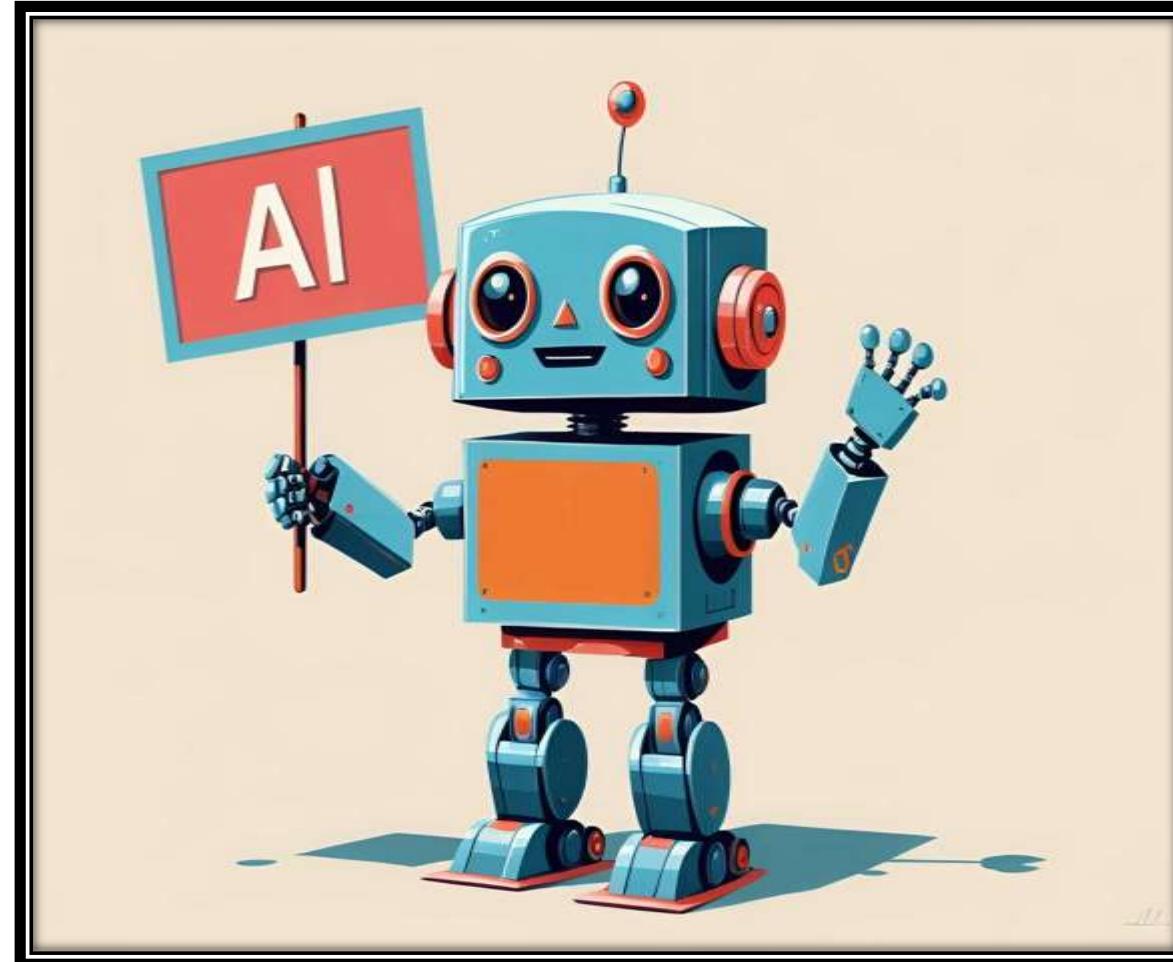


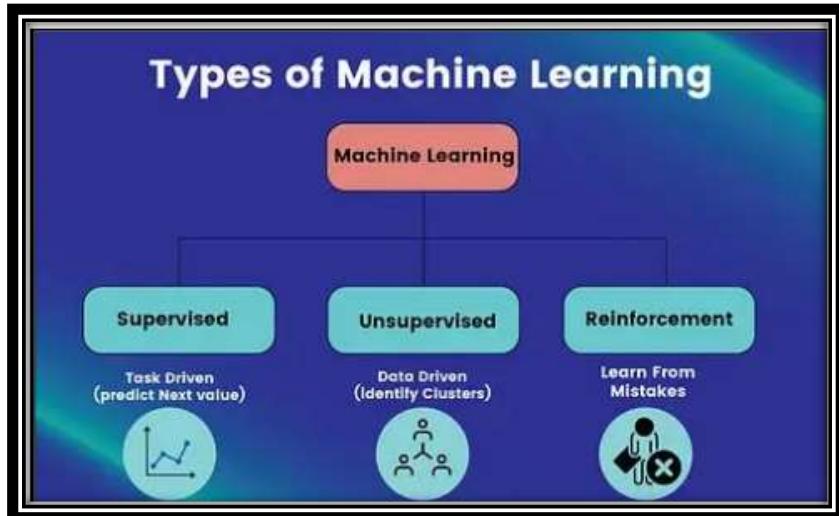
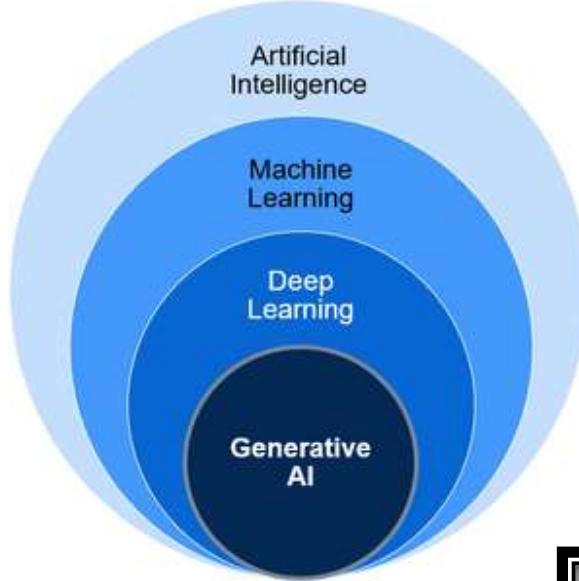
Generative AI with Bedrock, LangChain, RAG, Python, Streamlit, and Chatbot Development



Section 1 : Introduction to AI



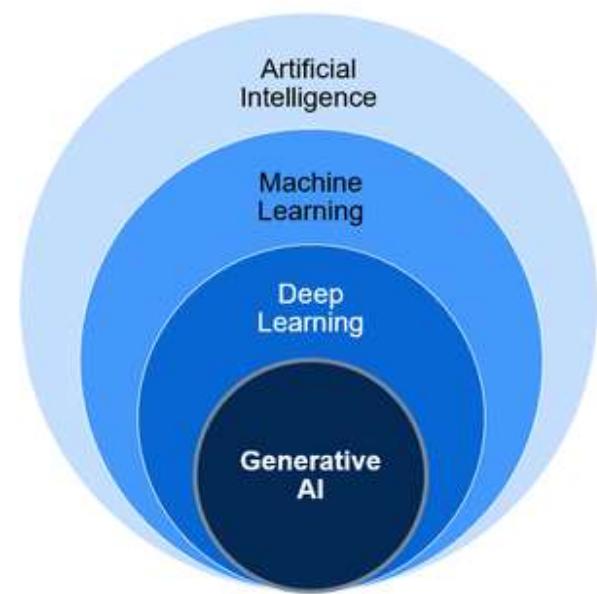
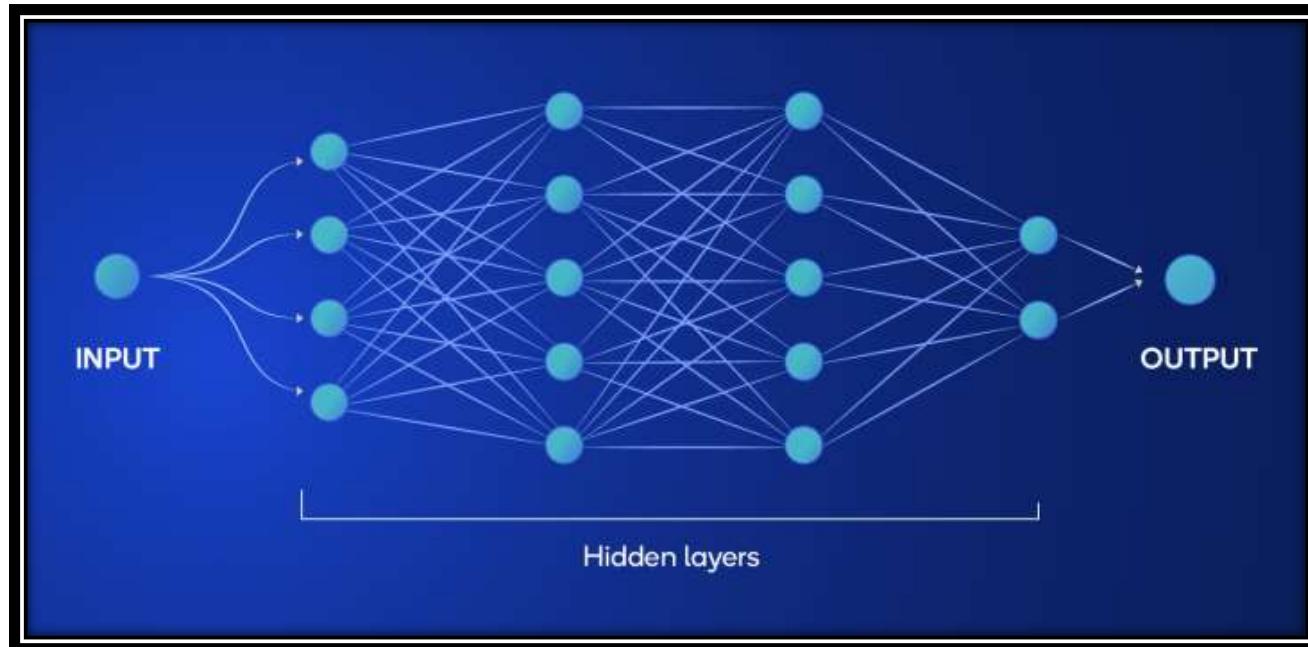
- ❑ Introduction to AI
- ❑ Real-World Applications of AI
- ❑ Machine Learning Overview
- ❑ Machine Learning Applications
- ❑ AI and ML: Understanding Their Relationship
- ❑ Types of Machine Learning: Supervised Learning



Section 2 : Foundations of Deep Learning



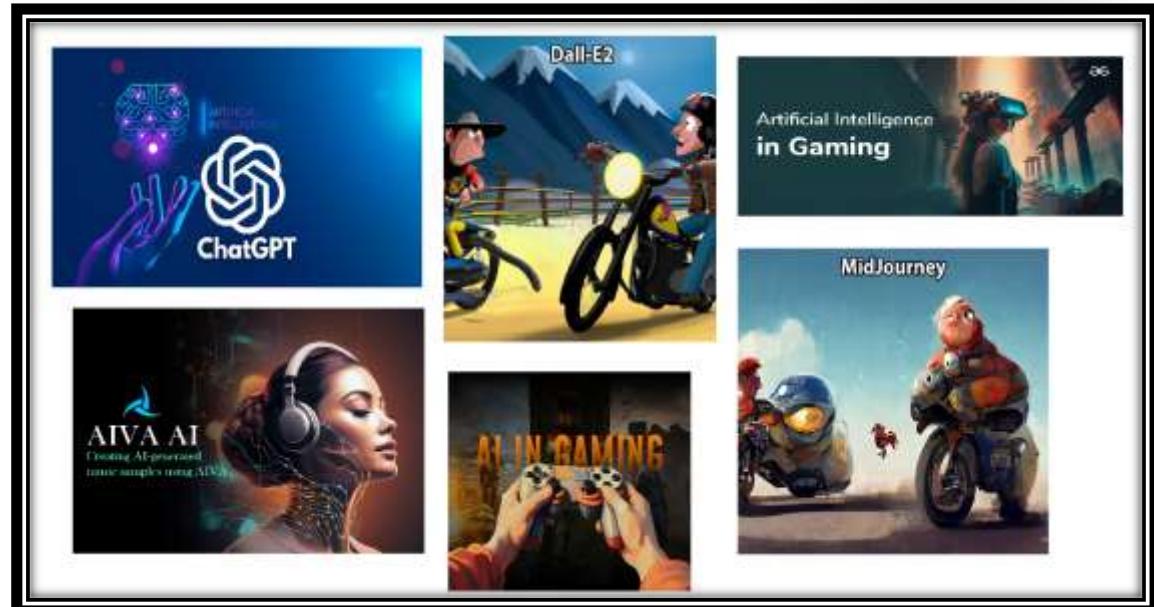
- ❑ Introduction to Deep Learning
- ❑ Deep Learning, AI and ML
- ❑ Neural Network



Section 3 : Introduction to Generative AI and Its Applications

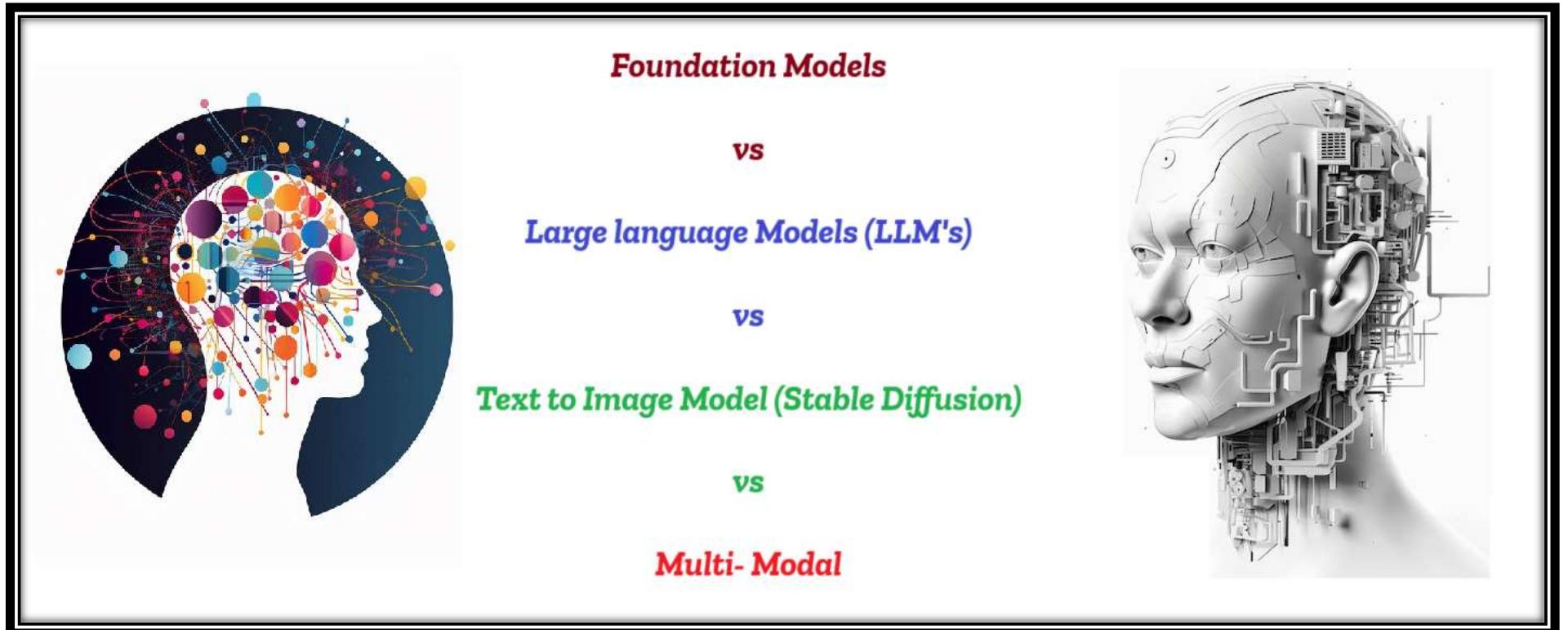


- ❑ Introduction to Generative AI
- ❑ Real-World Application of Generative AI
- ❑ Benefits of Generative AI
- ❑ Relationship Between AI, ML, DL and Generative AI



Section 4 : Foundation Models, LLMs, Text-to-Image, and Multimodal AI

- ❑ Introduction to Foundation Models
- ❑ LLM, Text-to-Image Models
- ❑ Multimodal Models



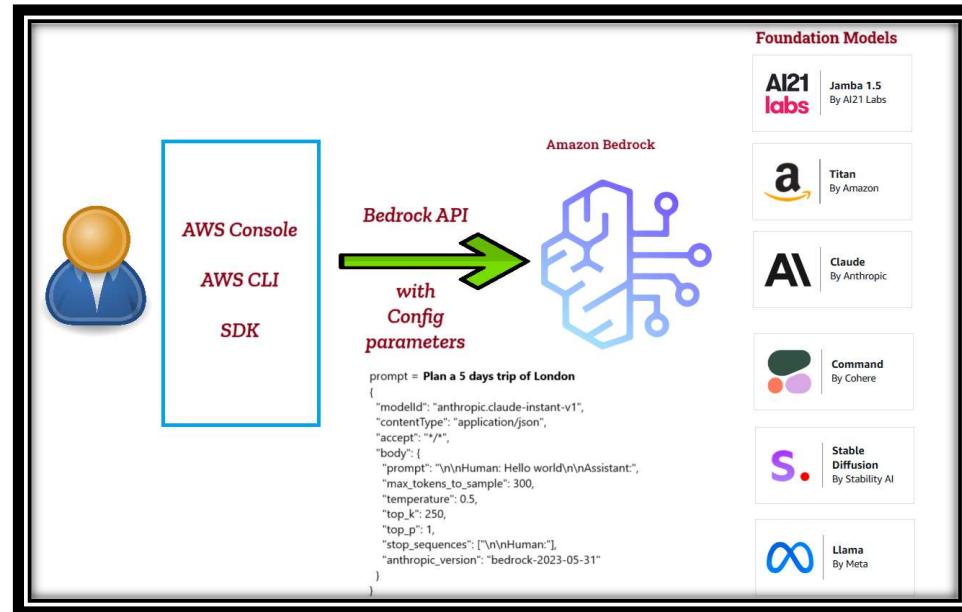
Section 5 : Amazon Bedrock and Foundation Models: An In-Depth Exploration

