Generative AI & LLM in Practice

Kishoreraj Jayakumar

Self Introduction

KISHORERAJ JAYAKUMAR | Full Stack & Gen AI Engineer |

11 YEARS EXPERIENCE EXPERIENCE

Java

spring

Google Cloud Platform

AI

React JS

kafka

kubernetes

JS HTML CSS

python™

node JS

docker

https://www.linkedin.com/in/jkishoreraj/

NOKIA

Agenda:

- ➢ Gen AI Introduction
- ➢ Gen AI Usage & Platforms
- ➢ LLM Overview
- ➢ Prompt Engineering
- ➢ Prompting Techniques
- ➢ RAG
- ➢ Agentic AI
- ➢ Short Demo

Beginner Questions

1. What is Gen AI?
2. How Gen AI works?
3. How to talk to an LLM?
4. How can custom knowledge be provided to LLM?
5. How to use LLM for Autonomous Use cases?

Generative AI in AI/ML

Artificial Intelligence

This is the overarching concept, aiming to create machines capable of performing tasks that typically require human intelligence, such as understanding language, recognizing images, and making decisions

Machine Learning

A subset of AI, ML focuses on algorithms that allow computers to learn from data without being explicitly programmed.

Deep Learning

A further specialization within ML, DL uses artificial neural networks with multiple layers (deep neural networks) to analyse data and identify complex patterns.

Generative AI

A type of AI that uses deep learning techniques to create new content, such as text, images, music, or video, based on learned patterns from existing data.

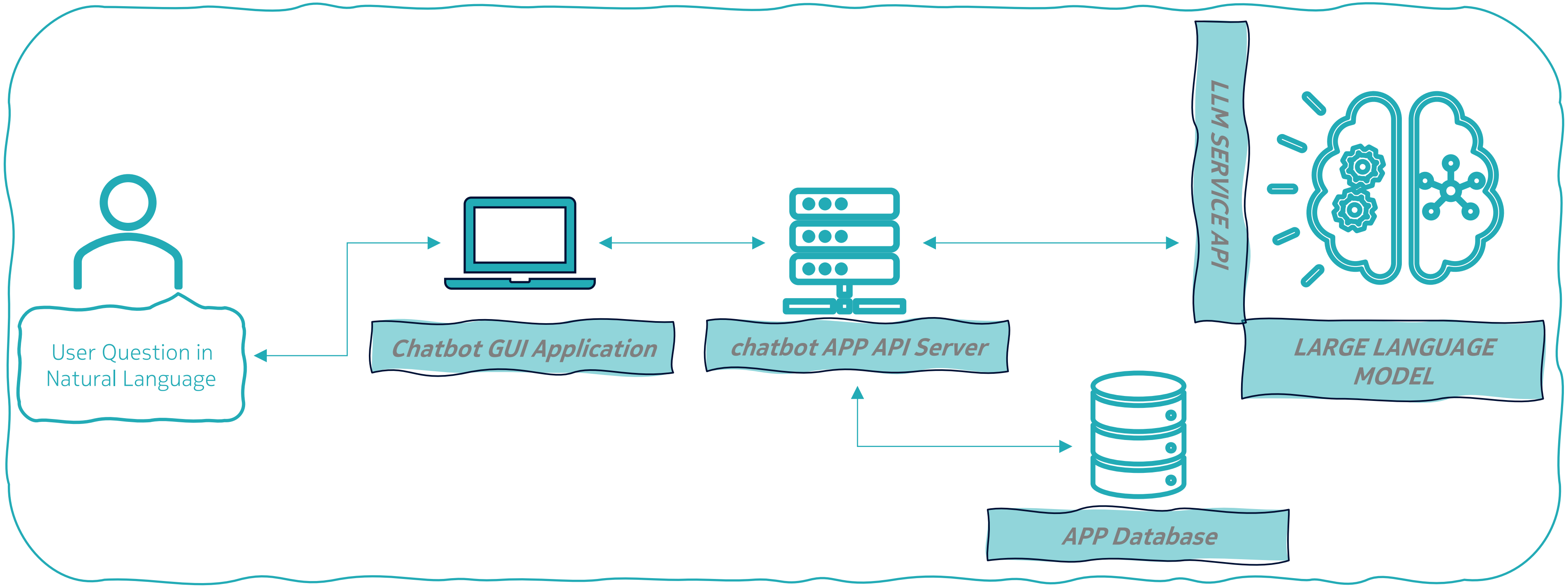
NOKIA

Generative AI Chatbots

some of the famous Gen AI Chatbots

ChatGPT	Owned By: **Open AI** Models: GPT-3*,GPT-4*,GPT-O*	DALL·E	Owned By: **Open AI** Models: DALL.E-2, DALL.E-3
Copilot	Owned By: **Microsoft** Models: GPT-4*	GitHub Copilot	Dev By: **GitHub & Open AI** Models: DALL.E-2, DALL.E-3
META AI	Owned By: **Meta** Models: Llama 3.2, Llama 3.1 -405B	Claude	Owned By: **Anthropic** Models: Haiku, Sonnet, Opus
Gemini	Owned By: **Google** Models: Gemini -1.5 & 2.0 (Flash/Pro/Lite)	perplexity	Owned By: **Perplexity AI** Models: sonar pro/reasoning

