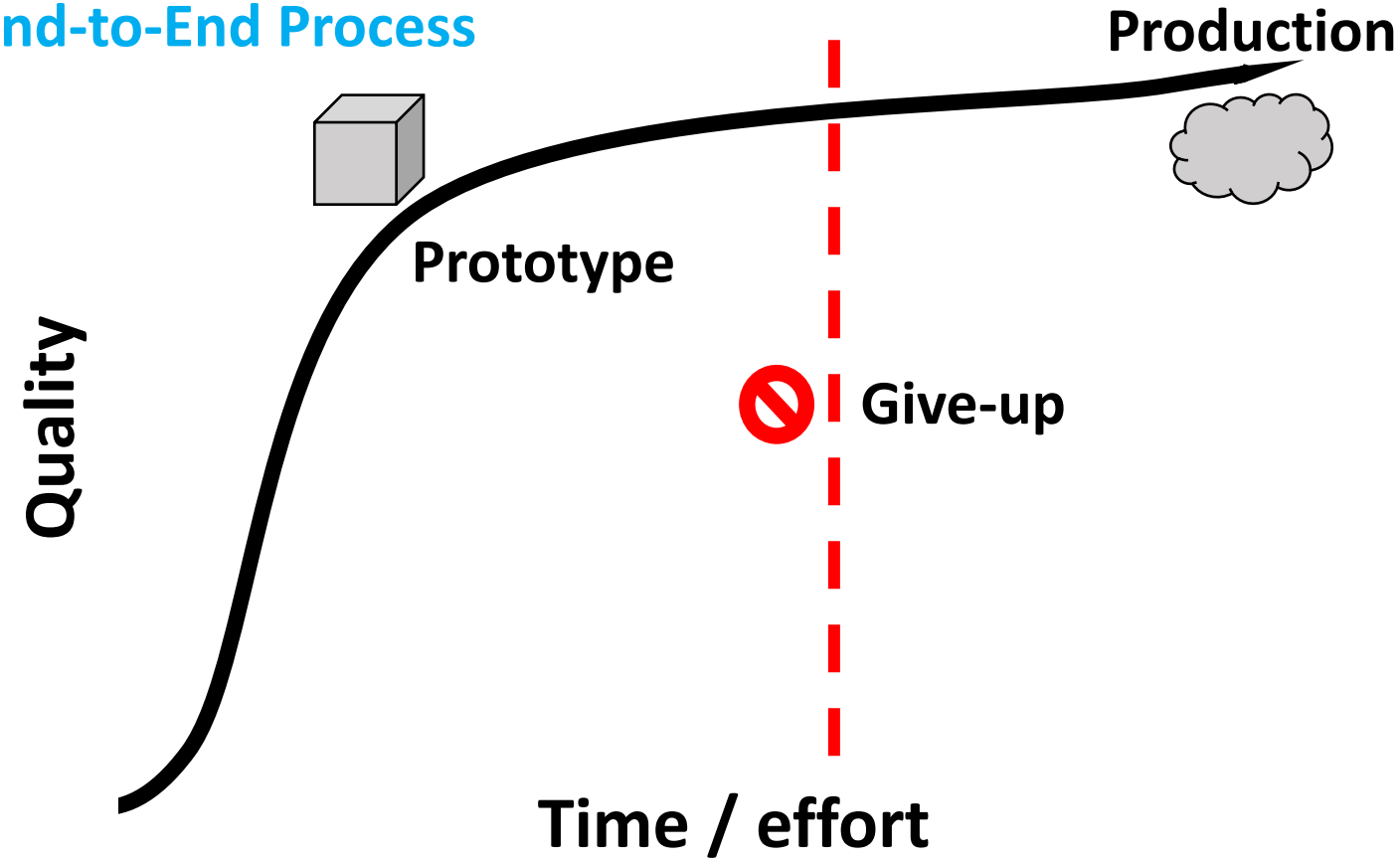


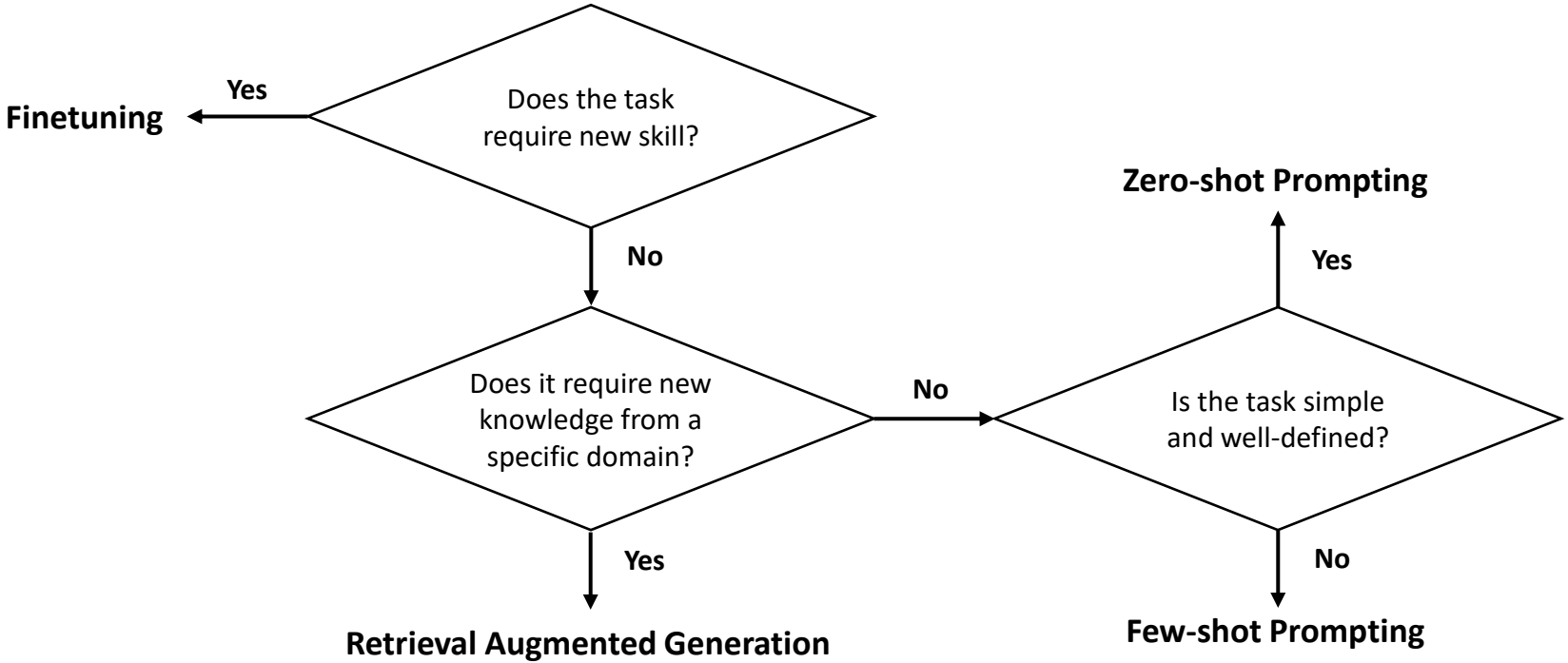
# 2.7 End-to-End Process



# 2.7 End-to-End Process



# 2.8 Prompting, RAG and Fine-tuning



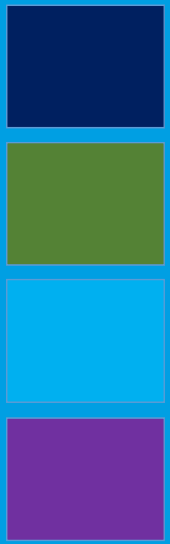
# Session Aims

Introduction

Retrieval Augmented Generation (RAG)

**AI Agents and Agentic AI**

The Path to Artificial General Intelligence



# 3.1 Modern AI 1.0



*Image Credit*: https://www.sologo.ai/blog/decoding-ai-product-logos-trends-key-elements-and-typography/