- ❑ Introduction to Amazon Bedrock
- ❑ How Amazon Bedrock Works?
- ❑ Foundation Models in Amazon Bedrock
- ❑ Various Foundation Models via Amazon Bedrock



Section 6 : Exploring Amazon Bedrock Console and Features

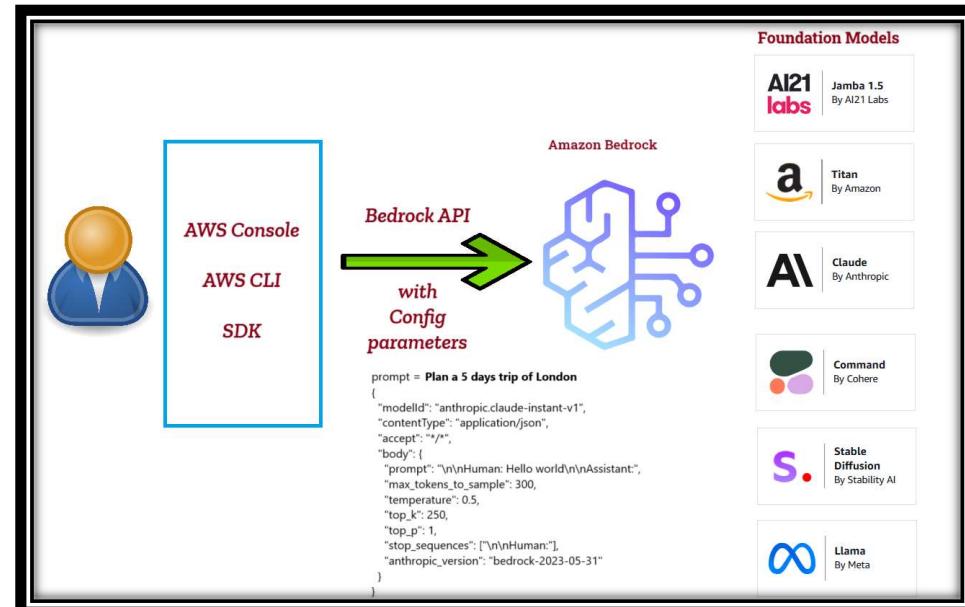
- ❑ Amazon Bedrock Console
- ❑ Playgrounds Feature in Amazon Bedrock
- ❑ Builder Tools Features in Amazon Bedrock
- ❑ Safeguard Feature in Amazon Bedrock
- ❑ Model Access in Amazon Bedrock



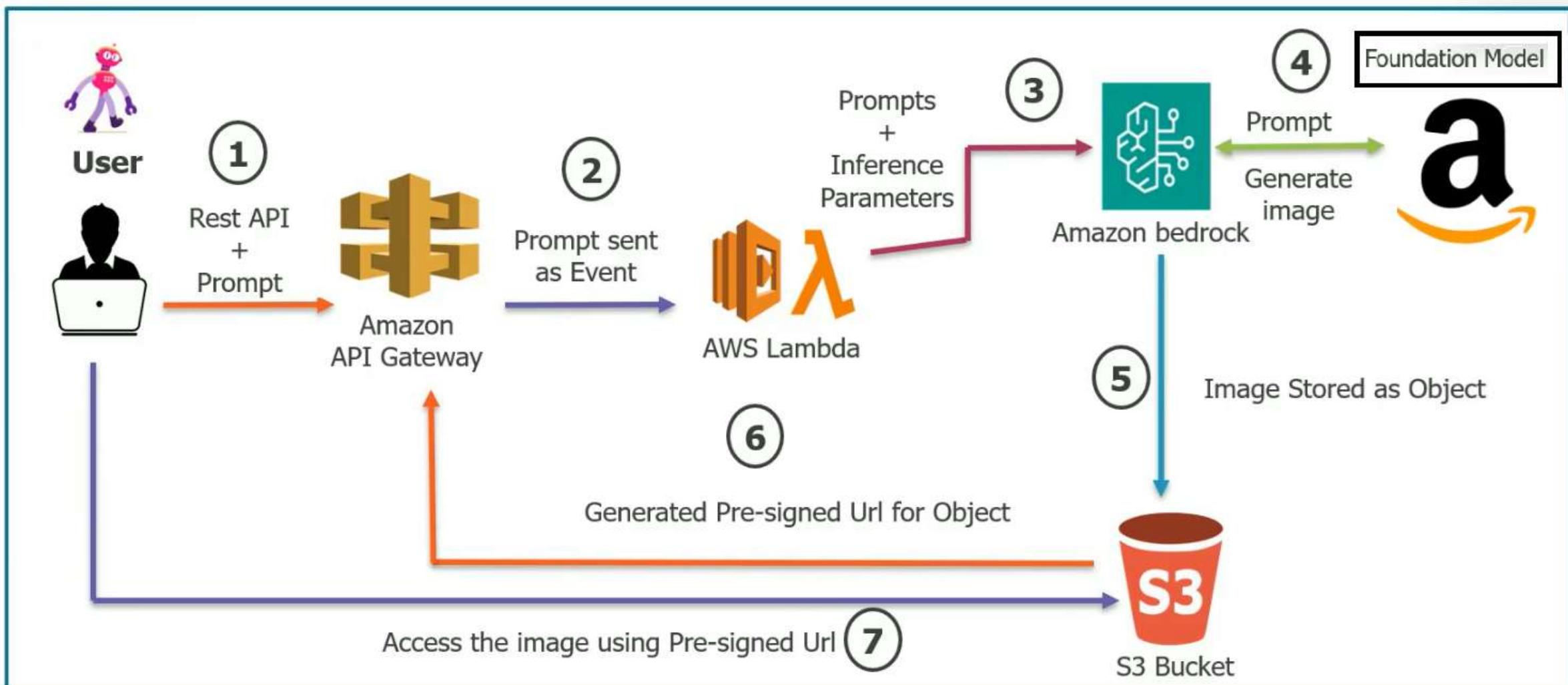
Section 7 : Inference Parameters of Foundation Models



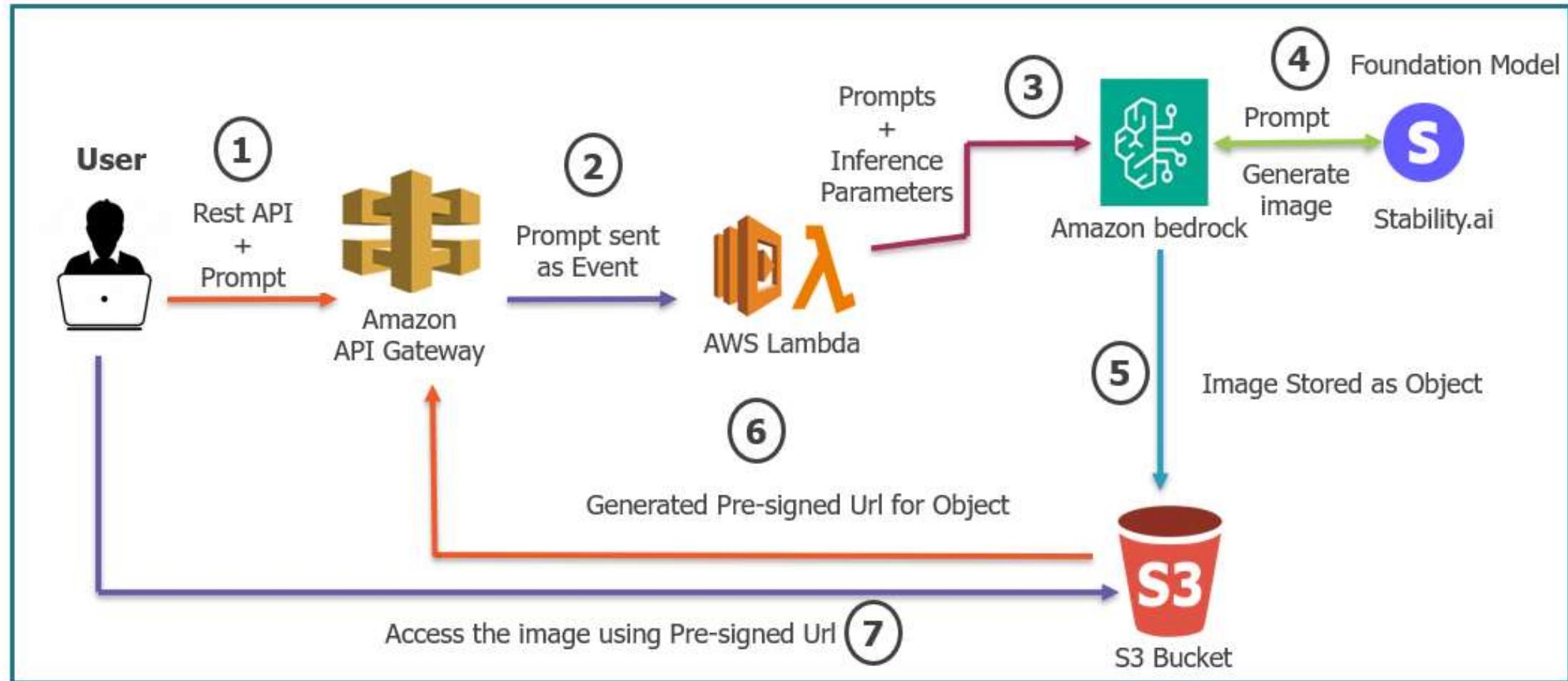
- ❑ Randomness and Diversity
- ❑ Temperature, Top P, Top K & More
- ❑ Length Control: Response Length, Stop Sequence, & Length Penalty



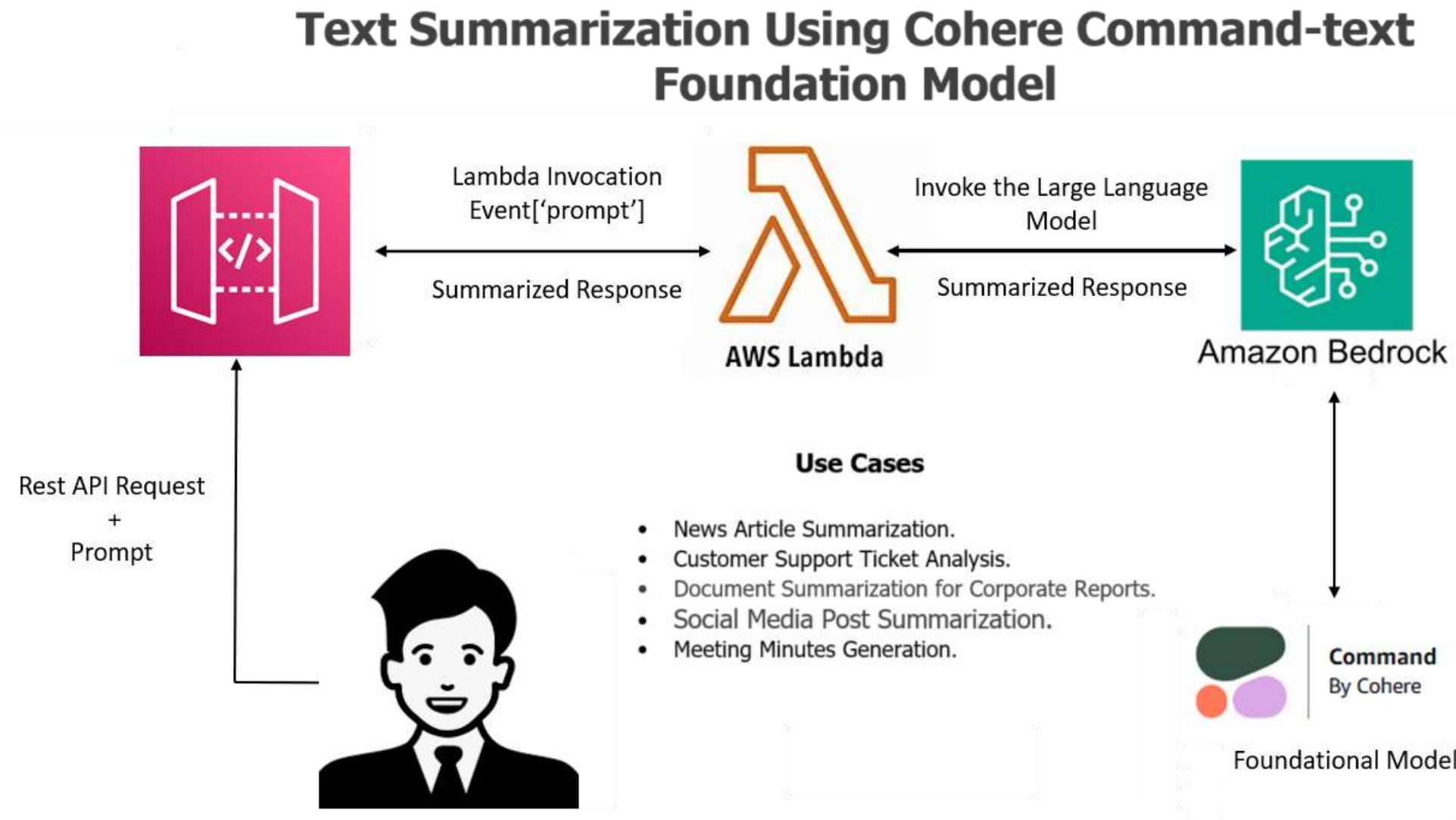
Section 8 : Generative AI Use Case: Text-to-Image Generation with Lambda and Amazon Model



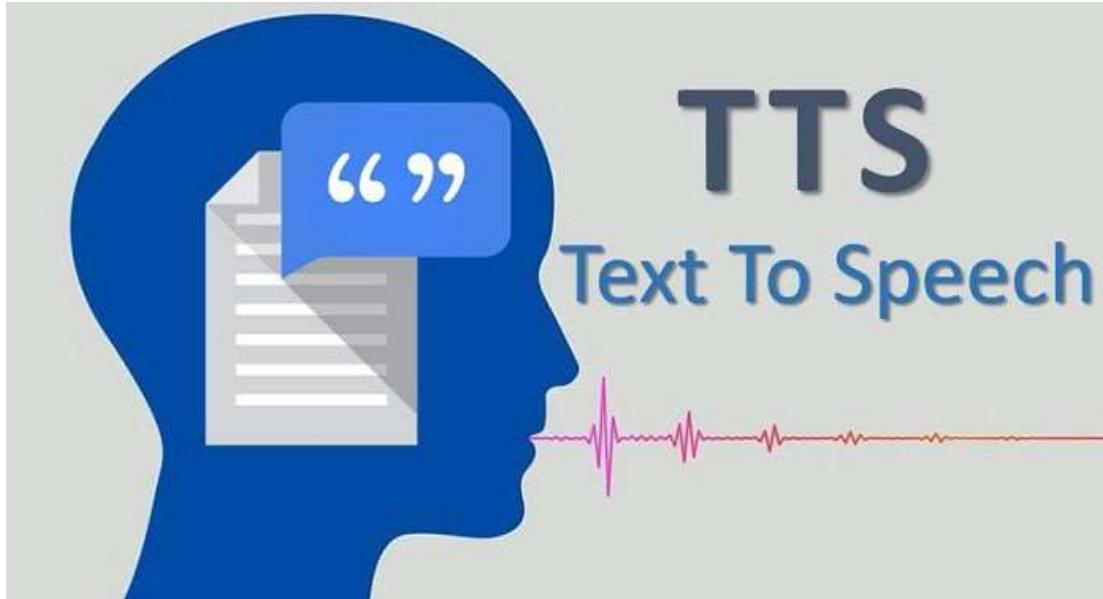
Section 9 : Generative AI Use Case: Text-to-Image Generation with Lambda and Stable Diffusion



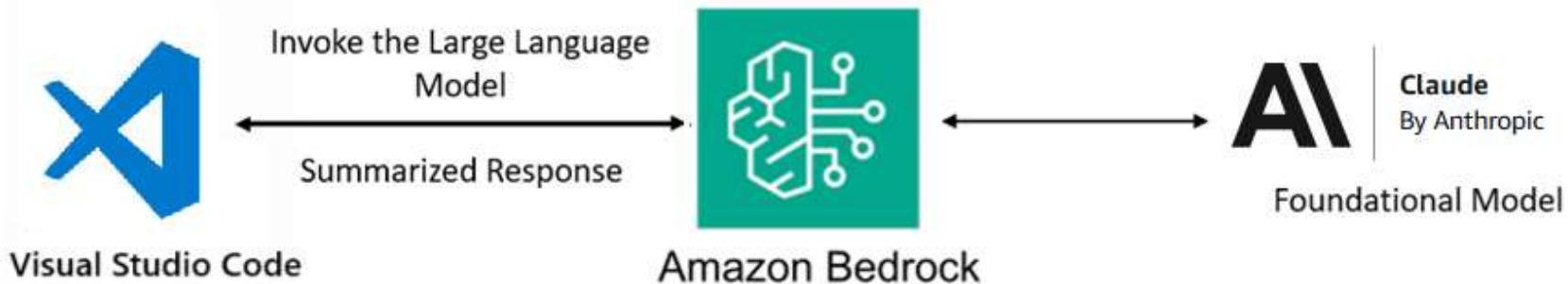
Section 10 : Use Case: Text Summarization Generation Using Cohere Command - Text FM