Gen-AI App – High-Level Design

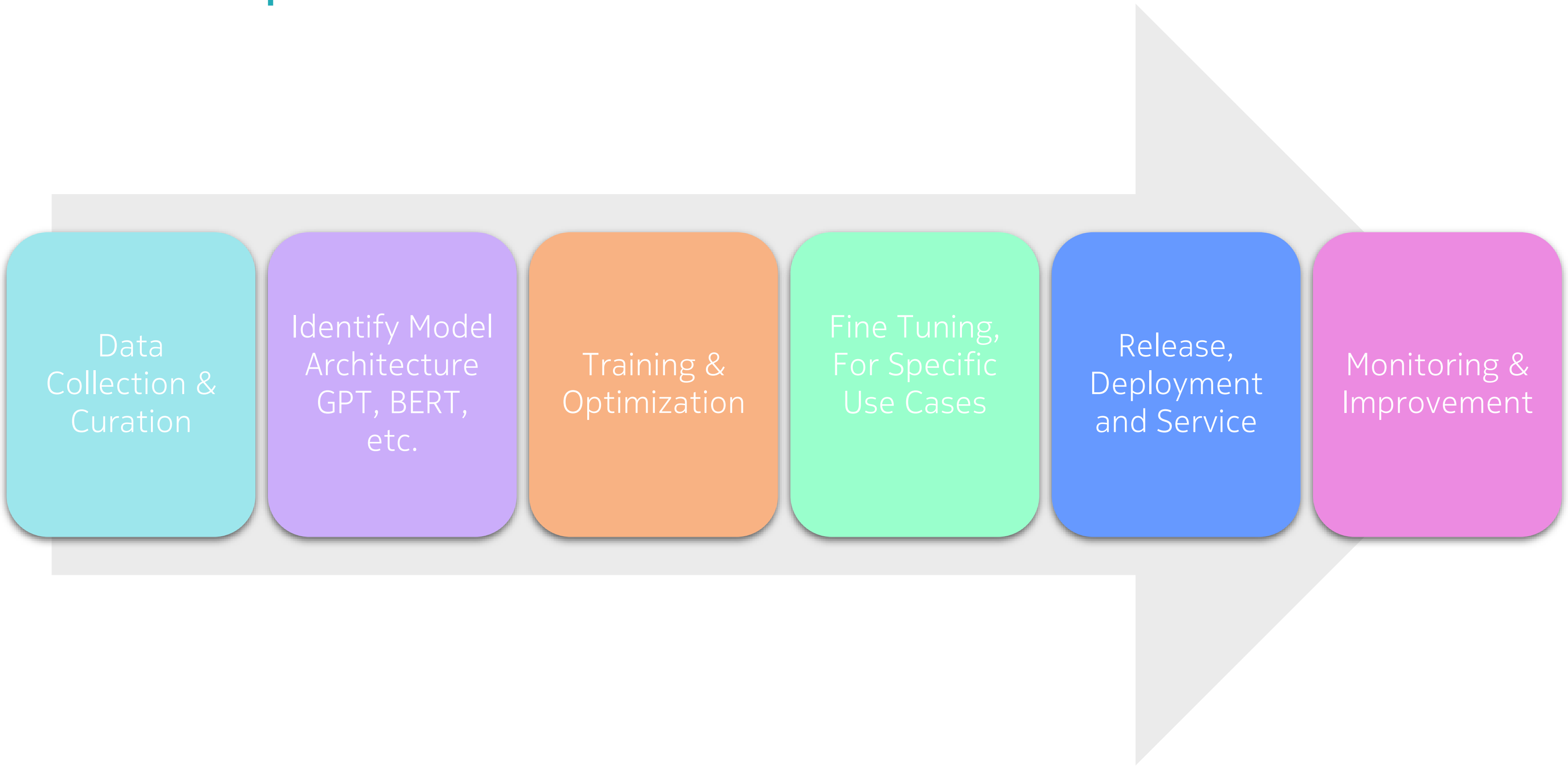


Large Language Models (LLMs)

- AI/ML Model built with deep learning techniques
- Neural network model trained with massive data
- Understands, interprets, and generates human-like text
- Mostly uses transformers architecture, and undergoes curation, and tokenization processes while training
- Knowledge level is commonly identified by “parameters”,

    Ex: Llama 8B , gemma 27B
- Widely used for text generation, language translations, chatbots, summarization and data reasoning, etc.,
- Also, multimodal LLMs are used for text, image, audio, video generation and interpretations

LLM Development Overview


Data Collection & Curation
Identify Model Architecture GPT, BERT, etc.
Training & Optimization
Fine Tuning, For Specific Use Cases
Release, Deployment and Service
Monitoring & Improvement

Open Vs Closed source LLMs

Aspect	**Open-Source LLMs 🛠️**	**Closed-Source LLMs 🔒**
Access	Free to use, modify, and distribute	Restricted access, proprietary licenses
Transparency	Fully transparent, code and training data available	Opaque, internal algorithms and data not disclosed
Customization	Fully customizable for specific needs	Limited customization (depends on API features)
Deployment	Can be deployed on-premise or in a private cloud with GPUs by ourselves	Hosted by provider, SaaS-based API access
Data Privacy	Full control over data and model usage	Data may be logged or used for improvement
Security	User-controlled security measures	Provider ensures security, but data exposure risks exist
Cost	Free to use, but infrastructure costs apply	Typically pay-per-use or subscription-based
Performance	Varies based on hardware and optimization	Generally well-optimized and high-performing
Scalability	Requires own infrastructure for scaling	Scales easily with provider's cloud infrastructure
Community Support	Strong open-source communities (Hugging Face, Meta, Mistral)	Limited, mostly official support
Regulatory Compliance	Easier to ensure compliance by self-hosting	Compliance depends on provider's policies
Updates & Maintenance	Requires manual updates and improvements	Automatically updated by provider
Examples	LLaMA, Mistral, Falcon, GPT-NeoX, BLOOM	GPT-4, Gemini, Claude, CoPilot

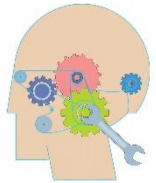
LLM Integration - Key Techniques:

Prompt Engineering

Crafting Effective Inputs

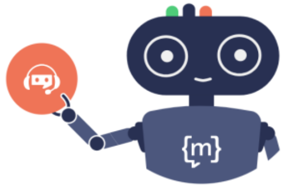
Retrieval Augmented Generation

Providing External Data



Fine-Tuning

Customizing a Pre-trained LLM



Agentic AI

Enabling LLM response to trigger actions

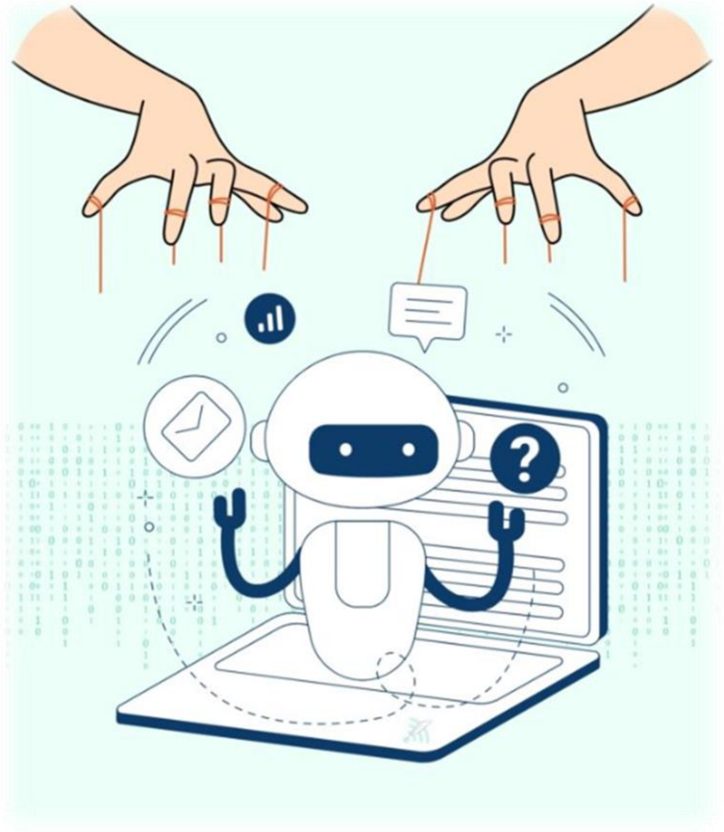
Prompt

A prompt is a query, command, or statement guiding the LLM to generate a desired output

The LLM model analyzes the prompt and uses its training data to produce a response that is relevant and appropriate to the prompt.

Prompt Engineering

1. The art and science of crafting input prompts to maximize the quality of output from language models.
2. Sets the context and intention
3. Directly influences model accuracy and relevance.
4. It improves communication between humans and machines, ensuring the resulting interaction is efficient and effective.



Why Does Prompt Engineering Matter?

Precision: Reduces ambiguity in model output.

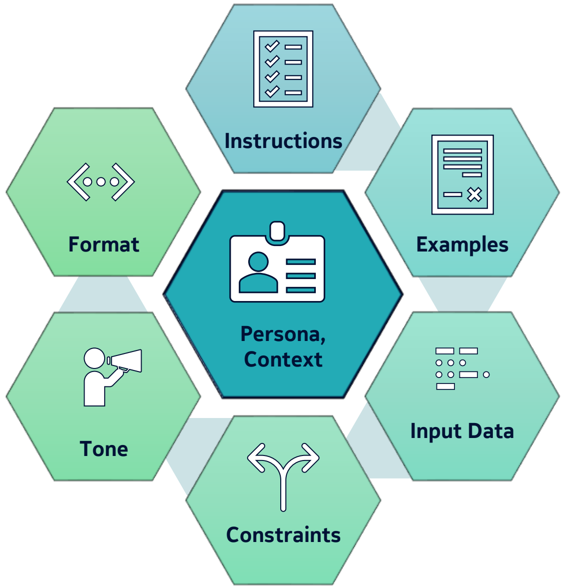
Efficiency: Minimizes trial-and-error cycles.

Adaptability: Tailors responses to specific domains.

Example: "Summarize this text" vs. "Summarize this text in 3 bullet points for a business audience."