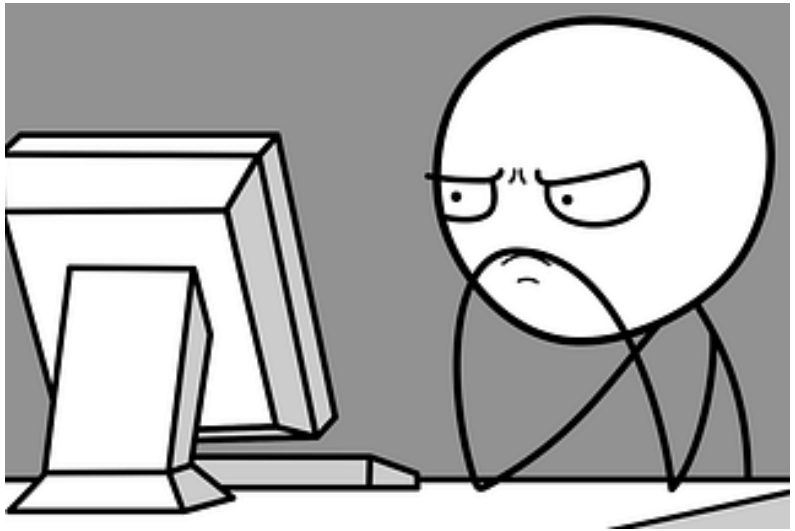
# 3.2 Modern AI 2.0




*Image Credit:* AMC

# 3.2 LLMs and Agents (Reasoning Agents)

- Previously programming was the only way we could instruct computers to perform tasks.
- LLMs allows us to **instruct computers in human language**.
- LLMs allows **computers to talk to each other in human language**.
- Software (traditionally) is a **deterministic process**. It requires us to determine “the rules” up front.
- LLMs are **non-deterministic**. It can work with any input we provide it.



# 3.2 LLMs and Agents (Reasoning Agents)

- An agent is an AI tool (today this is transformer-based models) that specialises in a specific task or set of tasks.
- We can achieve this specialisation by:
  - Giving it access to specific data (e.g. an agent with a knowledge base designed to answer questions on HR policies).
  - Giving it access to specifics tools and systems (e.g. an agent that summarises online meetings).
  - Re-training (fine tuning) the AI on company specific data and to complete specific training tasks.
  - Building networks of agents that can work together and influence each other to complete complex tasks.

# 3.3 AI Agent Components

- **Goal:** An agent should be given a specific goal to achieve.
- **Memory**: In most agent systems, the agent will have access to short-term memory (i.e. session history) and, sometimes, long-term memory (the ability to store and retrieve information in a database for instance).
- **Tools:** Access to coded tools or external systems and networks (next slide).
- **Environment:** Agents will typically operate within some form of environment. This may be a simulated world / gym or a set of interactions with users or other agents.
- **Logic:** Some logical approach to reasoning (normally specified in the prompt template).

# 3.3 AI Agent Components

- Provision of tools allows agents to perform tasks that they have not been trained on and/or that transformers are ill-suited to, such as:
  - Calculators;
  - Internet search or database search;
  - Retrieval Augmented Generation (RAG) – text-based knowledge bases;
  - Analytical models – e.g. optimisation or ML models;
  - Business systems – e.g. Outlook, a CRM system, etc.;
  - Bespoke software designed to achieve particular tasks.

# 3.4 MCP: Model Context Protocol

- A framework for working with AI agents introduced by Anthropic (i.e. the Claude models).
- **Model** – e.g. an LLM
- **Context** – e.g. instructions via a prompt, contextual information from documents or internet search, the actual user query
- **Protocol** – a set of rules and standards for building multi-aspect systems (often mixing commercial or open-source LLMs, internal data stores, programming repositories (such as Github) for tools and more).
- "*Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect your devices to various peripherals and accessories, MCP provides a standardized way to connect AI models to different data sources and tools.*" (Anthropic, 2025).

# 3.5 Think → Act → Observe

- An evolution on Chain of Thought (CoT) prompting to extend beyond chatbots and into agent systems.
- **Think:** what is the best way to solve the specific user problem/query?
- **Act:** which tool or technique can be used to best resolve the problem/query?
- **Observe:** has this solved the problem? Do we need to complete the cycle again (with a different action)?