Section 11 : Project - Text2Speech Player

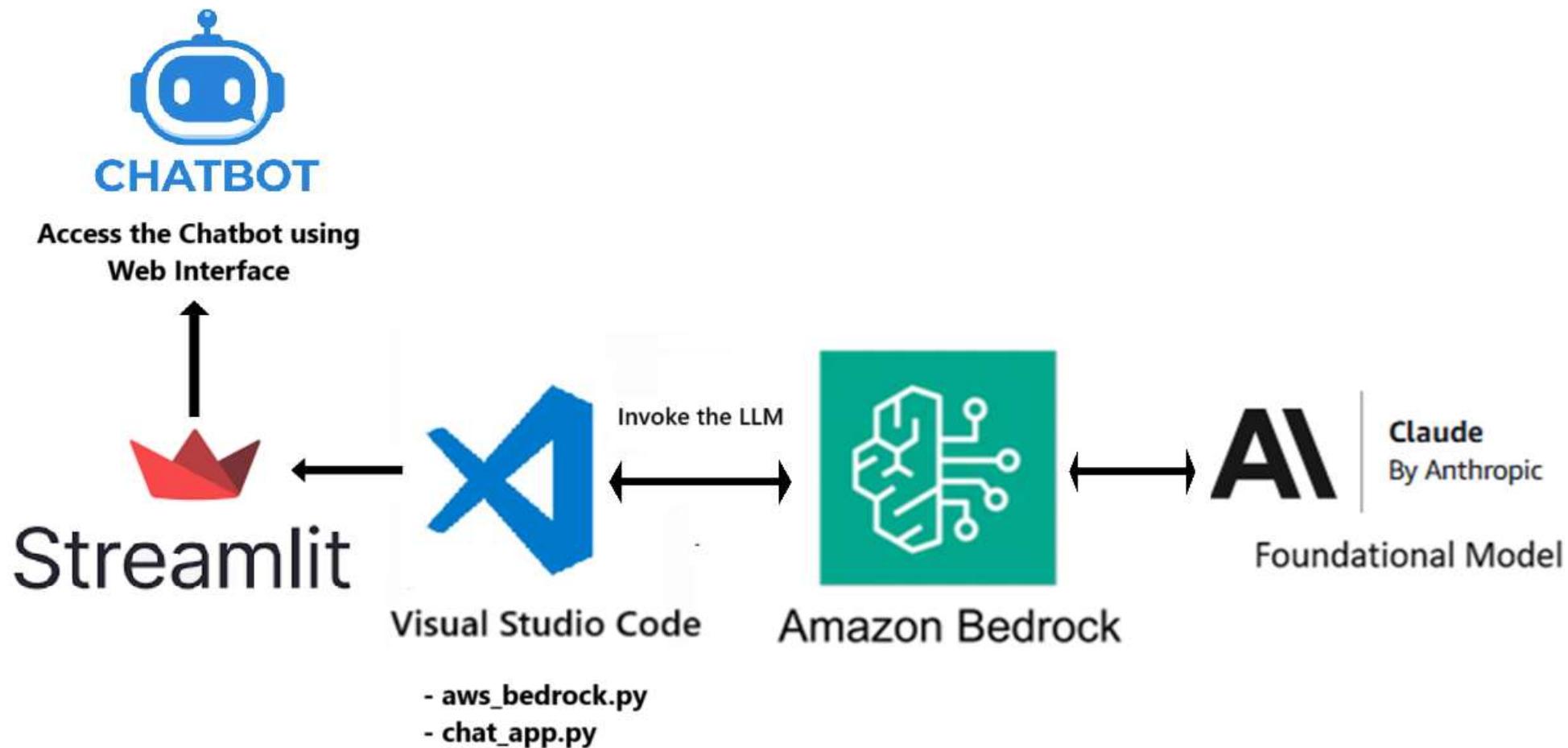


Section 12 : Building a Python-Based Chatbot with AWS Bedrock and Anthropic Claude FM

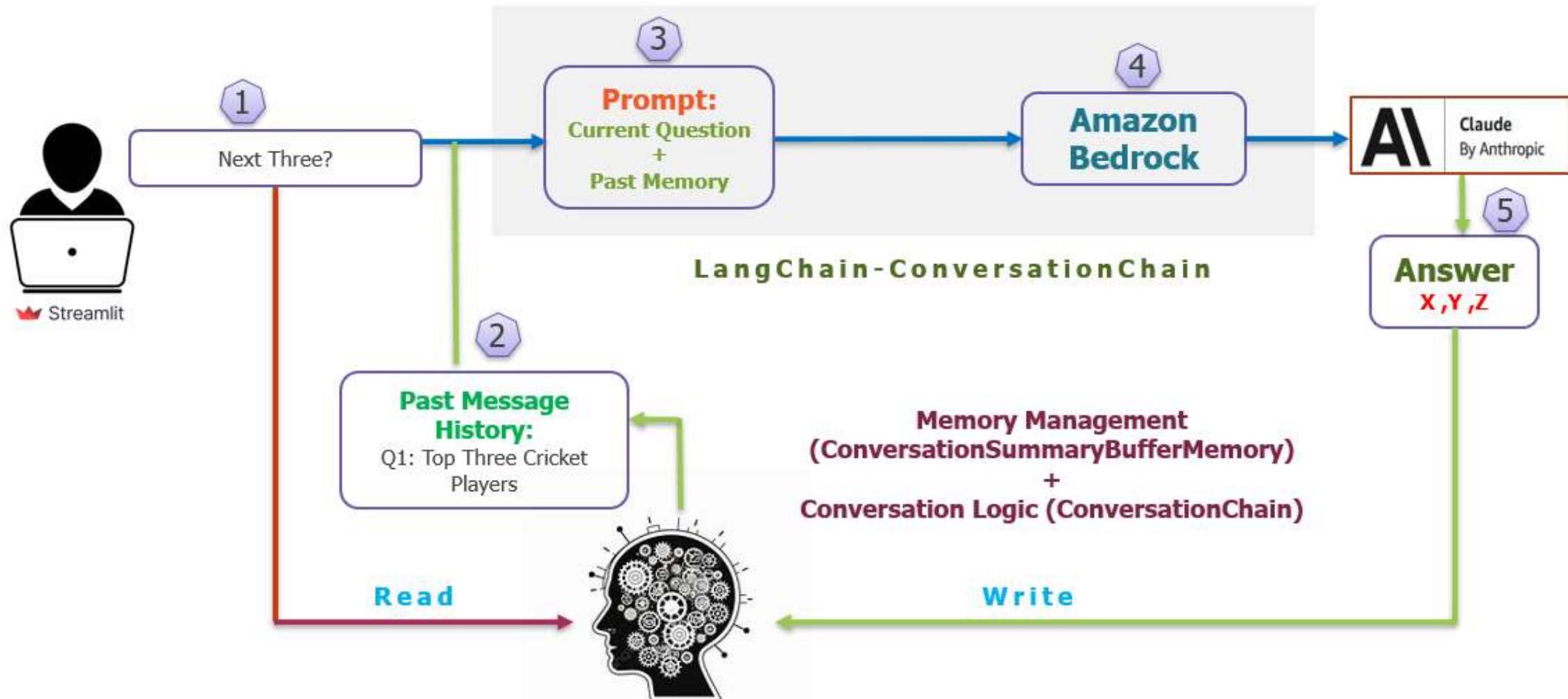


- **User Input:** Text entered by the user.
- **API Request:** Input is sent to AWS Bedrock for processing via the Claude model.
- **Response:** Claude processes the input and sends back a generated response.
- **Output:** Displayed to the user through the chatbot interface.

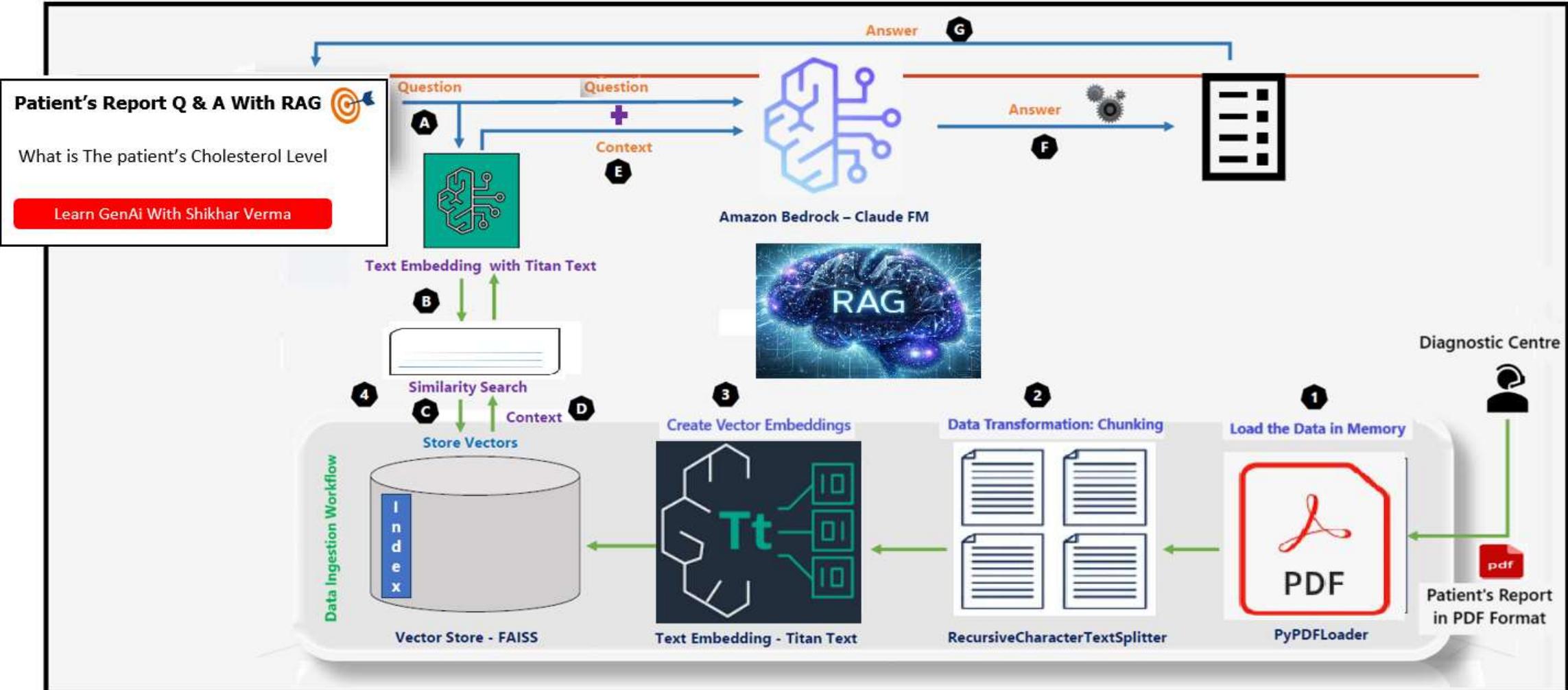
Section 13 : Streamlit-Based Python Chatbot with AWS Bedrock and Anthropic Claude



Section 14 : LangChain-Driven Streamlit Chatbot Using Python, AWS Bedrock, Anthropic Claude



Section 15 : Patient Q&A Using RAG, Amazon Bedrock Claude, LangChain, and Patient Report



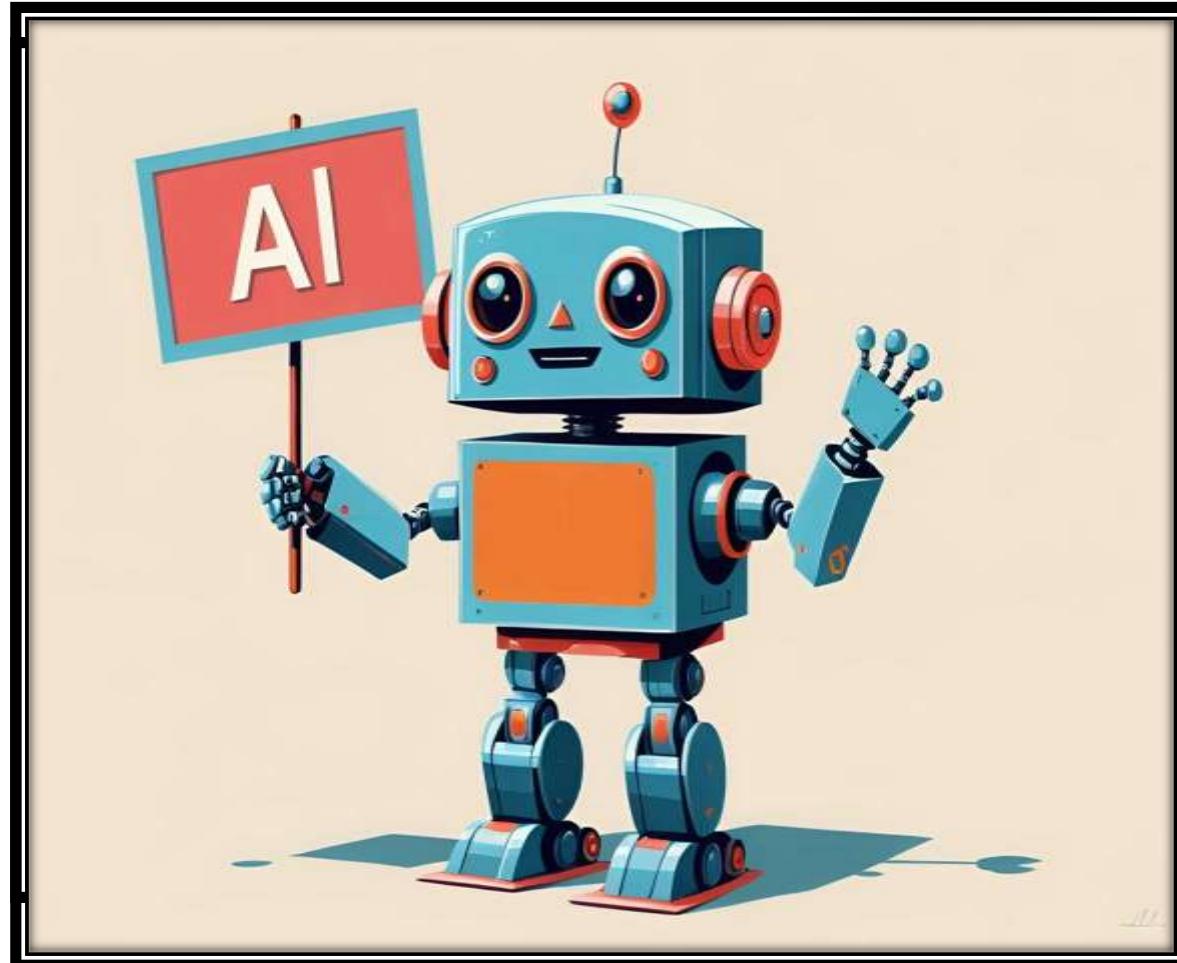
Artificial Intelligence



- **AI** refers to the simulation of human intelligence in machines.
- **AI** includes a broad range of technologies that enable machines to perform tasks that typically require human intelligence, such as understanding natural language, recognizing patterns, solving problems, and making decisions.

□ Example:

- **Voice Assistants:** A virtual assistant like Siri or Alexa can understand spoken commands, answer questions, and perform tasks like sending reminders or controlling smart home devices.



Artificial Intelligence



□ Example:

- **Organizing Photos:** our phone groups photos by recognizing faces, places, and things, so you can find what you're looking for faster.
- **Netflix and Spotify:** AI suggests movies, shows, and songs you might like based on what you've watched or listened to before.
- **Content Filtering:** AI helps filter out inappropriate content to keep social media safe.
- **Product Suggestions:** Sites like Amazon show you products similar to what you've browsed, based on your shopping habits.
- **Fraud Alerts:** AI detects unusual account activity and can alert you if there's potential fraud.
- **Fitness Tracking:** Apps like Fitbit and Apple Health track your steps, sleep, and exercise, using AI to provide health tips.



Artificial Intelligence



□ Example:

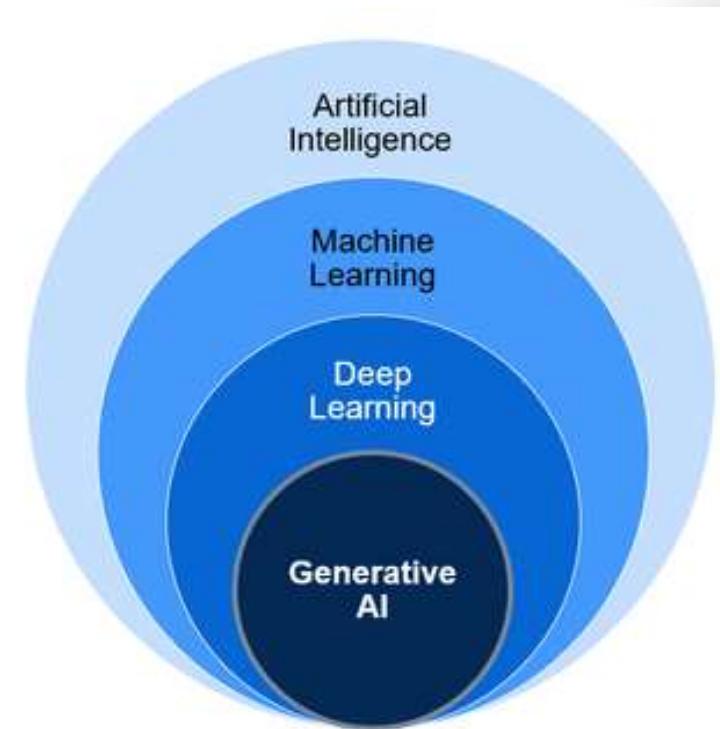
- **Security Cameras:** Smart cameras can recognize faces and detect movement, notifying you if anything unusual happens.
- **Chatbots:** Many websites have chatbots that answer questions instantly, helping with simple tasks like checking order status or answering FAQs.
- **Automated Calls:** When you call customer support, AI can sometimes help you with basic inquiries or route your call to the right department.
- **Spam Filters:** AI sorts junk emails into the spam folder automatically.
- **Translation:** Apps like Google Translate use AI to translate languages, making travel and cross-language communication easier.
- **Speech-to-Text:** AI converts spoken words into text for those who need help typing or prefer speaking.



Machine Learning



- **Machine Learning** is a subset of AI that focuses on the development of algorithms that allow computers to learn from and make predictions based on data.
- Machine learning is a way for computers to learn from examples and data instead of being directly programmed. Imagine teaching a computer by showing it many pictures of cats and dogs. Over time, it starts to recognize the difference on its own, even with new pictures.
- **For Example:**
- A recommendation system on platforms like Netflix or Amazon uses ML algorithms to analyze your viewing or purchasing habits and suggest content or products that you might like. Over time, as it gathers more data about your preferences, it refines its recommendations.