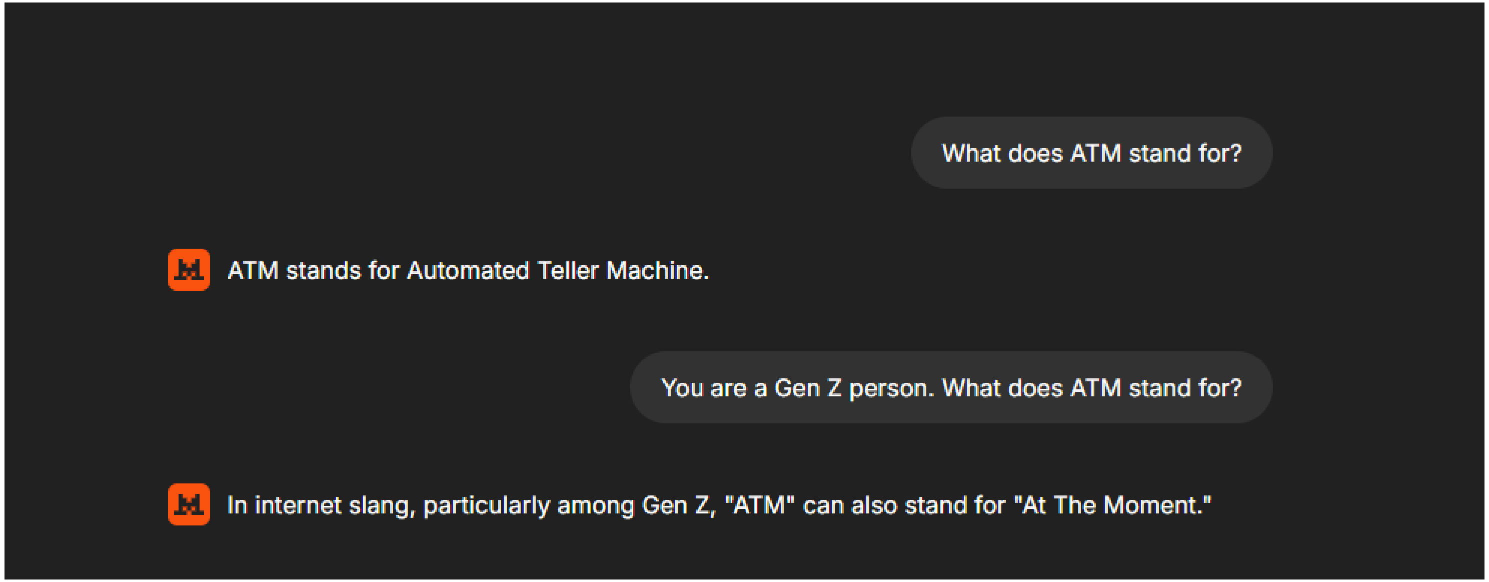
Prompting Elements

Different sections of a prompt, helps in guiding LLM for a desired response



Prompt Structure

Role

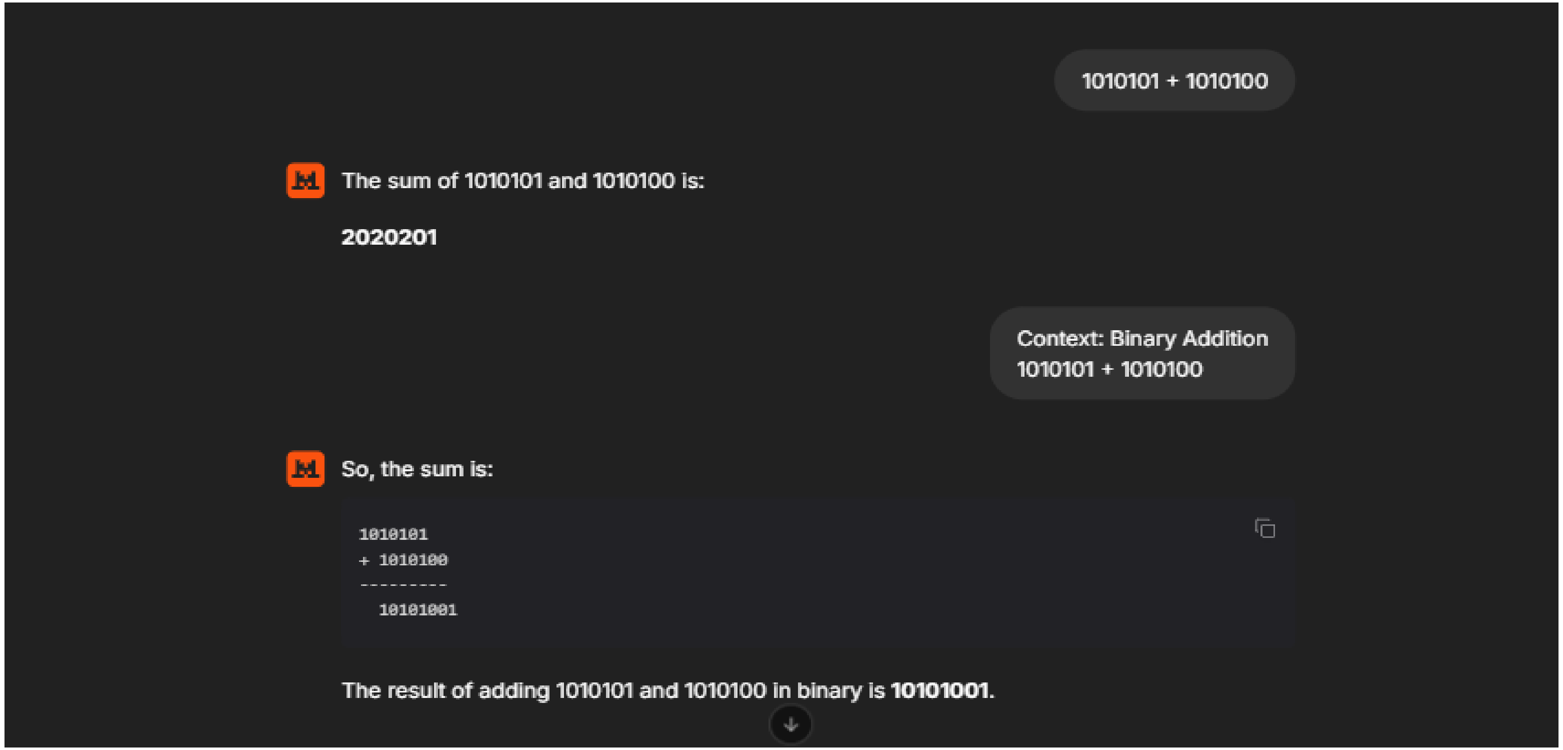

What does ATM stand for?

ATM stands for Automated Teller Machine.

You are a Gen Z person. What does ATM stand for?

In internet slang, particularly among Gen Z, "ATM" can also stand for "At The Moment."

Prompt Structure
Context



1010101 + 1010100

The sum of 1010101 and 1010100 is:

2020201

Context: Binary Addition
1010101 + 1010100

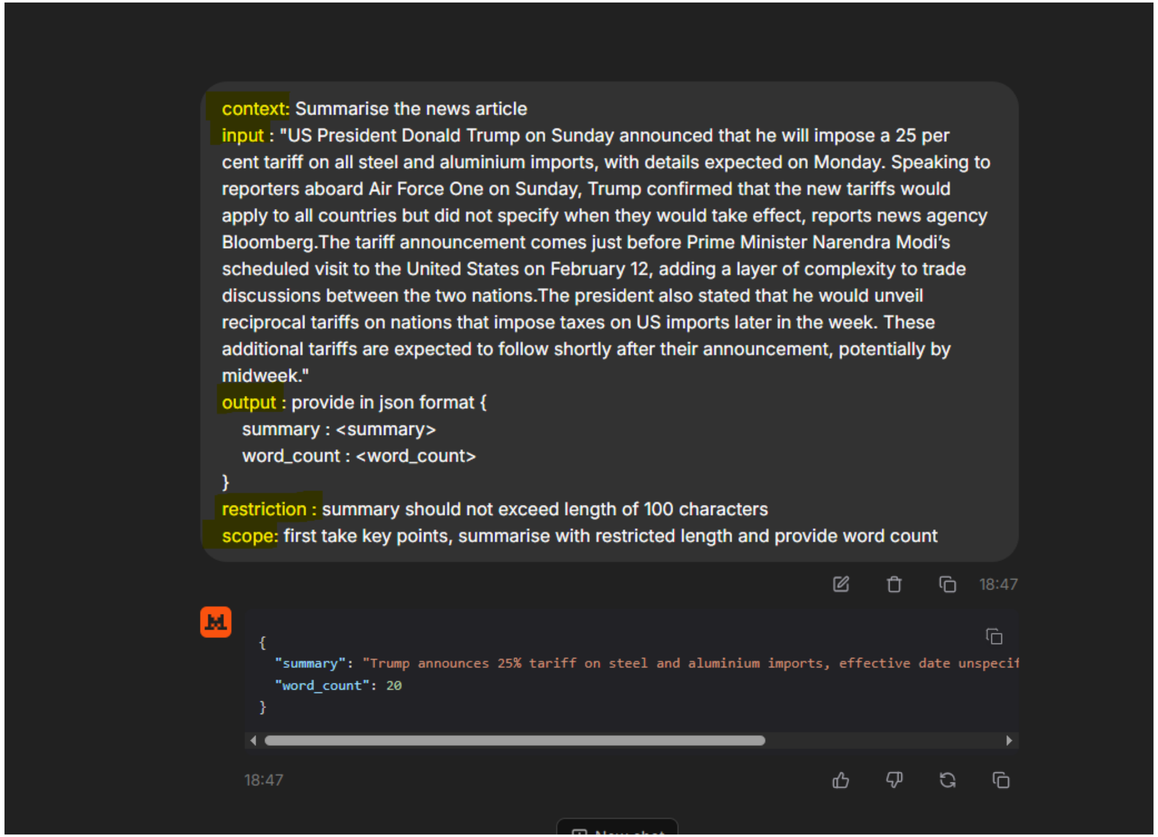
So, the sum is:

```
1010101
+ 1010100
---------
  10101001
```

The result of adding 1010101 and 1010100 in binary is **10101001**.

