جامعة الحسين التقنية
Al Hussein Technical University

Center of Professional Development and Community Outreach

Generative AI Training Program - Professional Development (Reskilling) Training Program

Generative AI - GenAI Tools، Platforms، and Ecosystem

جامعة الحسين التقنية
Al Hussein Technical University

Module 6:

GenAI Tools, Platforms, and Ecosystem (Working with the current stack)

Dr. Abeer Al-Hyari
Abeer.alhyariups@htu.edu.jo

Module-6

Module	Module Name and Focus	Key Topics to Cover	PLOs Addressed
Module 6	**GenAI Tools, Platforms, and Ecosystem** (Working with the current stack)	1. The GenAI landscape 2. Key APIs and Platforms (OpenAI/Azure, Gemini, Anthropic, Hugging Face). 3. Building LLM Applications: Introduction to Frameworks like LangChain and Semantic Kernel. 4. The Power of Retrieval Augmented Generation (RAG) for enterprise use cases. 5. Model Deployment on Cloud Platforms (AWS SageMaker, Google Vertex AI, Azure ML).	Model Development, Prompt Engineering

Objectives

جامعة الحسين التقنية
Al Hussein Technical University

1. Identify and compare major GenAI platforms and APIs
2. Use foundational GenAI APIs in Python to perform common tasks (text generation, embeddings, multimodal input, model selection) and understand authentication, rate limits, and best practices.
3. Design and build modular LLM applications using modern orchestration frameworks such as LangChain and Semantic Kernel, applying concepts like chains, agents, and tools.
4. Explain the principles, workflow, and architecture of Retrieval Augmented Generation (RAG) and apply them to create enterprise-ready knowledge assistants using embeddings and vector databases.
5. Evaluate and select appropriate vector stores
6. Deploy GenAI applications on cloud platforms (AWS SageMaker, Google Vertex AI, Azure ML), understanding endpoints, containerization, scaling, monitoring, and security practices.

Topics Covered

جامعة الحسين التقنية
Al Hussein Technical University

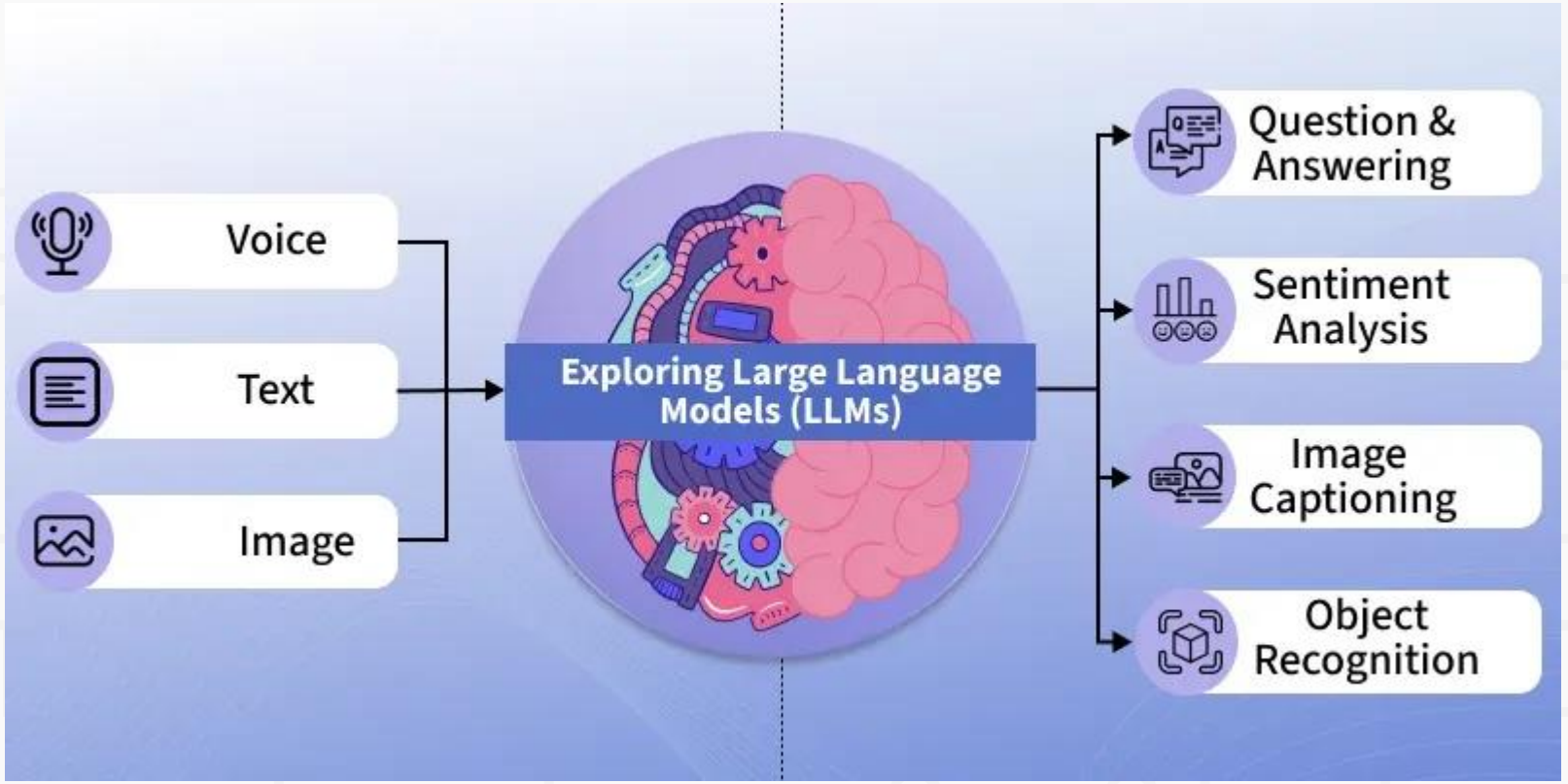
1. The GenAI landscape
2. Key APIs and Platforms (OpenAI/Azure, Gemini, Anthropic, Hugging Face).
3. Building LLM Applications: Introduction to Frameworks like LangChain and Semantic Kernel.
4. The Power of Retrieval Augmented Generation (RAG) for enterprise use cases.
5. Model Deployment on Cloud Platforms (AWS SageMaker, Google Vertex AI, Azure ML).