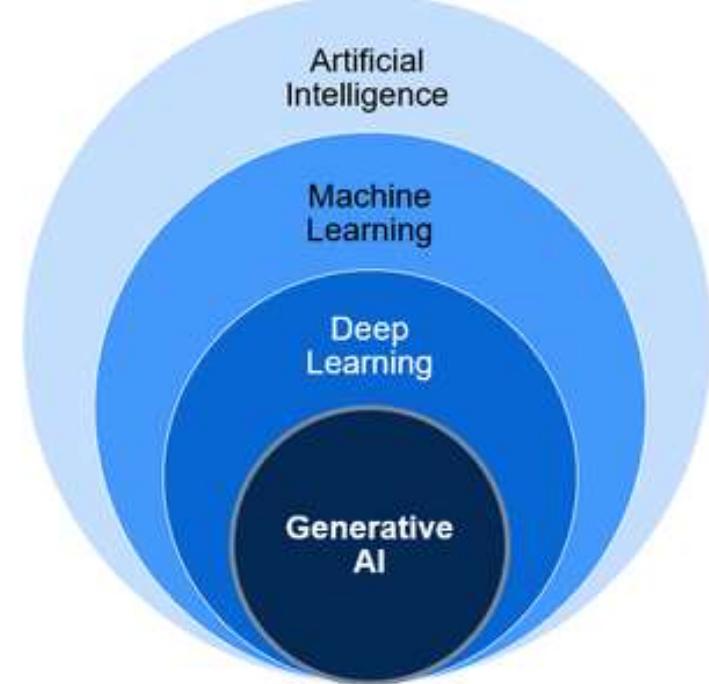


Machine Learning

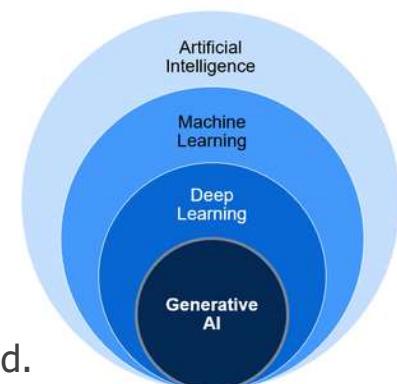
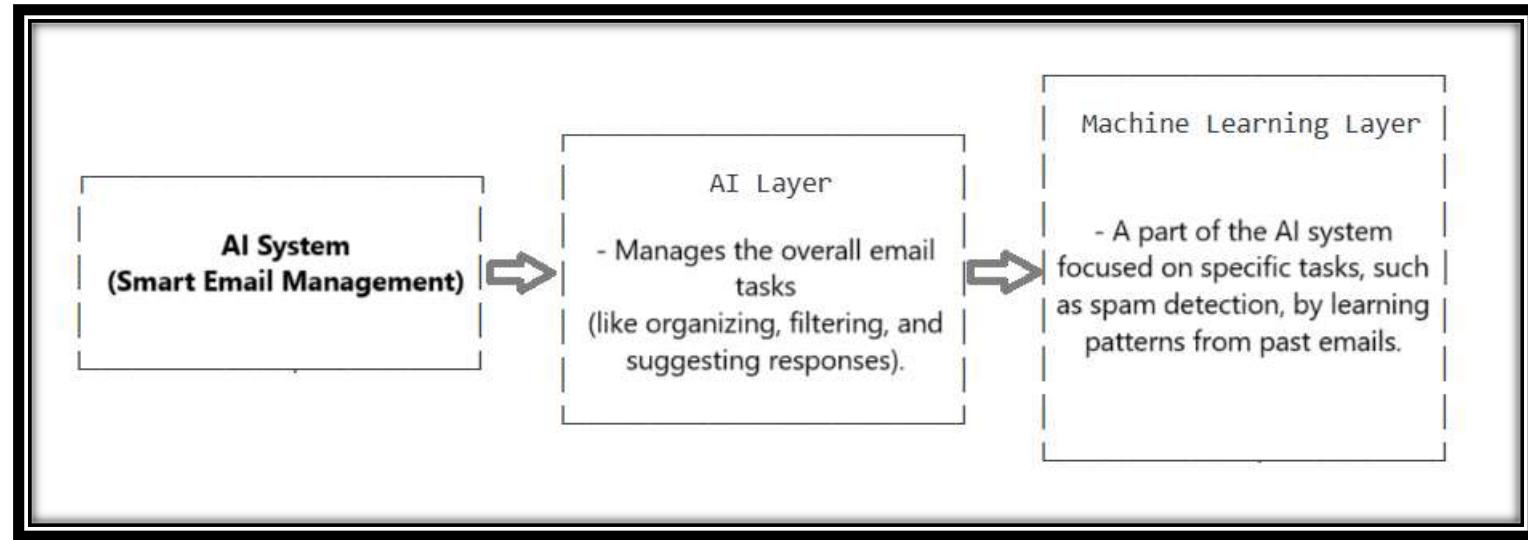


□ Few Examples of Machine Learning:

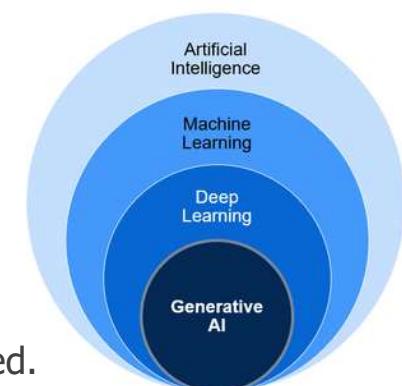
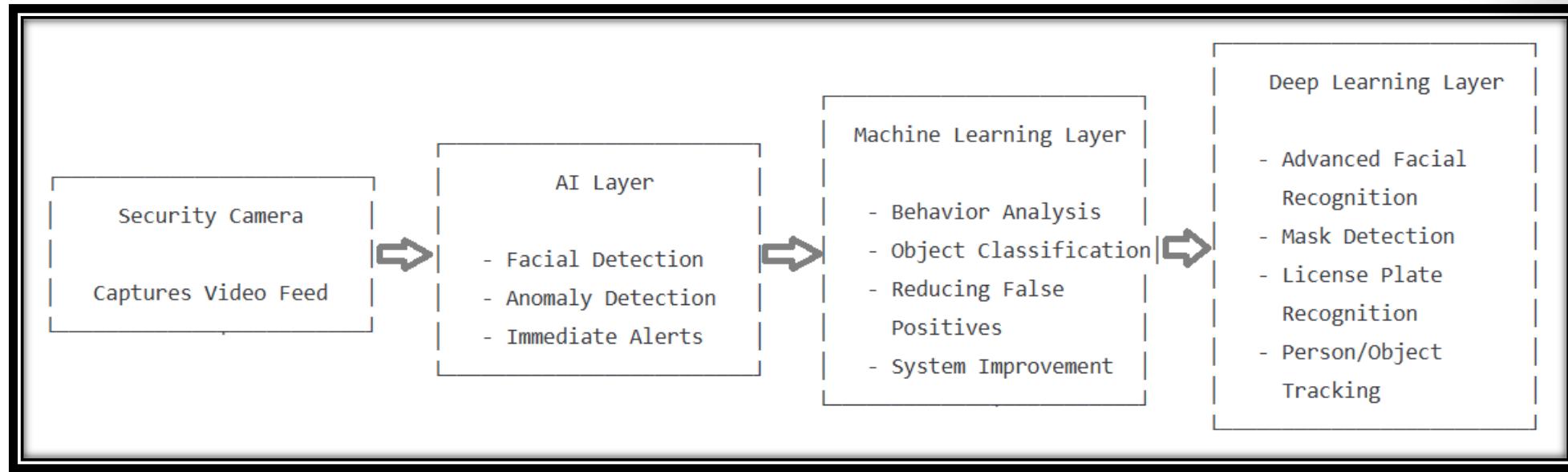
- Spam Filtering in Email
- Face Recognition
- Customer Support Chatbots
- Language Translation
- Self-Driving Cars
- Voice Assistants



Relationship Between AI and ML



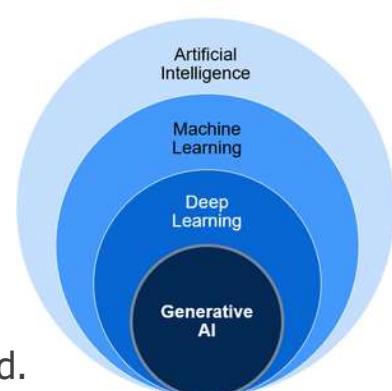
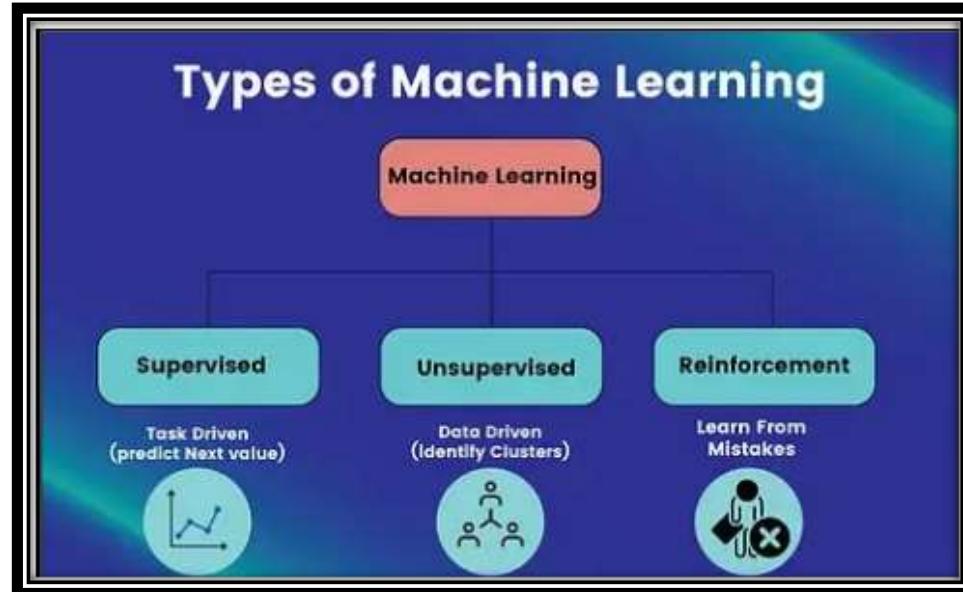
Relationship Between AI and ML



Types of Machine Learning



- **Machine learning (ML)** is a type of technology that allows computers to learn from data and make decisions or predictions without being explicitly programmed. It's like teaching a computer how to recognize patterns and make choices based on examples, similar to how we learn from experiences.

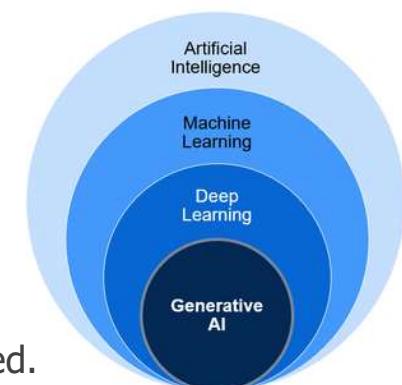
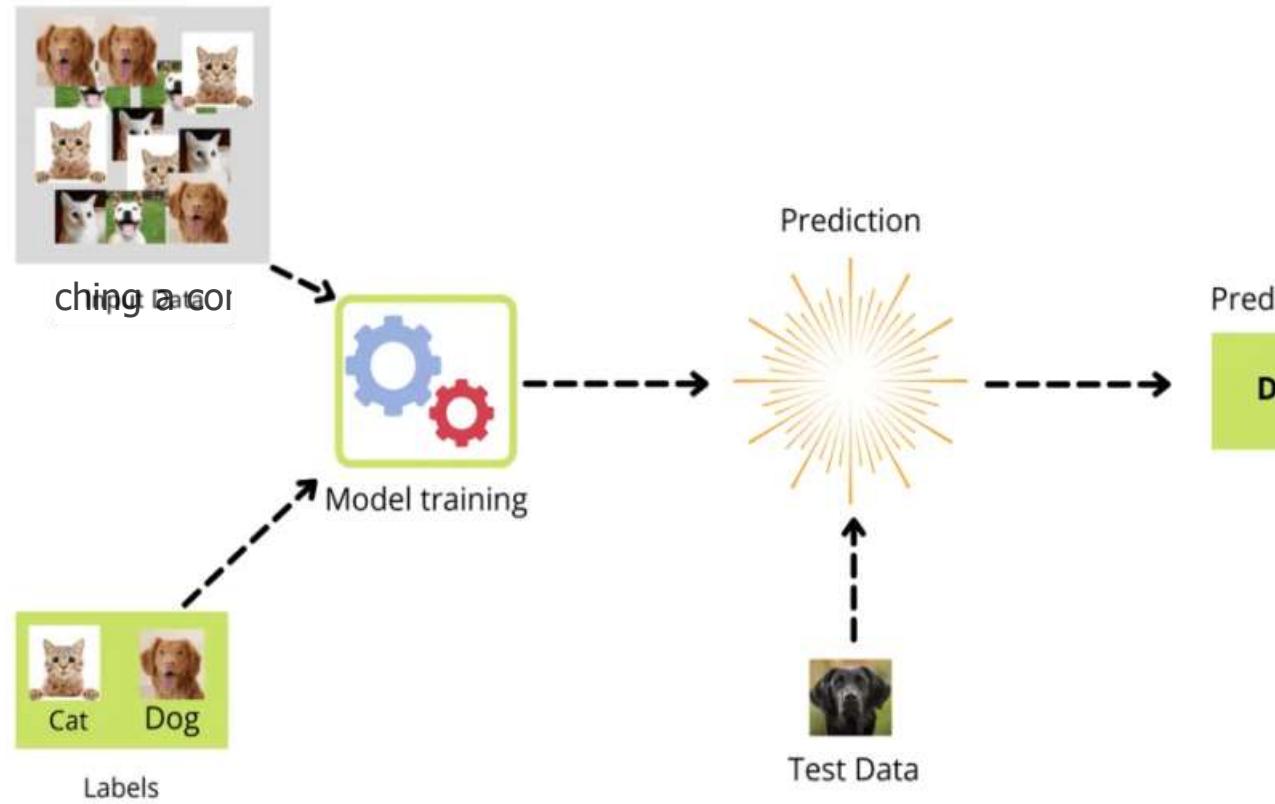


Types of Machine Learning



➤ Supervised Machine Learning:

- **What it is:** In supervised learning, the training data includes both features and labels.
- **How it works:** The machine learns from the training data and then uses this knowledge to make predictions on new, unlabeled data.
- **Example:** Imagine you have a dataset of images labeled "cat" or "dog." After training a model on this data, it can predict whether a new image is a "cat" or "dog."



Types of Machine Learning



➤ Unsupervised Machine Learning:

➤ What it is:

In patterns or stru

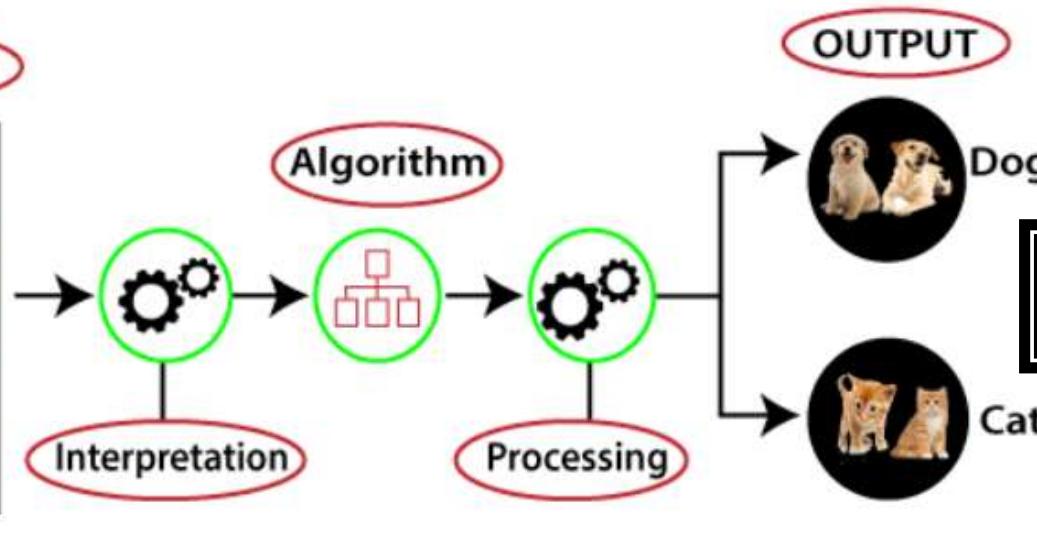
INPUT RAW DATA

➤ How it works

knowing what t

➤ Example:

If yo photos togethe



- Ear Shape: Pointed or Rounded
- Body Size: Small, Medium, or Large
- Tail Length: Short, Medium, or Long
- Face Shape: Round or Angular
- Sound: Whether it meows or barks

- Group 1: Animals with pointed ears, smaller body sizes, round faces, and the sound "meow." This group likely contains mostly cats.
- Group 2: Animals with rounded ears, larger body sizes, angular faces, and the sound "bark." This group likely contains mostly dogs.

Types of Machine Learning

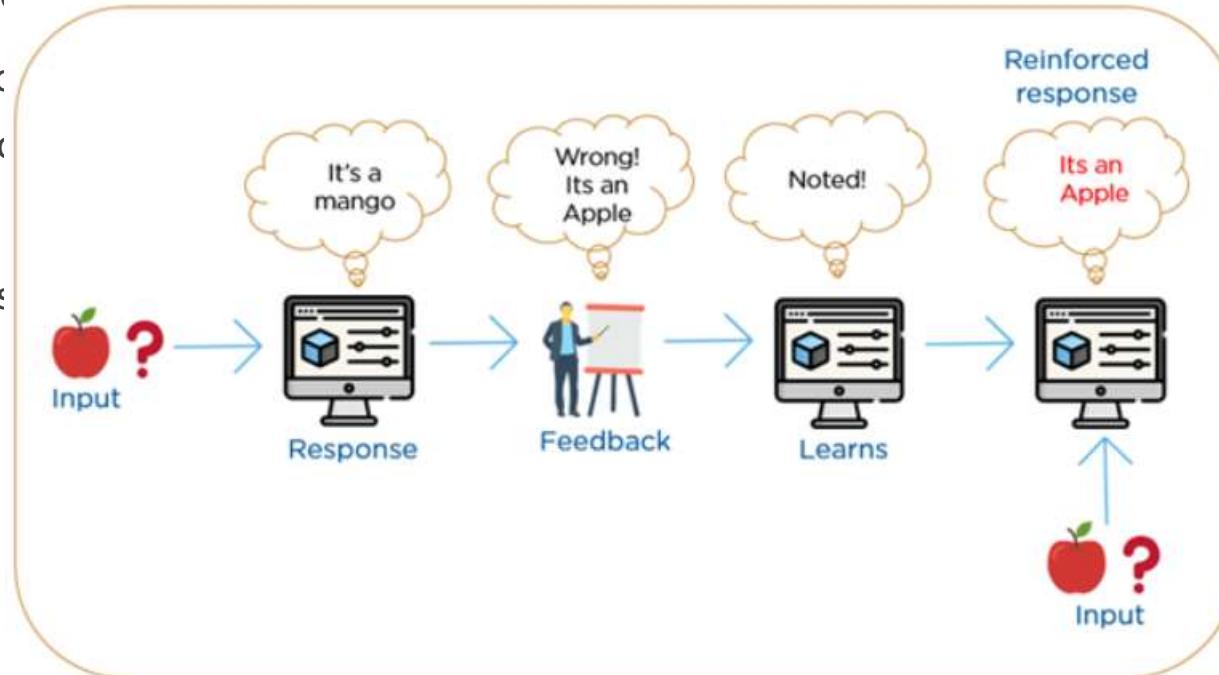


➤ Reinforcement Machine Learning:

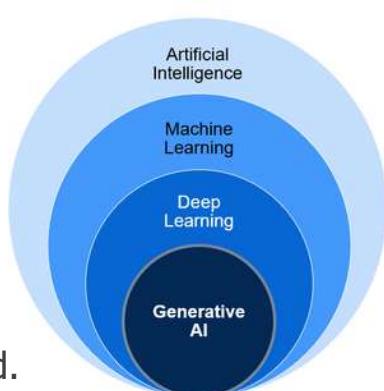
➤ **What it is:** In reinforcement learning, the computer learns by interacting with an environment and receiving feedback in the form of rewards or penalties.