Prompt Structure

Final Example



context: Summarise the news article
input : "US President Donald Trump on Sunday announced that he will impose a 25 per cent tariff on all steel and aluminium imports, with details expected on Monday. Speaking to reporters aboard Air Force One on Sunday, Trump confirmed that the new tariffs would apply to all countries but did not specify when they would take effect, reports news agency Bloomberg.The tariff announcement comes just before Prime Minister Narendra Modi's scheduled visit to the United States on February 12, adding a layer of complexity to trade discussions between the two nations.The president also stated that he would unveil reciprocal tariffs on nations that impose taxes on US imports later in the week. These additional tariffs are expected to follow shortly after their announcement, potentially by midweek."
output : provide in json format {
  summary : <summary>
  word_count : <word_count>
}
restriction : summary should not exceed length of 100 characters
scope: first take key points, summarise with restricted length and provide word count

18:47

```
{
  "summary": "Trump announces 25% tariff on steel and aluminium imports, effective date unspecif
  "word_count": 20
}
```

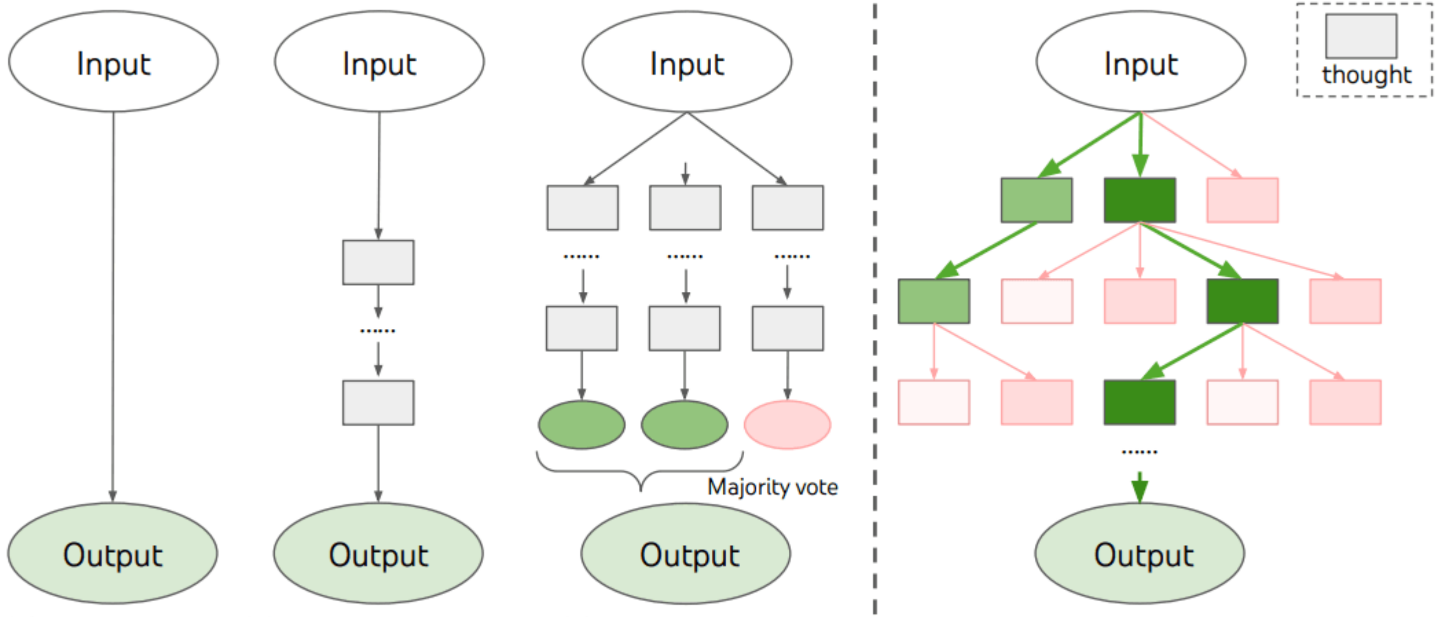
18:47

Prompting Techniques

Different ways of crafting prompts based on requirements

Zero Shot	**Direct questions without any examples** Example: “Explain black hole in simple terms”
Few Shot	**Provide a few examples before asking the model to generate a response** **Example**: *"Translate the following: 1. Hello → Hola 2.Thank you → Gracias 3.Good night → ???"* **Response**: "Buenas noches."
Role Based	**Assigning a specific role to the AI for contextual responses.** **Example**: "You are a South Indian grandmother. Explain why filter coffee is the best.“ **Response**: "Ayyo! Nothing beats fresh decoction and thick milk—better than any machine coffee!"
Chain of Thought	**Encourages step-by-step reasoning before answering.** **Example**: "If a train moves at 50 km/h for 3 hours, how far does it go? Think step by step.“ **Response**: "Step 1: Speed = 50 km/h - Step 2: Time = 3 hours Step 3: Distance = Speed × Time = 50 × 3 = 150 km."
Self Consistent – COT	**The model generates multiple CoT answers for n iterations of CoT prompts and picks the most consistent one** **Use Case:** Mathematical problems, logic puzzles, and complex decision-making.
Tree of Thought	**Instead of a single CoT path, the model branches into multiple reasoning paths, evaluates them, and picks the best.** **Example**: "How can I prepare for my college final exams effectively? Think in multiple ways before deciding the best approach.“ **Response**: "Path 1: Structured Study Plan, Path 2: Group Study Approach , Path 3: Visual & Memory Techniques150 km."