# 3.6 Succeeding with AI Agents

- ✓ Use SLM (small language models) that have been trained on our specific tasks.
- ✓ Carefully design agent prompts to provide clear instructions, but also represent the ‘philosophy’ of how things are done.
- ✓ Develop strong guardrails and governance tools.
- ✓ Carefully manage supervision of agents – via Supervisor Agents and Human-in-the-Loop.

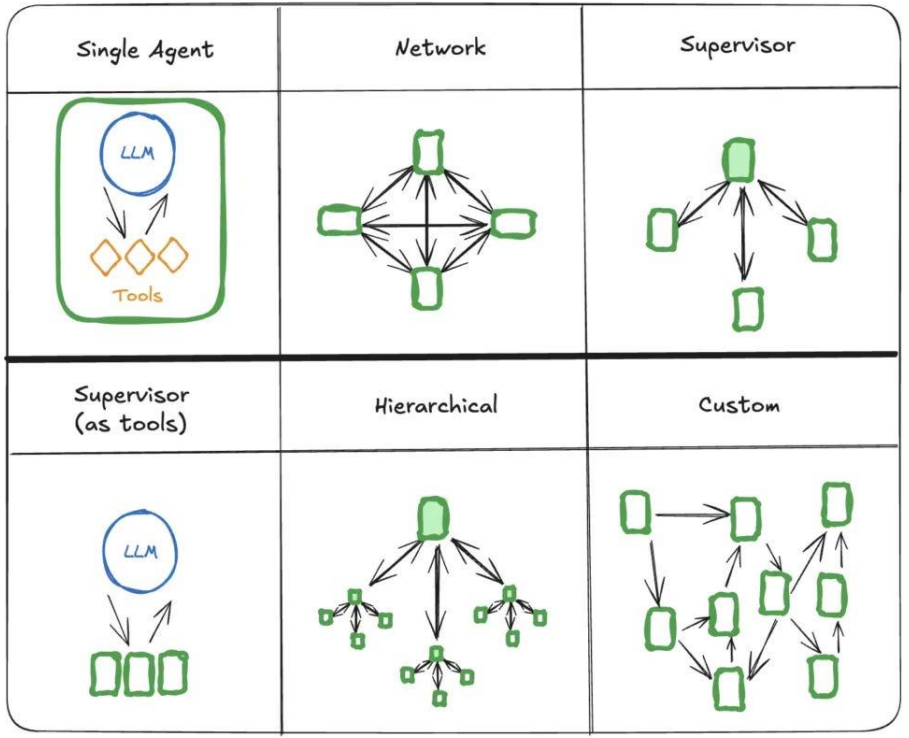





*Image Credit*: https://pub.towardsai.net/supervisor-style-the-king-of-multi-agent-systems-139f0990ed17

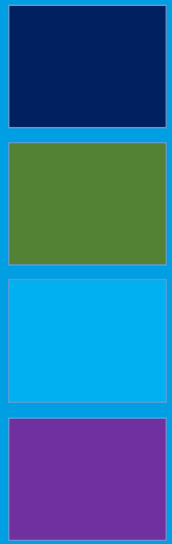
# Session Aims

Introduction

Retrieval Augmented Generation (RAG)