➤ **How it works:** The computer takes an action (Input) and receives a response (Output). This response is compared against the expected output, and the difference is used to calculate a reward or penalty (Feedback). This feedback is used to update the computer's internal model (Learns), which helps it learn which actions are more effective.

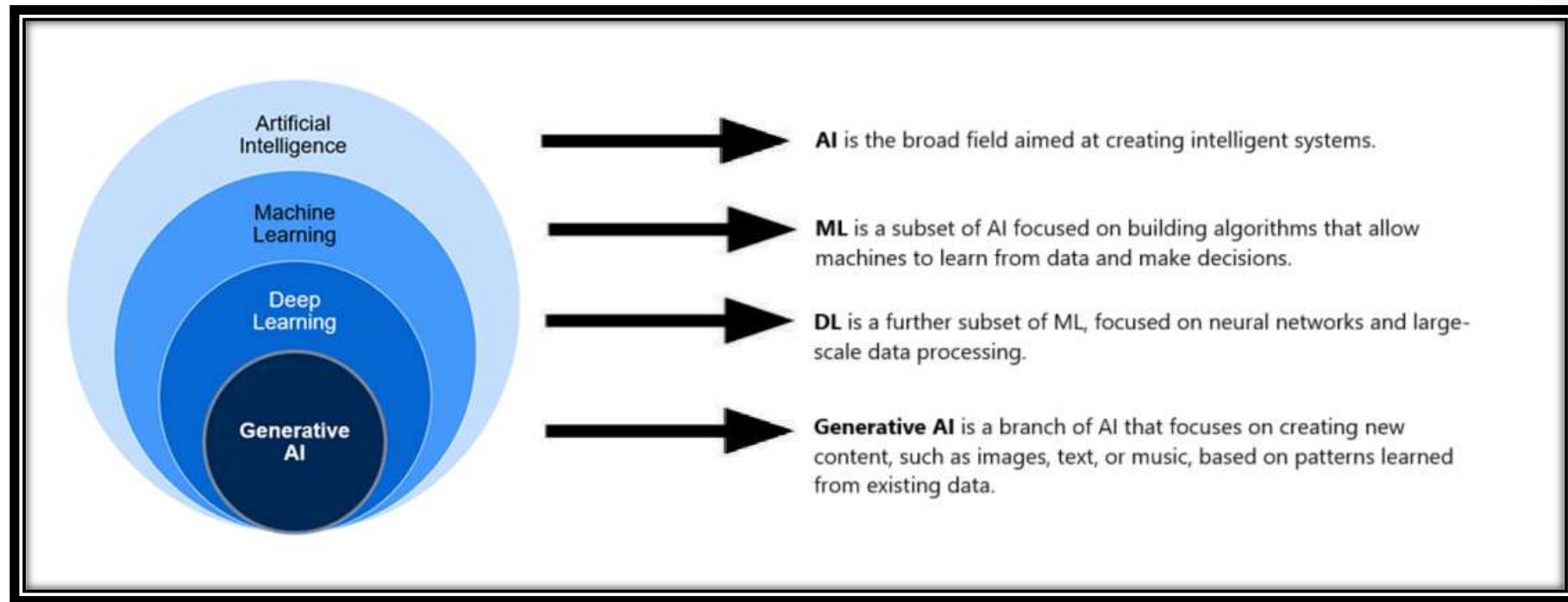
➤ **Example:** Think of a game like chess. As the AI wins or loses, it learns from its mistakes and successes (Feedback) to improve its strategy (Learns).



Its or a score) that
ing on whether it



Deep Learning

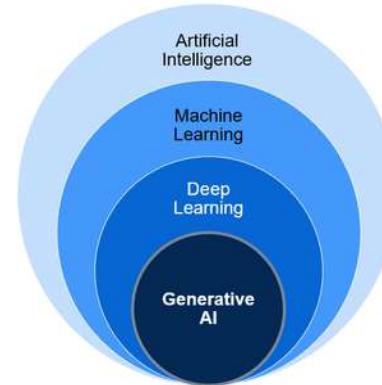


Deep Learning



- **Deep learning**, AI, and machine learning (ML) are closely related fields, with deep learning being a subset of machine learning and machine learning itself being a subset of AI.
- Deep learning is a type of machine learning that uses **neural networks** with many layers (hence the name "deep") to find patterns in large amounts of data.
- Deep learning is particularly powerful when working with unstructured data, like images, audio, and text.
- **For Example –**

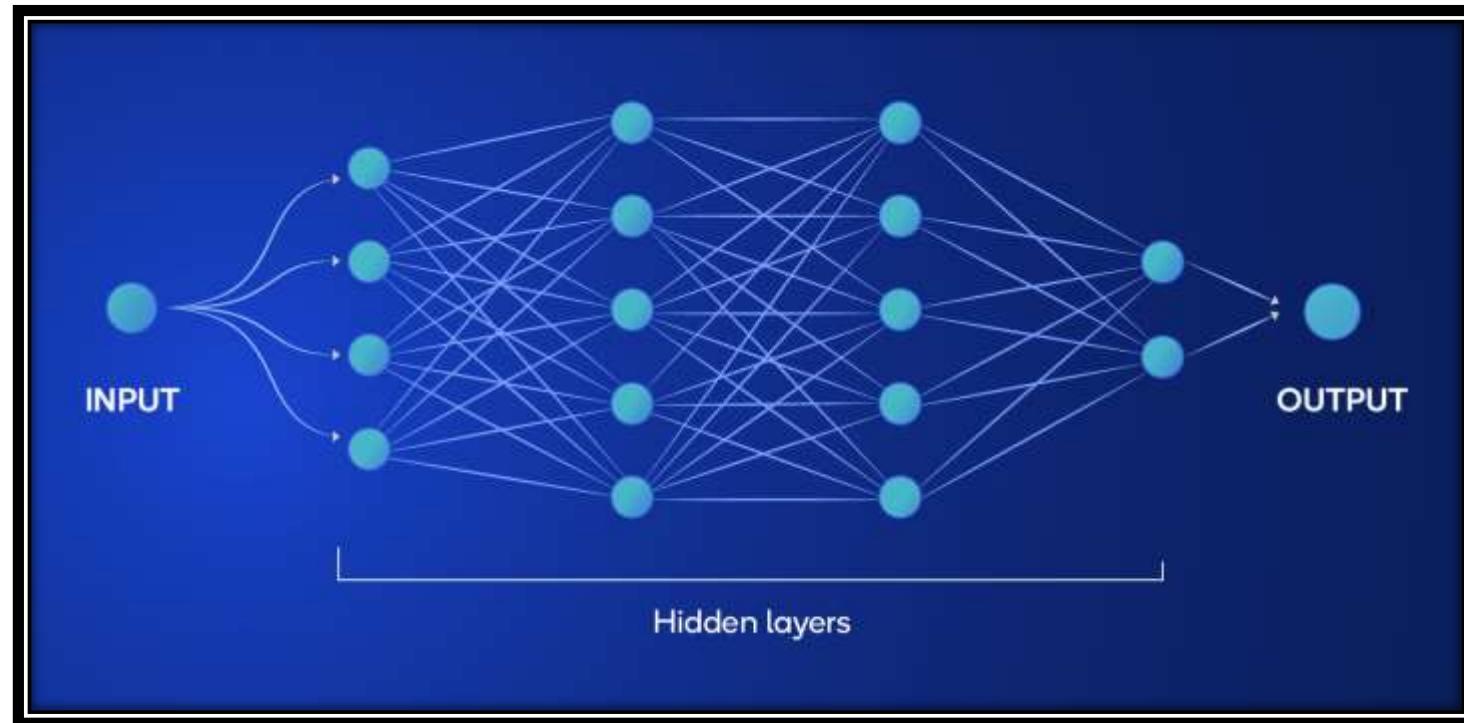
Image recognition in medical diagnostics is a common deep learning application. For example, a deep learning model trained on thousands of X-ray images can be used to detect tumors or diseases. It can process and analyze visual data, helping doctors identify conditions that may be hard to spot with the human eye.



Deep Learning



- A **neural network** in deep learning is a type of computer system designed to mimic how the human brain works.
- Just like our brain processes information using neurons, a neural network processes data using **artificial neurons**.
- These neurons are connected to each other in layers, and each connection has a weight that helps the system learn from data.

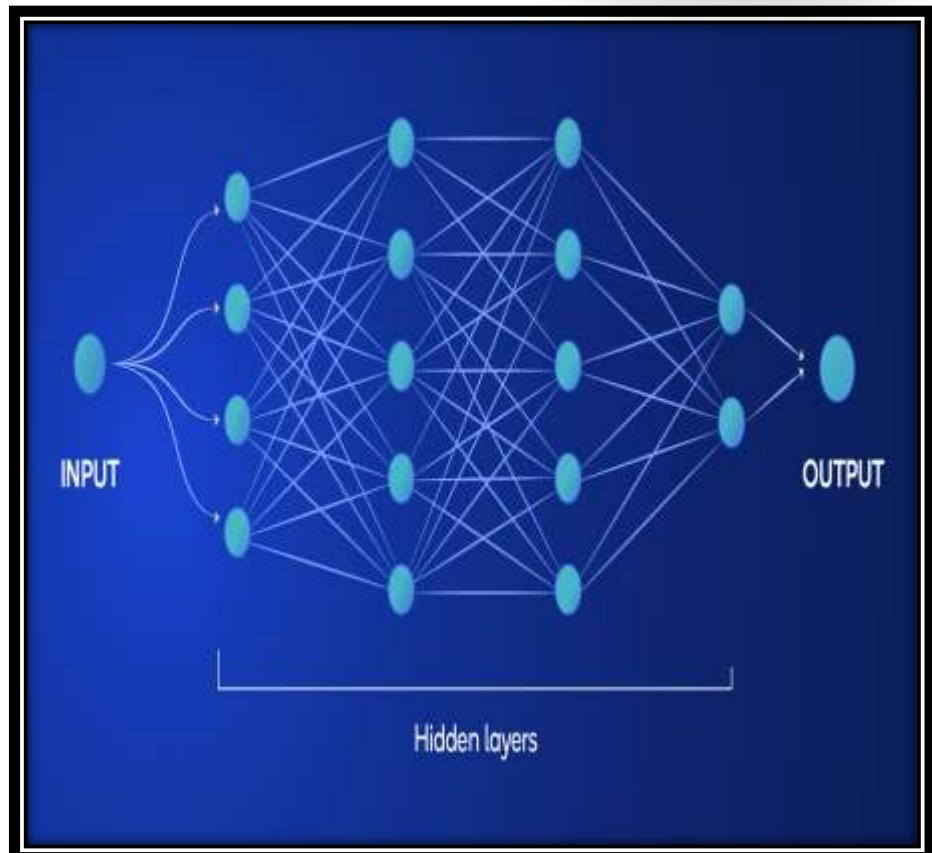


Deep Learning

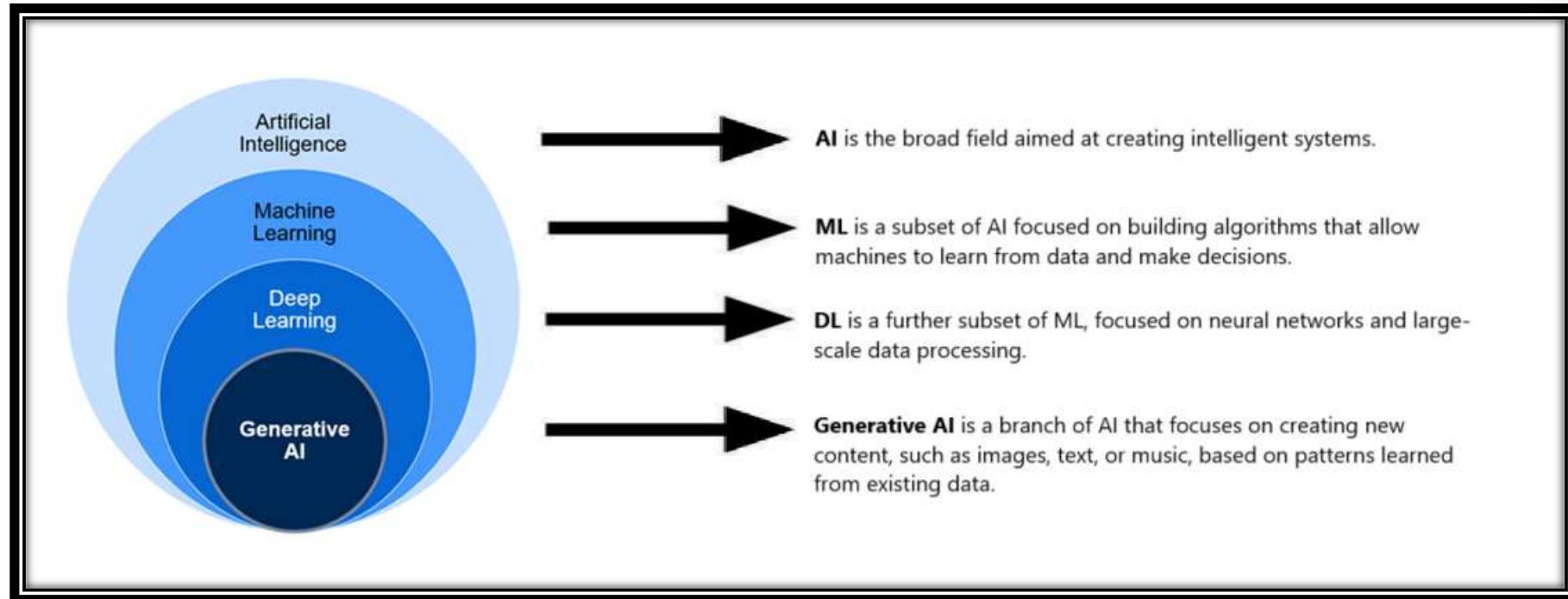


□ Basic Structure:

- **Neurons:** Think of these as tiny decision-makers. Each neuron takes input (data), processes it, and sends the result to the next layer of neurons.
- **Layers:**
 - **Input Layer:** This is where the data enters the network (e.g., an image, text, or numbers).
 - **Hidden Layers:** These layers do the work of processing the input data, learning from it, and figuring out patterns. The more hidden layers, the "deeper" the network, which is why it's called deep learning.
 - **Output Layer:** This is where the network gives its final result (e.g., whether an image is a cat or a dog).
- **Connections (Weights):** Neurons in one layer are connected to neurons in the next layer. Each connection has a weight that helps the system learn how to process information. If a connection has a higher weight, the neuron it connects to is more important.



Relationship between AI, ML, DL, and Generative AI





Types of Machine Learning

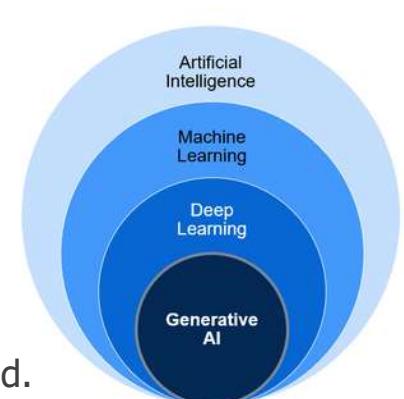
➤ Supervised Machine Learning

Dataset with Features and Labels:

- **Features** are the input variables that the model uses to make predictions (e.g., "Hours Studied").
- **Labels** are the known outcomes or the target variable that we want to predict (e.g., "Pass" or "Fail").

Example dataset:

Hours Studied	Result (Label)
1	Fail
3	Pass
5	Pass



Types of Machine Learning



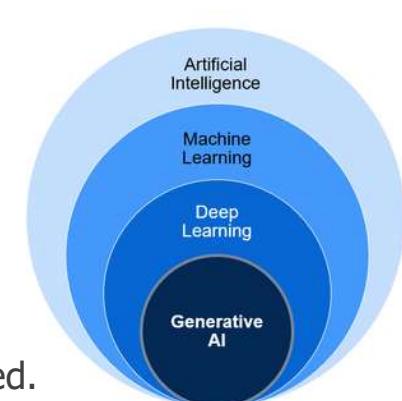
➤ Supervised Machine Learning:

Dataset with Features and Labels:

- **Features** are the input variables that the model uses to make predictions (e.g., "Hours Studied").
- **Labels** are the known outcomes or the target variable that we want to predict (e.g., "Pass" or "Fail").