Prompting Illustrations



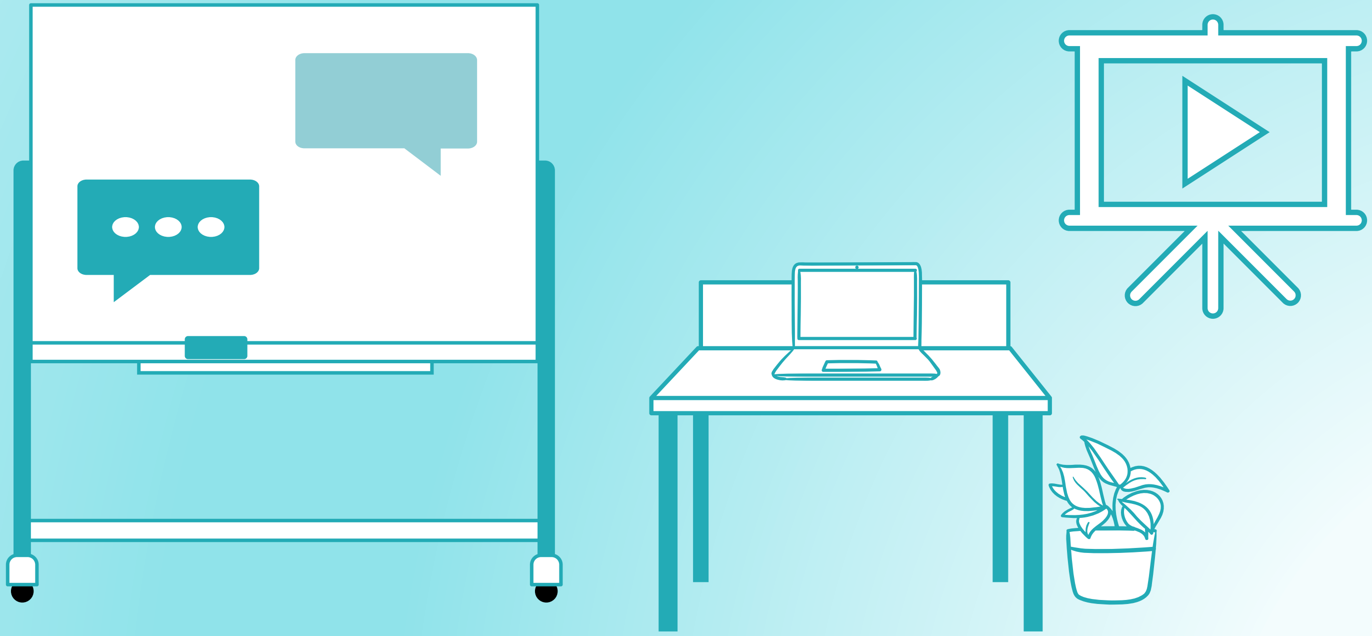
(a) Input-Output Prompting (IO) (c) Chain of Thought Prompting (CoT) (c) Self Consistency with CoT (CoT-SC)

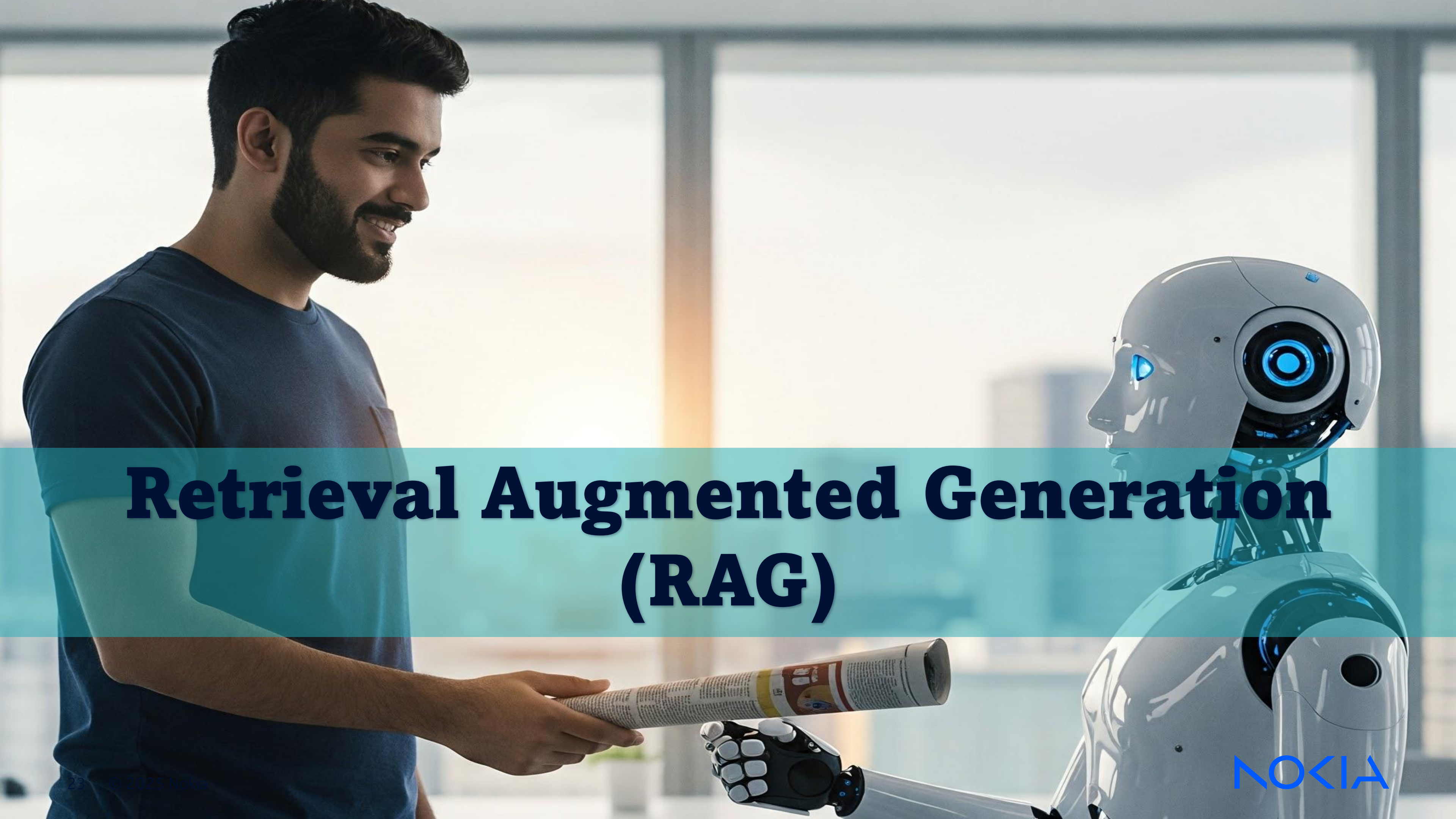
(d) Tree of Thoughts (ToT)