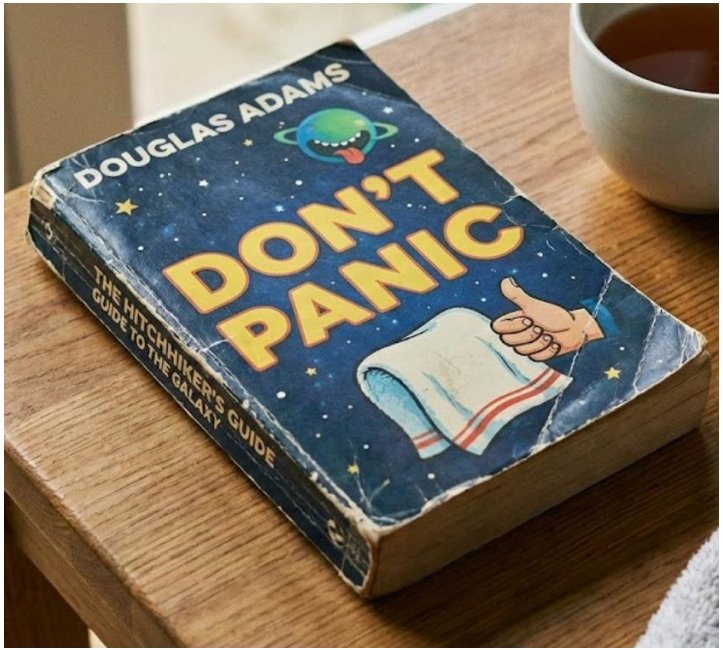
AI Agents and Agentic AI

**The Path to Artificial General Intelligence**



# 4.1 AI Agents Today: A reality check

- Building AI systems is hard. It is very much the bleeding edge.
- Many agent orchestration systems are in the very early stages of development and are hard to use.
- GIGO (Garbage In, Garbage Out) is a constant challenge – without good data your systems will fail.
- Agent systems are very hard to test. **Non-deterministic** systems behave … non-deterministically in practice.



# 4.2 The AGI Revolution will be Distributed

# 4.3 Agent Orchestration / Multi-Agent Systems

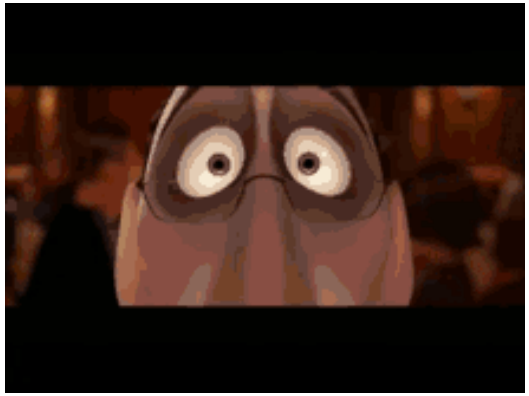
- Tools have emerged to orchestrate multi-agent systems. These systems support task such as:
  - **Routing** tasks and queries to agents.
  - **Tracking the state** of the system and access to **short- or long-term memory** (session data and database storage respectively).
  - **Tool** management and calling.
  - **Co-ordination between agents** e.g. passing messages and sharing results.
  - **Evaluating responses** and allowing agents to reflect/revise previous work.
  - **Error handling and recovery**.
  - **Prompt templates**.
  - **Human-in-the-Loop** – delivering outputs and queries to human experts to support the operation of the agent system.

# 4.4 What Might an Agentic AI System Look Like?

- An Agentic AI system would/could require:
  - **More intelligent and adaptive agents:** Current agents are typically instantiated as objects of a class, designed with a predefined goal, backstory, and task-specific logic. To reach higher levels of autonomy and sustain goal-directed behaviour over time, agents must become more intelligent - able to self-configure, adapt goals dynamically, and modify their internal architecture or tools based on changing environments and feedback.

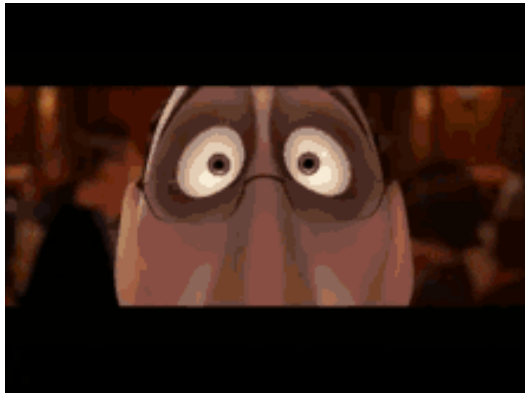
# 4.4 What Might an Agentic AI System Look Like?

- An Agentic AI system would/could require:
  - **More intelligent and adaptive agents:**
  - **More robust and naturalistic memory systems**
    Storing context in a session or external database is a good hack but still a hack. Agentic AI would require memory systems that are persistent, context-aware, and accessed in a more organic (non-deterministic, relevance-driven) manner. This might involve episodic and semantic memory structures similar to those found in cognitive architectures, enabling the agent to recall, reflect, and learn over time.