Example dataset:

Hours Studied	Result (Label)
1	Fail
3	Pass
5	Pass

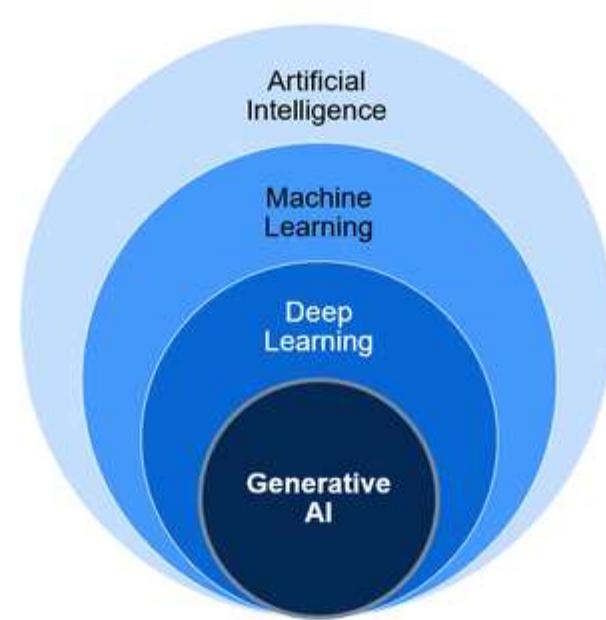


Generative AI



□ Introduction to Generative AI

- Generative AI is a type of artificial intelligence.
- It creates new content, such as text, images, music, and video.
- Learns patterns from existing data and generates unique outputs.

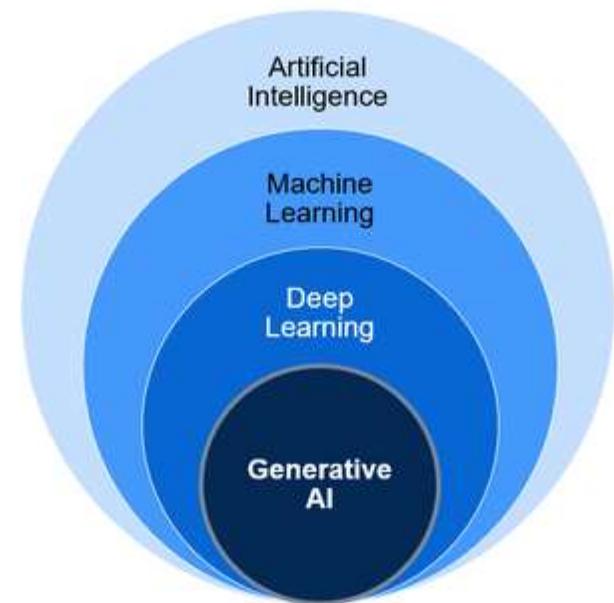


Generative AI



□ How Does Generative AI Work?

- Trained on large datasets to recognize patterns.
- Uses machine learning models to produce new content.
- Mimics creativity by blending learned data in new ways.

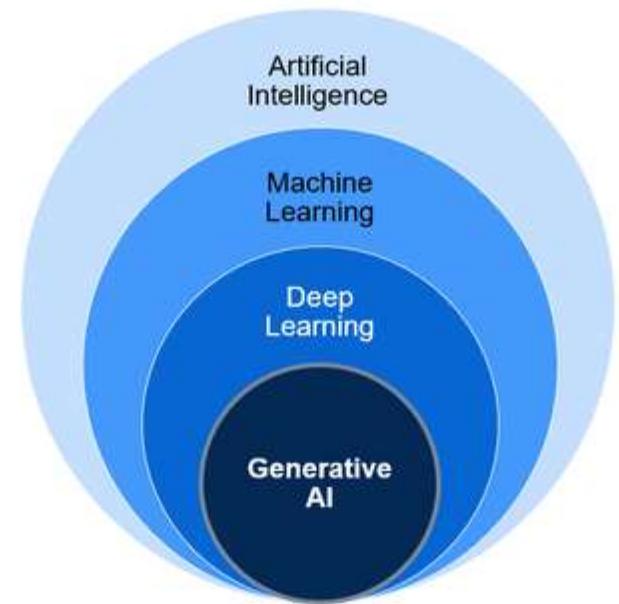


Generative AI



□ Real-World Application of Generative AI

- **Text Generation:** Chatbots and virtual assistants (e.g., ChatGPT).
- **Image Creation:** Tools like DALL-E, Midjourney for generating images.
- **Music Composition:** AI-generated music like AIVA in various styles.
- **Gaming:** Creates levels, characters, and storylines in games.
- **Marketing:** Personalized content creation for ads and emails.

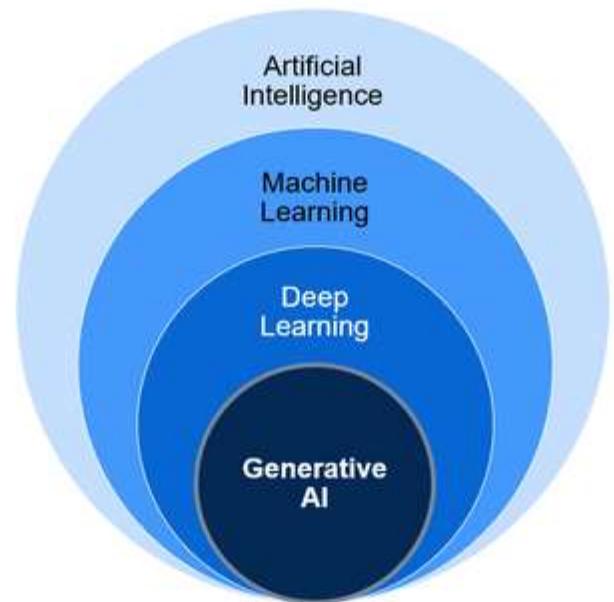


Generative AI



□ Benefits of Generative AI

- Boosts creativity by generating fresh ideas.
- Automates content creation, saving time and resources.
- Personalizes user experiences in marketing and media.
- Helps designers, artists, and writers in brainstorming.



Foundational Models, Large Language Models (LLMs), Text-to-Image, and Multimodal



Foundation Models

vs

Large language Models (LLM's)

vs

Text to Image Model (Stable Diffusion)

vs

Multi- Modal



Amazon Bedrock



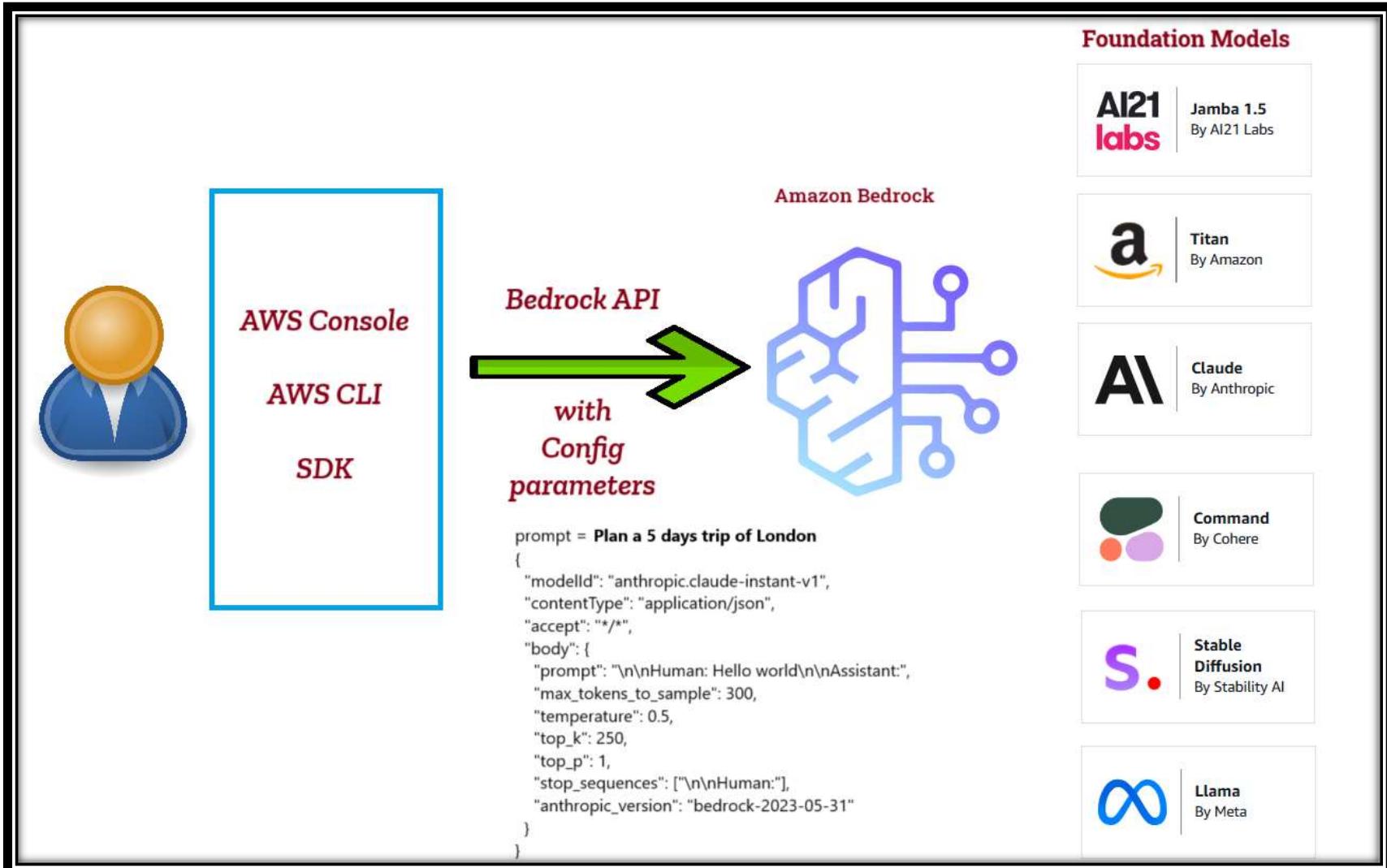
- **Amazon Bedrock** is a service for building generative AI applications on Amazon's cloud computing platform. Automates content creation, saving time and resources.
- **Amazon Bedrock** provides access to foundational models from Amazon and third-party providers through a simple API.



- **Amazon Bedrock** is a managed service by AWS that makes it easier for developers to use and build applications with advanced AI models, like chatbots, content generators, and data analyzers, without needing a deep understanding of AI technology.



How does Amazon bedrock work?





Foundation Model In Amazon Bedrock

- **Amazon Bedrock** lets you use pre-built AI models that are created by other companies.
- These models are called "foundation models" because they serve as a base for many different AI tasks.
- Foundation models are like ready-made brains that can do many AI tasks.
- Amazon Bedrock gives you access to these models, so you don't have to create or train them yourself.
- You can use them for tasks like **generating text**, **making images**, and **answering questions**.
- These models come from big companies like Anthropic, AI21 Labs, Stability AI, Meta, and Google.



Foundation Model In Amazon Bedrock



- **Jamba 1.5 by AI21 Labs** These models are called "foundation models" because they serve as a base for many different AI tasks.

□ Purpose

- Jamba 1.5 is part of AI21 Labs' language models designed for natural language understanding and generation.
- It is designed to handle a variety of language tasks, including interpreting complex sentences, generating human-like responses, summarizing information, and answering questions.

□ Key Features

- Specializes in tasks such as text summarization, content creation, and question-answering.
- Focuses on advanced capabilities for long-form content, offering improved coherence and style.



Foundation Model In Amazon Bedrock



➤ Titan by Amazon

□ Purpose:

- Amazon's Titan models are a special set of AI models created by Amazon.
- They are designed to help with tasks like understanding and generating text.

□ Key Features:

- Optimized for enterprise use, offering robust performance for customer service, document analysis, and code generation.
- Integrates seamlessly with AWS services like SageMaker and Bedrock, ensuring scalable deployment.



Foundation Model In Amazon Bedrock



➤ Claude by Anthropic

□ Purpose:

- Claude is a conversational AI model developed by Anthropic with the primary goal of providing a safe, helpful, and aligned interaction with users.
- Its design focuses on ethical AI practices by prioritizing safety and reducing harmful content, bias, and misuse.

□ Key Features:

- Focuses on natural and ethical conversational AI, with safeguards against generating harmful or biased content.
- Suitable for chatbots, personal assistants, and customer service tools.



Foundation Model In Amazon Bedrock



➤ Command by Cohere

□ Purpose:

- Command is a language model built by Cohere for complex natural language processing tasks.

□ Key Features:

- Excels in command execution, document summarization, and multilingual understanding.
- Tailored for businesses that require models for productivity and automation workflows.



Foundation Model In Amazon Bedrock



➤ Llama by Meta

□ Purpose:

- Llama (Large Language Model Meta AI) is Meta's open-source foundation model for research and enterprise applications.

□ Key Features:

- Designed for efficiency, offering competitive performance with fewer computational resources.
- Commonly used in research and development for AI-driven innovations in text generation and conversational AI.





Foundation Model In Amazon Bedrock

➤ Mistral by Mistral AI

□ Purpose:

- Mistral's models are designed for high-performance natural language understanding and generation.

□ Key Features:

- Known for smaller, more efficient architectures that provide competitive results.
- Aimed at various industries for tasks such as chatbots, summarization, and predictive text generation.





Foundation Model In Amazon Bedrock

➤ Stable Diffusion by Stability AI

□ Purpose:

- Stable Diffusion is a cutting-edge model for image generation from textual descriptions.

□ Key Features:

- Capable of producing high-quality, realistic images from prompts.
- Widely used in industries like art, marketing, and entertainment for creative content generation.



Foundation Model and Impact of Inference Parameters



- **Inference parameters influence the response generated by the model**

- **Randomness and diversity**

- Temperature
 - Top K
 - Top P

- In Summary**

- **High Randomness** = More unexpected and creative ideas.
 - **Low Randomness** = More predictable, safe ideas.
 - **High Diversity** = Wide range of different ideas or options.
 - **Low Diversity** = Narrow focus, sometimes repetitive.

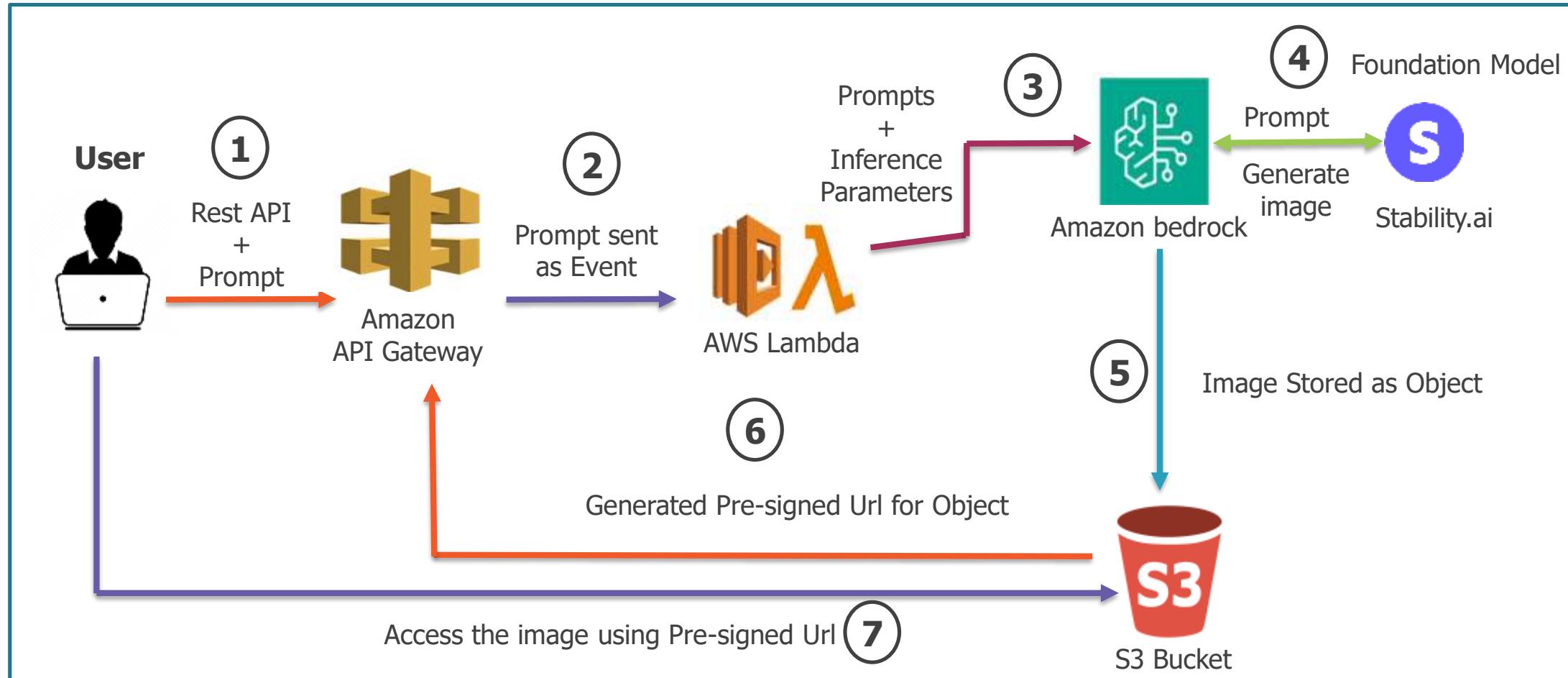
- **Length**

- Response Length
 - Stop Sequence
 - Length Penalty

Amazon Bedrock Console



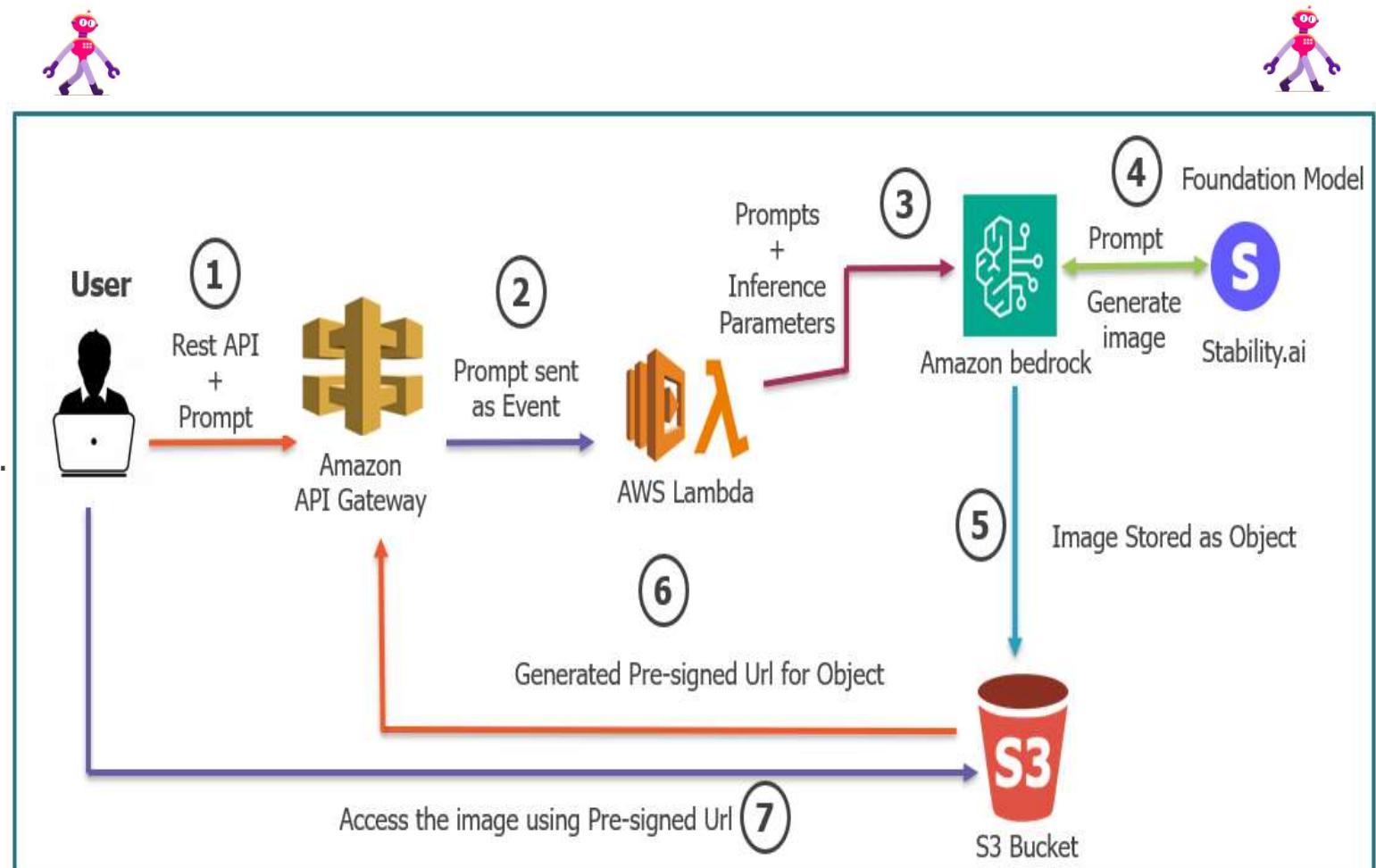
Text-to-Image Generation with Lambda Function and Foundation Model