Prompting – Output Controlling Attributes

Parameter	Definition	Effect	Value Range	Use Cases
Temperature 🌡️	Controls randomness in responses.	Lower values (0.1-0.5) make responses **deterministic**, higher values (0.7-1.5) increase **creativity**.	**0.0 to 2.0**	Code generation (low), Creative writing (high)
Top-k Sampling 🎯	Limits word selection to the **k most probable** words.	Lower k makes responses **focused**, higher k makes them **diverse**.	**1 to 100**	Factual answers (low k), Storytelling (high k)
Top-p Sampling 🏆	Chooses words based on cumulative probability.	Lower p makes responses **conservative**, higher p allows **more diverse** outputs.	**0.0 to 1.0**	Formal writing (low p), Casual conversations (high p)
Frequency Penalty 🔁	Reduces repetition of frequently used words.	Higher values discourage repeated phrases.	**0.0 to 2.0**	Prevents repetitive responses in essays, articles
Max New Tokens 📝	Limits the number of generated tokens (words).	Higher values allow **longer responses**, lower values keep them **concise**.	**10 to 4096+**	Short summaries (low), Long-form writing (high)

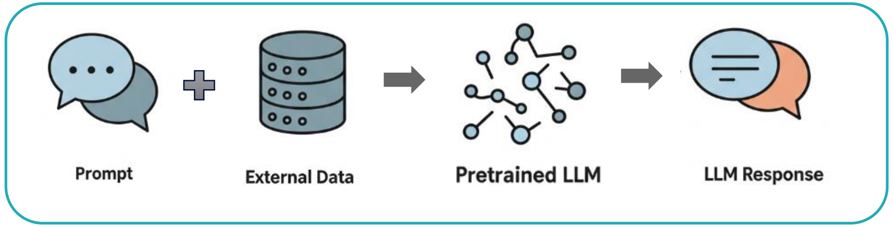
Prompting Demo





Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG)



What?

RAG is an AI technique that enhances text generation by retrieving relevant external knowledge before generating responses

Why?

Traditional LLMs rely only on pre-trained knowledge, which can become outdated. RAG **dynamically retrieves up-to-date information**, ensuring more factual and context-aware responses.

When?

It is used when AI needs to **answer domain-specific, dynamic, or real-time queries**, such as **legal, medical, or technical** support

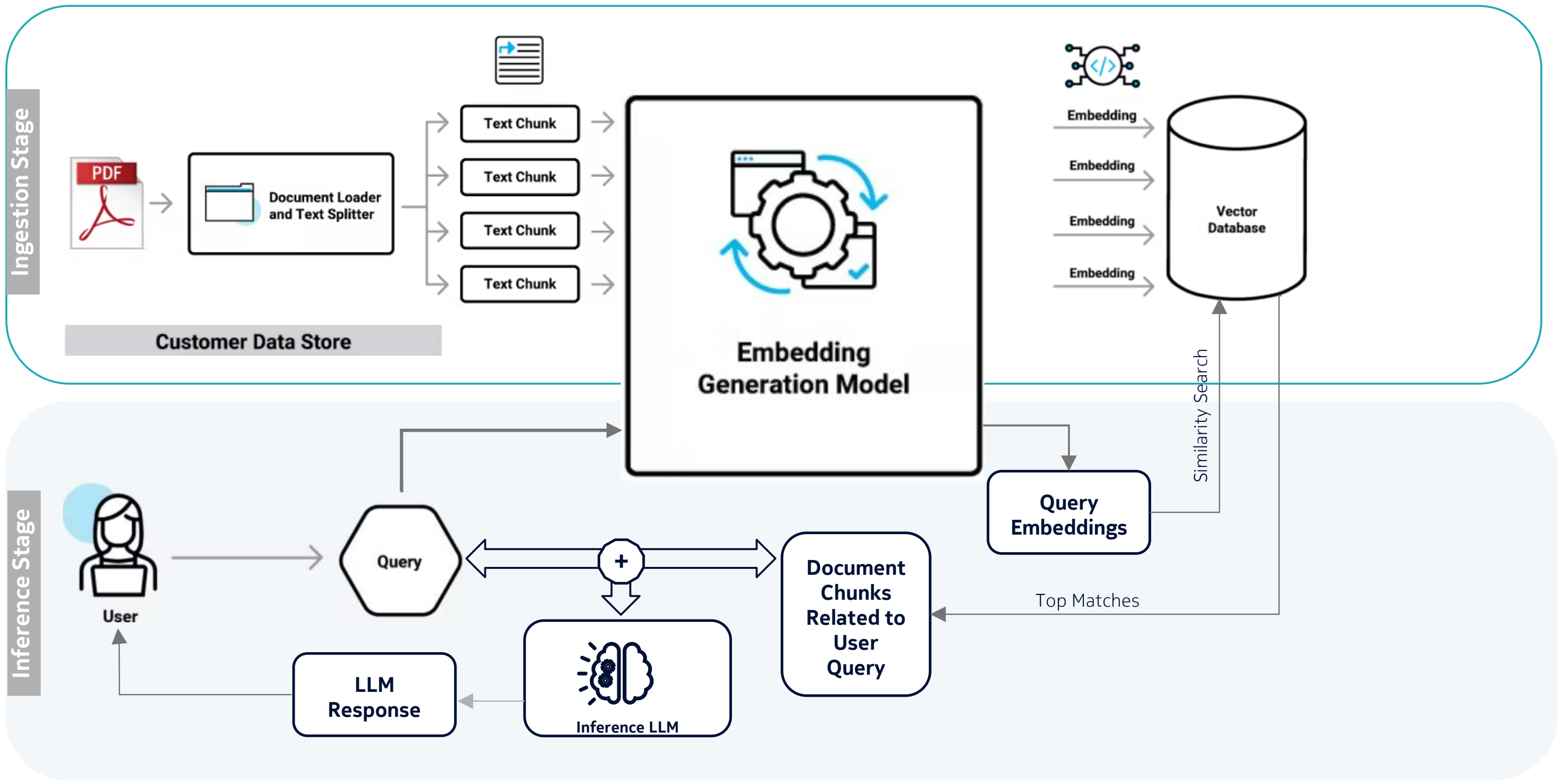
Where?

Commonly applied in **customer service chatbots, enterprise knowledge retrieval, AI search engines, and research assistants** that require factual grounding.

How?