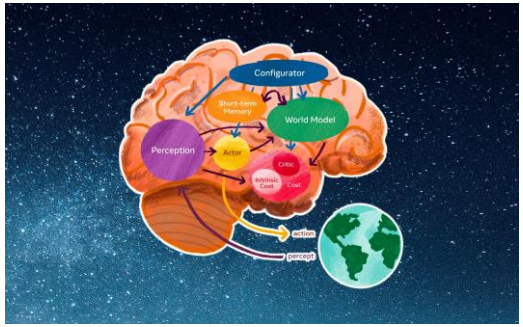
# 4.4 What Might an Agentic AI System Look Like?

- An Agentic AI system would/could require:
  - **More intelligent and adaptive agents:**
  - **More robust and naturalistic memory systems**
  - **Continuous learning and adaption:** Currently we have significant disjoint between training (often performed by third parties or vendors on proprietary datasets in a “closed source” fashion) and “test time” operations (the use of trained models as agents and the resources and instructions we provide). Agentic AI would require agents that can “learn” in use not just in training.



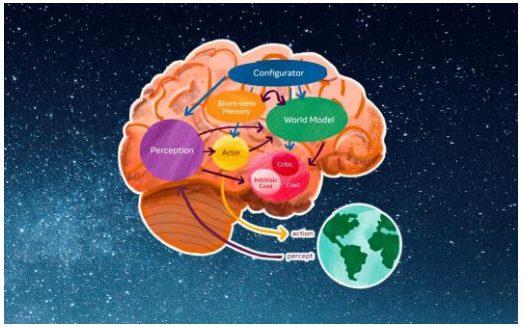
# 4.4 What Might an Agentic AI System Look Like?

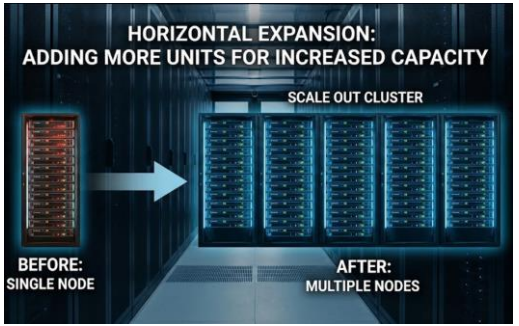
- An Agentic AI system would/could require:
  - **More intelligent and adaptive agents:**
  - **More robust and naturalistic memory systems**
  - **Continuous learning and adaption:**
  - **“Big Picture” as well as “Small Picture”:**
    The *ensembling* qualities of agents is attractive and resembles real-world approaches to organisational problem solving. However, a human agent would very much understand the “bigger picture” as well as their set of specific goals and tasks. Agentic AI agents would need a better world model, and an understanding of the overall state of the work and macro goals.

# 4.4 What Might an Agentic AI System Look Like?

- An Agentic AI system would/could require:
  - **More intelligent and adaptive agents:**
  - **More robust and naturalistic memory systems**
  - **Continuous learning and adaption:**
  - **“Big Picture” as well as “Small Picture”:**
  - **A more organic system and environment that is adaptable and responsive to system needs.**



# 4.5 Quality of Life

# 4.5 Quality of Life

# Further Reading


- 3Blue1Brown (2024). *Transformers (how LLMs work) explained visually | DL5*. https://www.youtube.com/watch?v=wjZofJX0v4M.
- Alammar J and Grootendorst M (2024). *Hands-On Large Language Models*. Sebastopol, CA: O'Reily.
- Alammar J (n.d.). *The Illustrated Transformer.* https://jalammar.github.io/illustrated-transformer/.
- **Goodfellow I, Bengio Y and Courville A (2016). *Deep Learning*. The MIT Press: Cambridge, MA.**
- Karpathy A (2023). *Intro to Large Language Models*. https://www.youtube.com/watch?v=zjkBMFhNj_g.
- Pajankar A and Joshi A (2022). *Hands-on machine learning with Python: implement neural network solutions with Scikit-learn and PyTorch.* Apress: Berkeley, CA.
- Vaswani A *et al.* (2017). Attention is all you need. In: *31st International Conference on Neural Information Processing Systems (NIPS '17)*. Red Hook, NY, USA, 6000–6010.