Text-to-Image Generation with Lambda Function and Stable Diffusion Foundation Model

Here's how it works step-by-step:

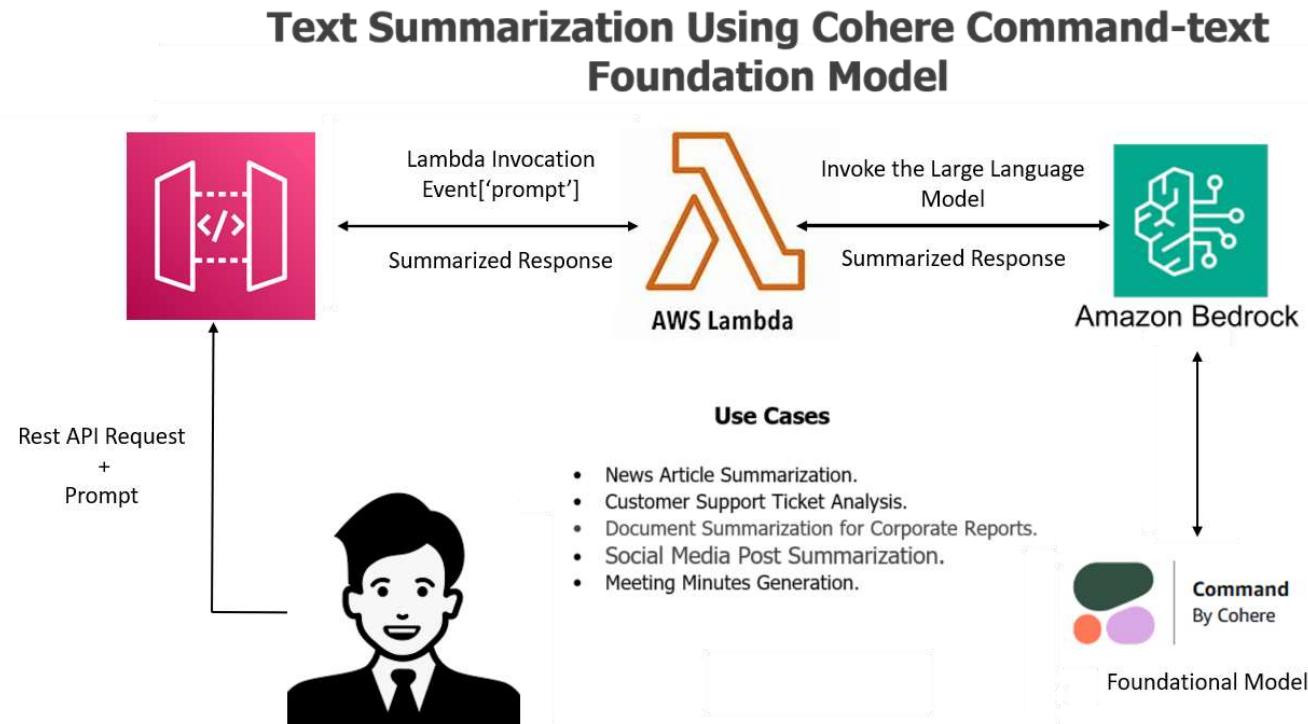
- User Sends a Text Prompt.
- API Gateway Forwards the Request to Lambda.
- The Lambda Function Processes the Request.
- The Generated Image is Saved in an S3 Bucket.
- A Pre-signed URL is Created for the Image.
- The URL is Sent Back to the User.



Use Case - Text Summarization Using Cohere Command-text Foundation Model



- Text summarization is a common natural language processing (NLP) task with various real-world use cases.
- The **Cohere Command-text-v1.4** foundation model, which is fine-tuned for instruction-following tasks, is highly suitable for summarization.

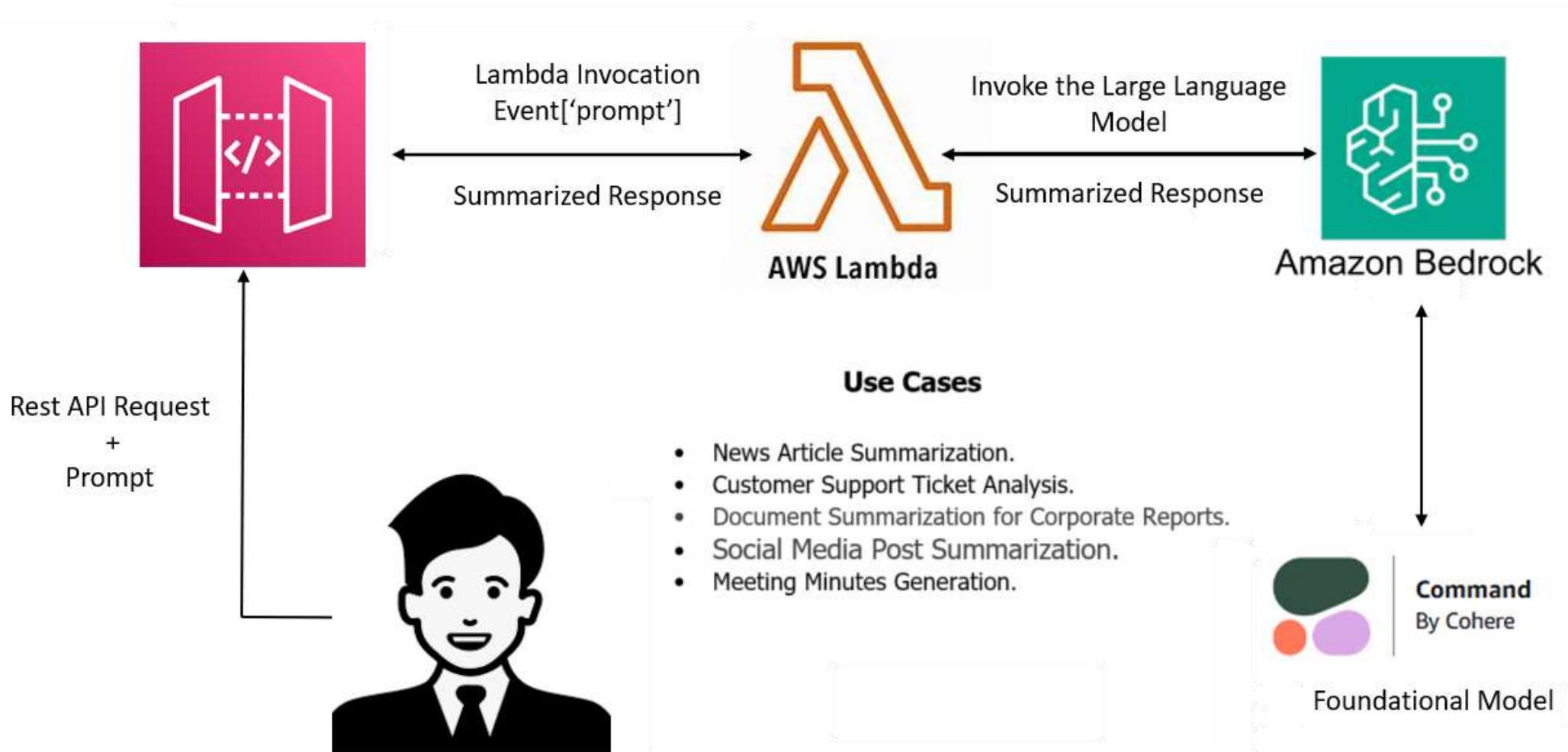


Use Case - Text Summarization Using Cohere Command-text Foundation Model

➤ **Below are a few use cases of text summarization using cohore command-text model:**

- Document Summarization for Corporate Reports.
- Meeting Minutes Generation.
- News Article Summarization.
- Academic Research Abstracts.
- Customer Support Ticket Analysis.
- Legal Brief Summarization.
- Social Media Post Summarization.
- E-commerce Product Review Summarization.

Text Summarization Using Cohere Command-text Foundation Model

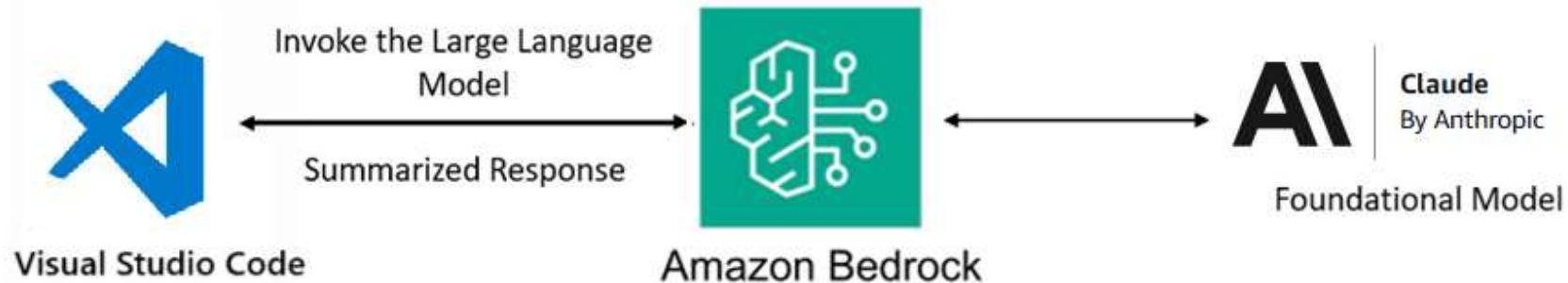


Text2Speech Player

- ❑ **Text2Speech Player** is a Python application that converts written text into speech.
- ❑ It uses Google Text-to-Speech (gTTS) to create audio from text.
- ❑ The Pygame library plays the audio instantly.
- ❑ Users can hear their text spoken aloud without saving it or using cloud storage.
- ❑ It's easy to use and perfect for:
 - Learning
 - Accessibility
 - Fun Projects
- ❑ This project shows how modern technology can make text-to-speech simple and efficient.

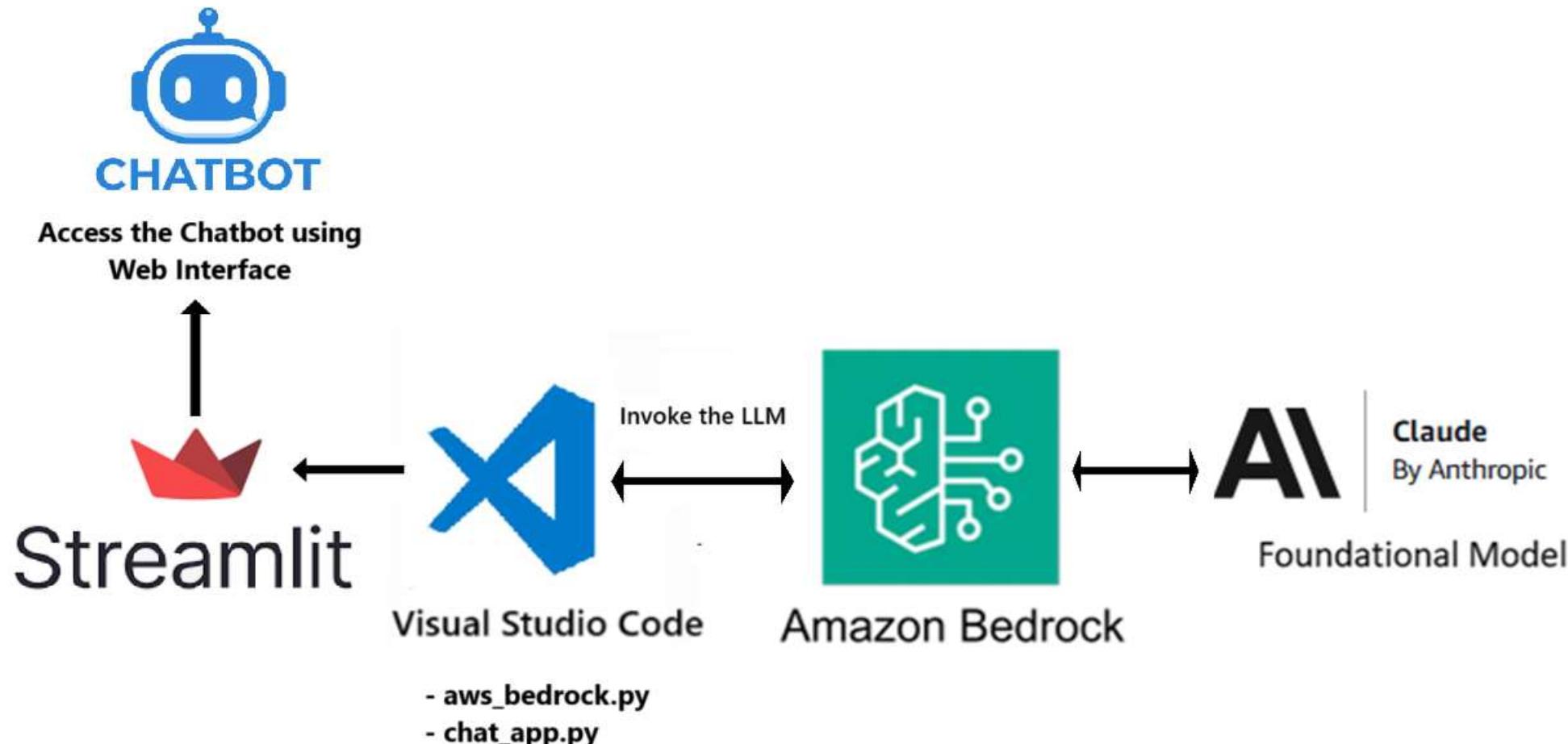


Developing a Python Chatbot Using AWS Bedrock, Anthropic Claude, and Streamlit

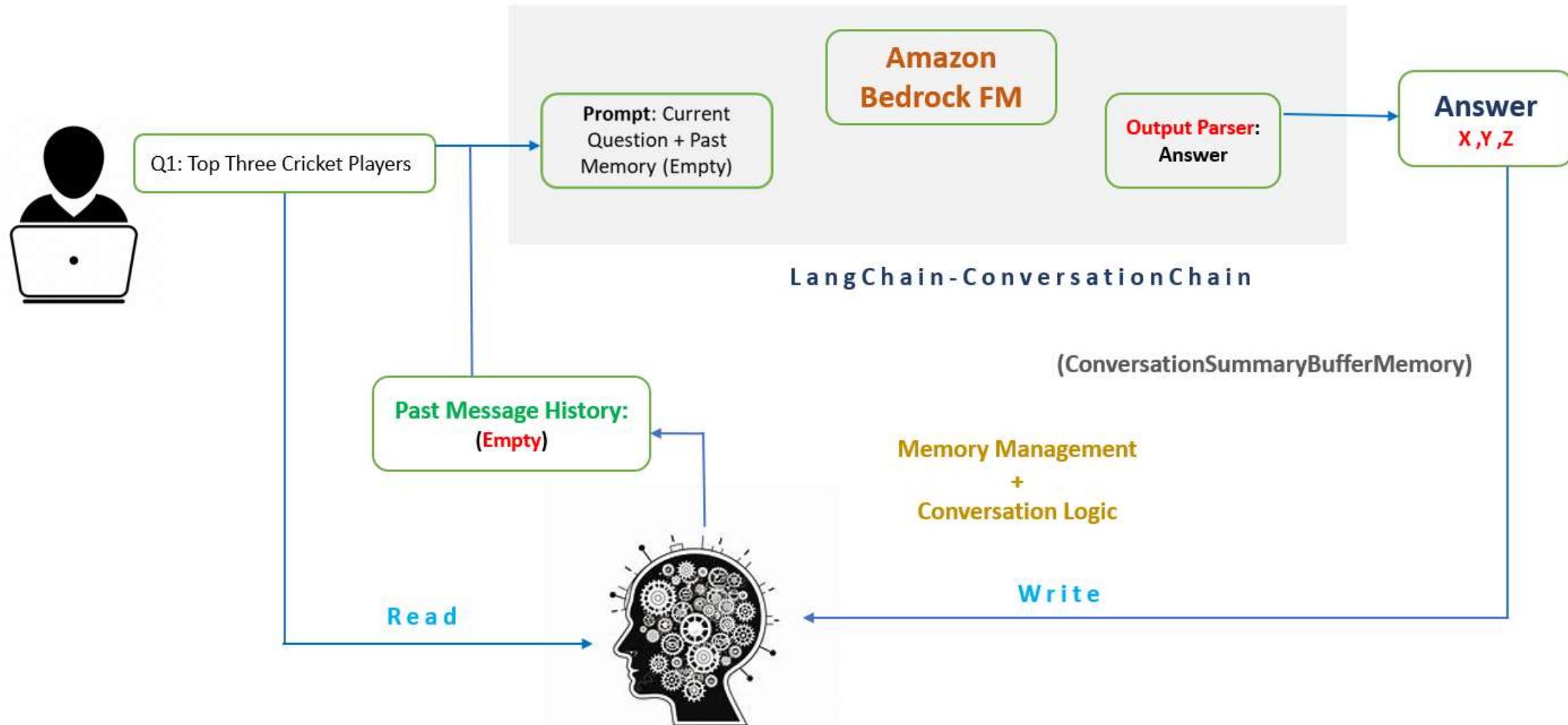


- **User Input:** Text entered by the user.
- **API Request:** Input is sent to AWS Bedrock for processing via the Claude model.
- **Response:** Claude processes the input and sends back a generated response.
- **Output:** Displayed to the user through the chatbot interface.