RAG first **retrieves** the most relevant data from a **vector database or document store** and then **uses an LLM** to generate an informed response, improving accuracy and reducing hallucinations.

RAG Implementation

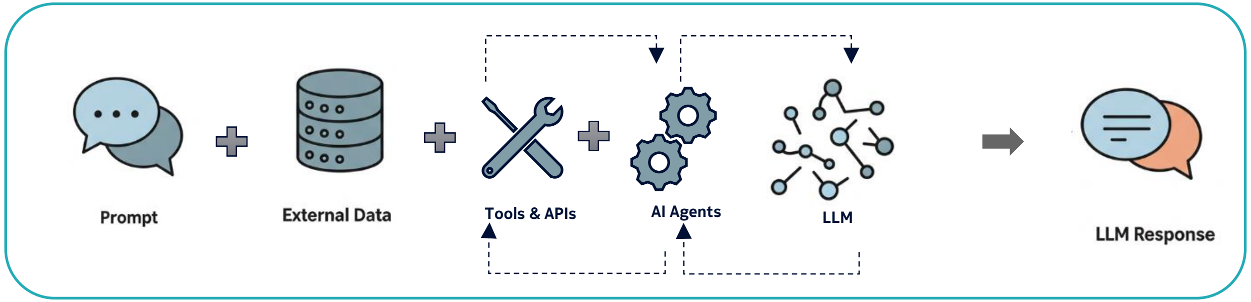


DEMONSTRATION

AGENTIC AI

Agentic AI

Extending LLM capabilities with AI Agents



- Agentic AI is a software system designed to interact with data and tools in a way that requires minimal human intervention.

- Agentic AI is sometimes referred to as autonomous AI. This is because it has the capability to communicate and collaborate with other AI systems and digital infrastructures on behalf of a human user or another AI agent

- To bring agentic AI to practice, you create a system that provides an LLM with access to external tools and algorithms that supply instructions for how the AI agents should use those tools

Agentic AI – System Design

Agent

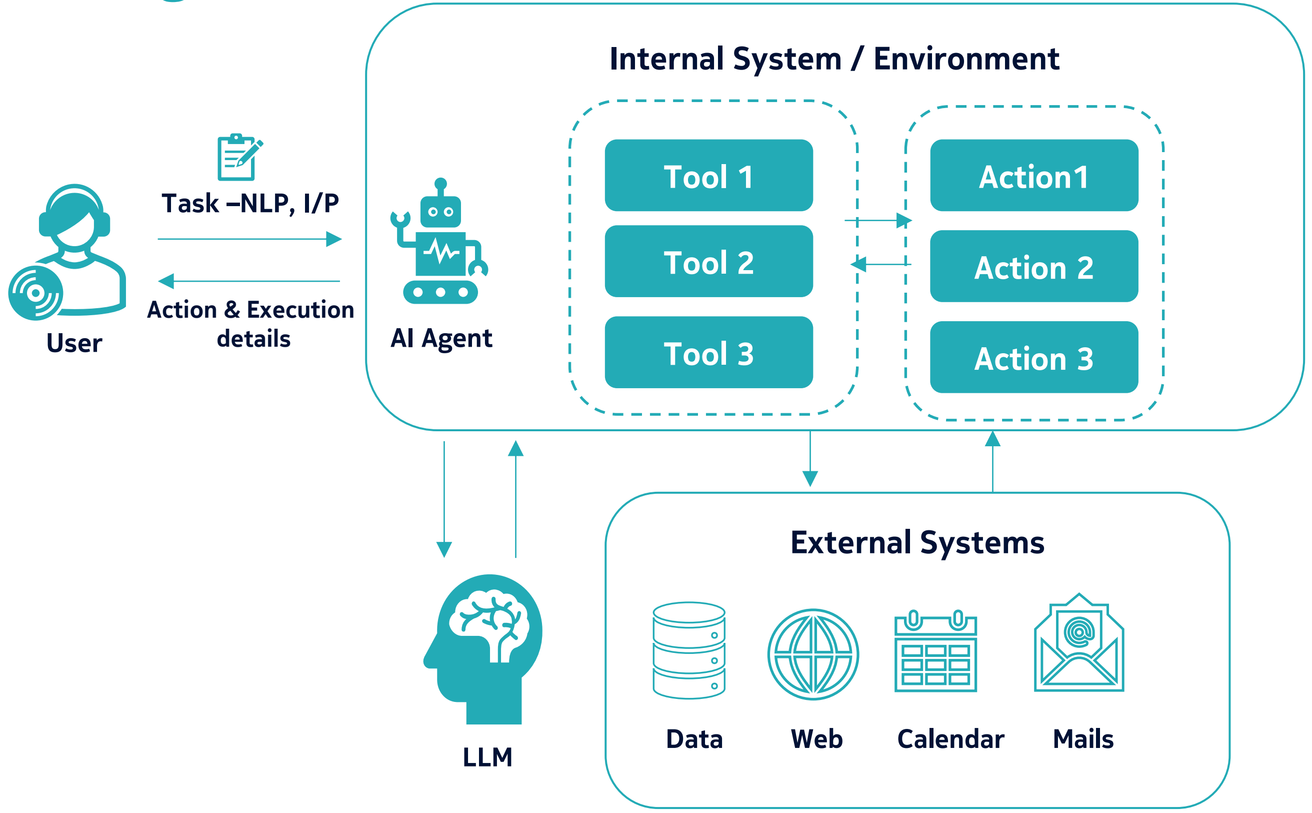
A software program that uses artificial intelligence (AI) to perform tasks on behalf of a user

Tools

A "Tool" refer to functions or APIs that enable the agent to interact with the external world or another software program

Task

A "task" refers to a specific, predefined goal or objective that an AI agent is programmed to achieve through interacting with its environment



Agentic AI – Real Life Examples

1. **Travel Planner –** Plans itinerary, logistics, accommodations
2. **Customer Service Bot** – Handles support queries and processes refunds.
3. **AI Research Assistants** – Fetches and summarizes data for users.
4. **Automated Coding Agents** – Writes, tests, and debugs code.
5. **Smart Assistants (like Jarvis)** – Plans and executes tasks proactively.

Q & A

Thank You

Endorsements

https://www.linkedin.com/in/jkishoreraj/

Kishoreraj Jayakumar
Technical Specialist - Nokia | Full Stack Engineer | Gen AI Enthusiast

Survey

https://forms.office.com/e/AkyCePZqbU