Developing a Python-based Chatbot Using AWS Bedrock, Anthropic Claude, and Streamlit



Developing a Python-based Chatbot Using AWS Bedrock, Anthropic Claude, and Streamlit

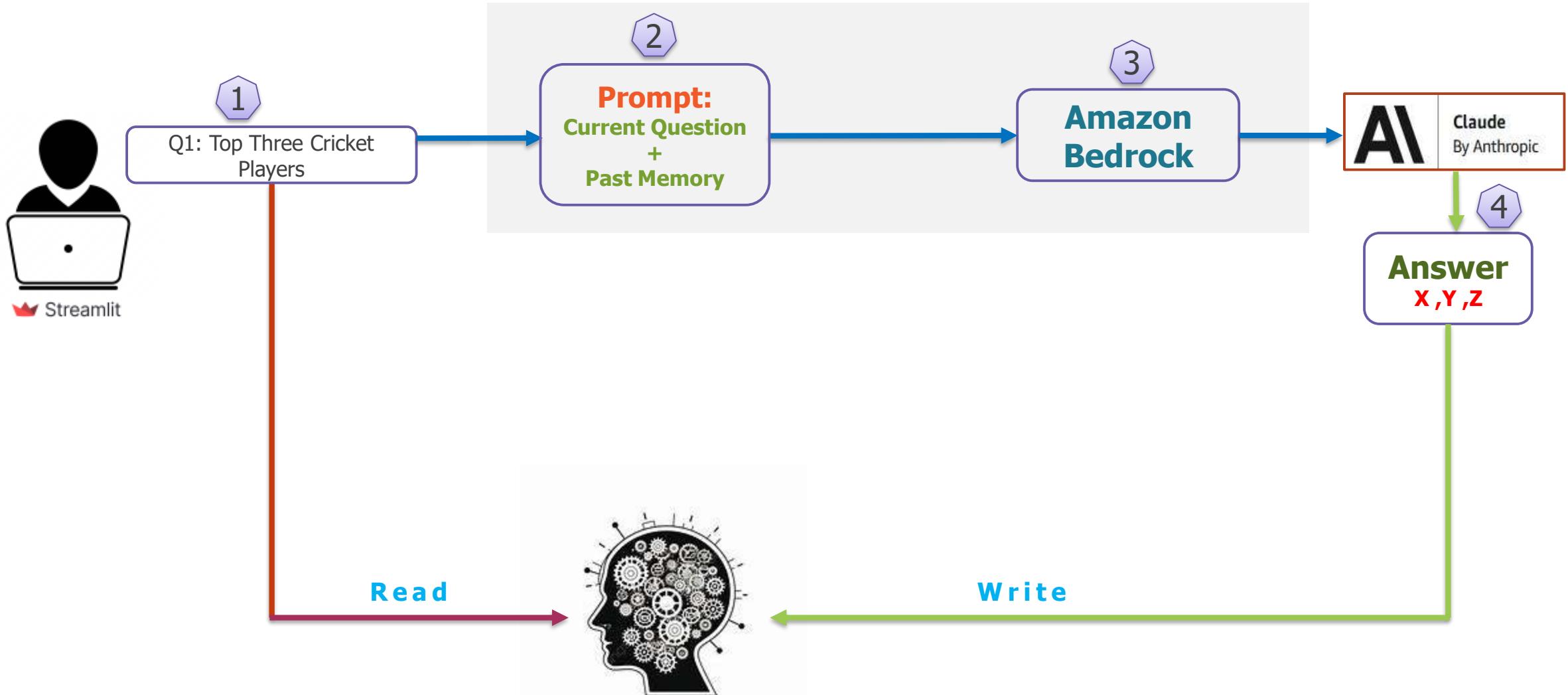


Building a Python-based Chatbot with AWS Bedrock and Anthropic Claude

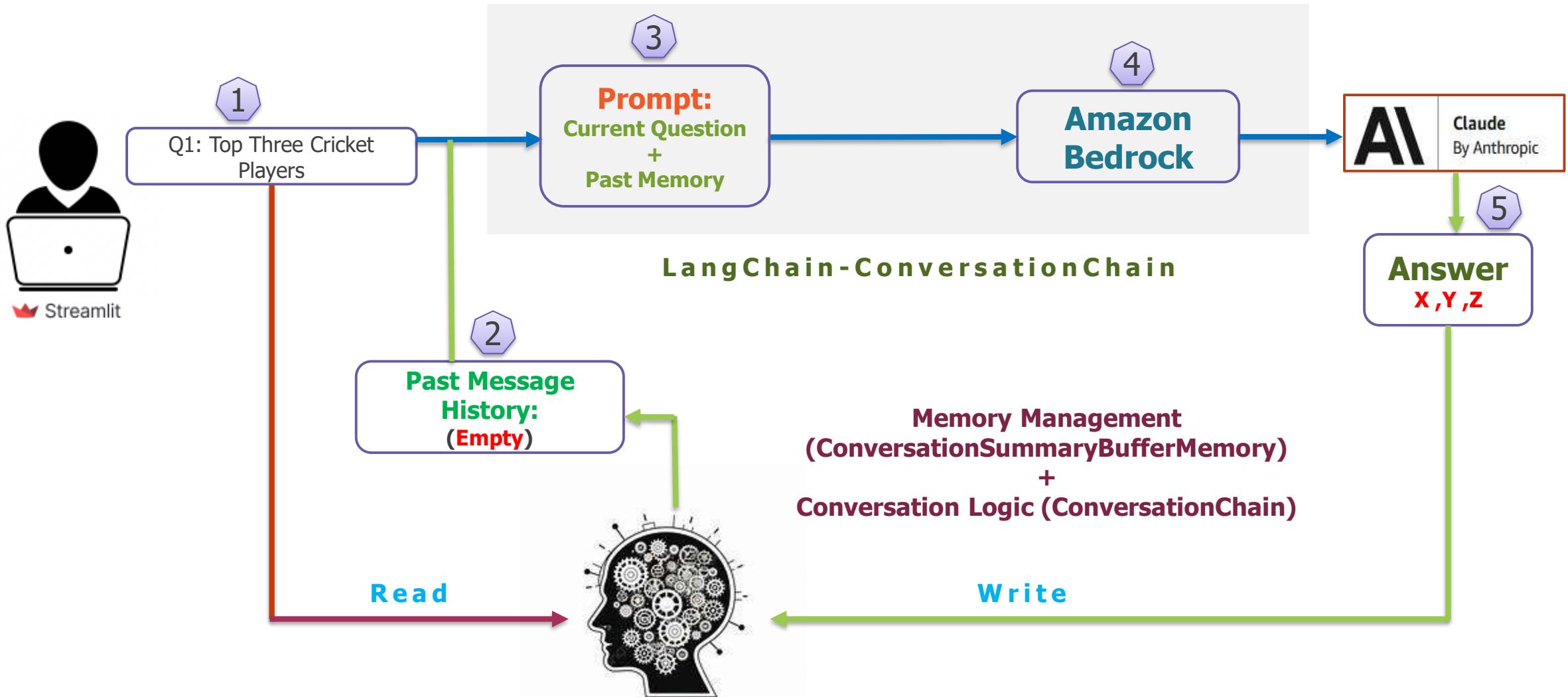
Project Flow:

- ❑ Initialize AWS Bedrock Client using boto3 (Python SDK).
- ❑ Collect user input from a text input box.
- ❑ Send the input as a prompt to the Claude model through AWS Bedrock.
- ❑ Receive and process the response from Claude.
- ❑ Display the model's response to the user.

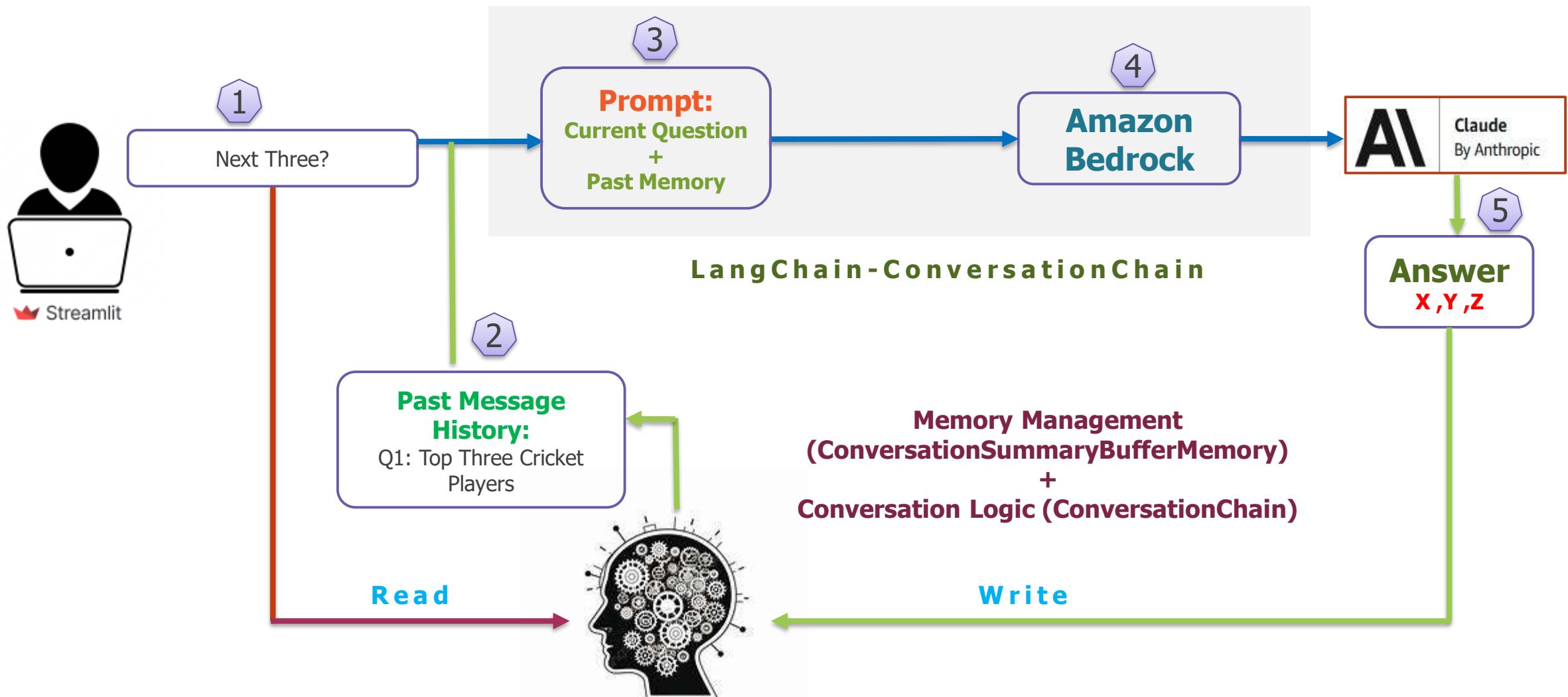
Building a Python Chatbot with AWS Bedrock, Anthropic Claude, LangChain, and Streamlit



Building a Python Chatbot with AWS Bedrock, Anthropic Claude, LangChain, and Streamlit



Building a Python Chatbot with AWS Bedrock, Anthropic Claude, LangChain, and Streamlit



Building a Python Chatbot with AWS Bedrock, Anthropic Claude, LangChain, and Streamlit

❑ `from langchain.chains import ConversationChain:`

- This line imports the **ConversationChain** class from the **LangChain library**.

- **What it does:**

ConversationChain is used to create a chatbot or conversational AI. It handles the flow of conversation and can remember past interactions if memory is attached.

- **Example:** It enables multi-turn conversations like:

- User: Who is the President of the USA?
- AI: Donald Trump.
- User: How old is he?
- AI: He is 81 years old.

Building a Python Chatbot with AWS Bedrock, Anthropic Claude, LangChain, and Streamlit

❑ **from langchain.memory import ConversationSummaryBufferMemory:**

- This imports ConversationSummaryBufferMemory, which is a type of memory provided by LangChain.
- **What it does:**
ConversationSummaryBufferMemory stores conversation history but doesn't save every detail.
Instead, it summarizes the past chat into a concise format to save memory and keep the context manageable.

❑ **Example:** Instead of storing the entire dialogue:

- Q: What's the capital of France?
- A: Paris.
- Q: What's its population?
- A: 2.1 million.

It might summarize: "*User asked about France's capital and population.*"

Building a Python Chatbot with AWS Bedrock, Anthropic Claude, LangChain, and Streamlit

❑ **from langchain_aws import ChatBedrock:**

- This imports ChatBedrock from the langchain_aws module.

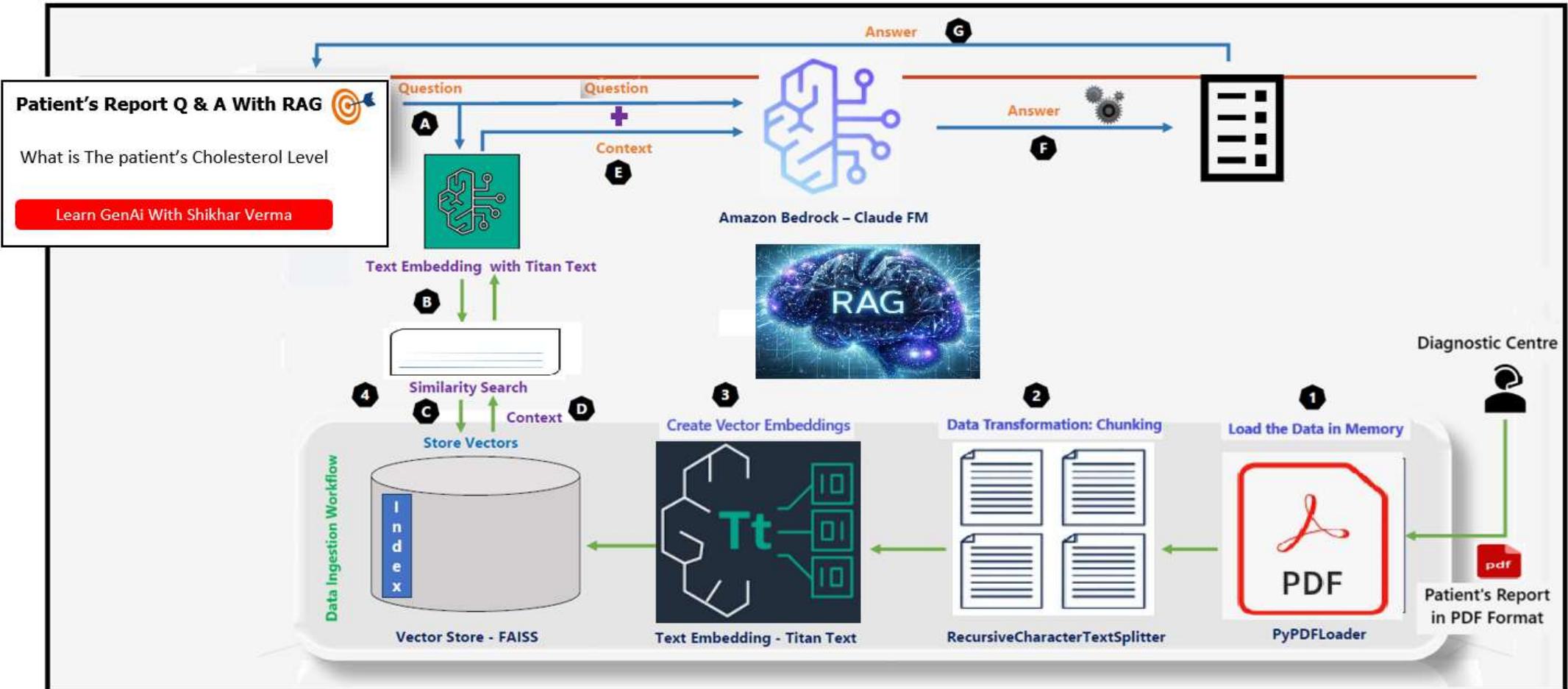
- **What it does:**

It integrates LangChain with Amazon Bedrock, allowing you to use foundation models (like Claude, Titan, or other supported models) from AWS.

- You can send questions or prompts to Bedrock's models and get AI-powered responses.

Use Case: Retrieval Augmented Generation (RAG)

- Patient Report Q & A Using RAG with Amazon Bedrock Claude Model, LangChain, and Patient's Report Document



Use Case: Retrieval Augmented Generation (RAG)

□ Prerequisites – Required Installations and Setup

- Download and install VS Code.
- Install Python.
- Install the AWS CLI. (pip install awscli)
- Configure an IAM Role for use with VS Code.
- Install the Boto3 library. (pip install boto3)
- Install LangChain. (pip install langchain, pip install langchain_aws)
- Install Streamlit. (pip install streamlit)
- Install PyPDF. (pip install pypdf)
- Install Flask-SQLAlchemy. (pip install flask-sqlalchemy)
- Install FAISS:
 - Use faiss-cpu for CPUs.
 - Use faiss-gpu for systems with CUDA-supported GPUs (or based on your Python version).