Large Language Models (LLMs)

جامعة الحسين التقنية
Al Hussein Technical University

- **Large Language Models (LLMs)** are advanced AI systems trained on vast amount of textual data to generate and understand human-like language. These models can perform a wide range of natural language processing tasks from text generation to sentiment analysis and summarization.

Large Language Models (LLMs)

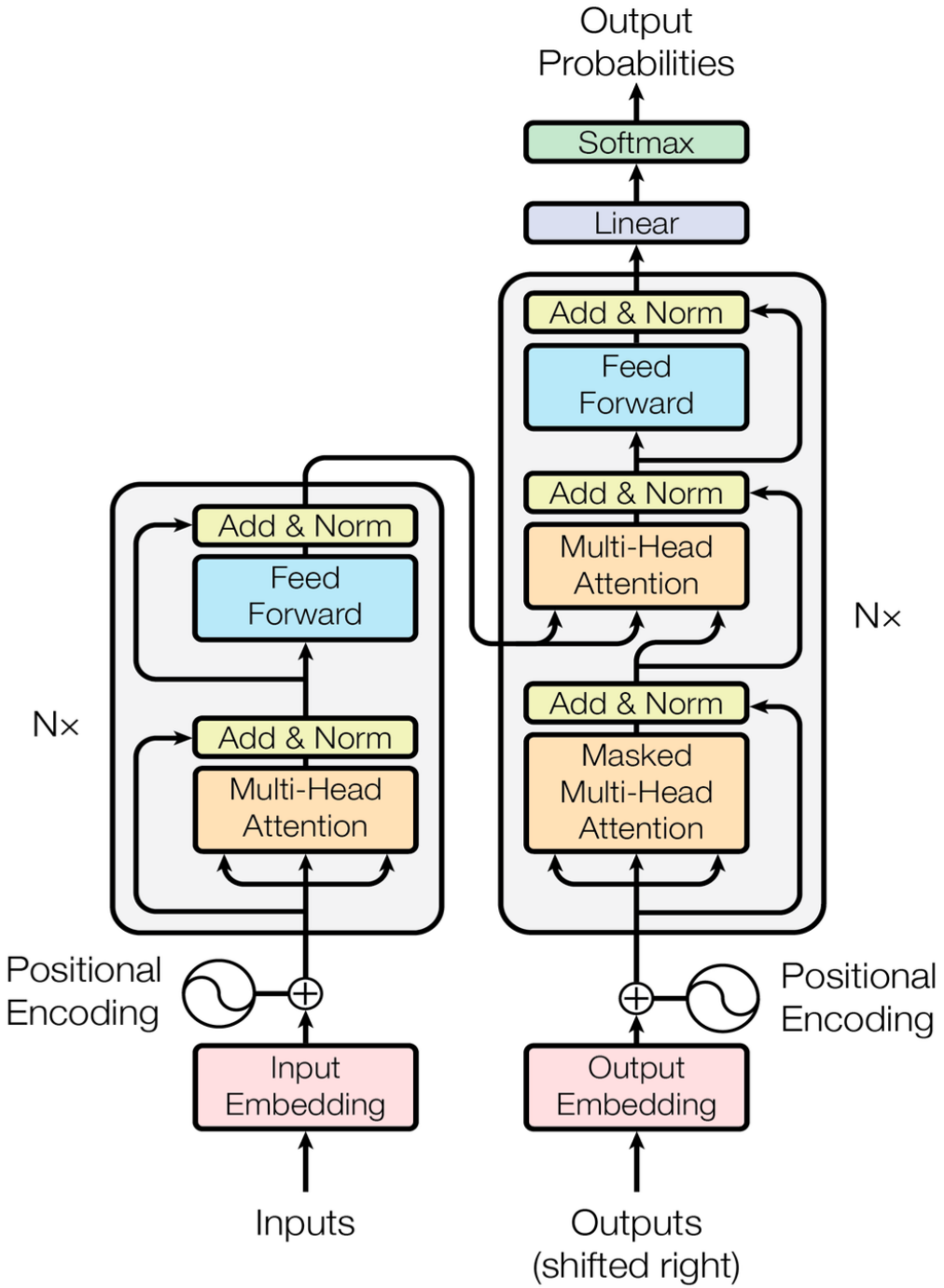


https://media.geeksforgeeks.org/wp-content/uploads/20250820175240431060/exploring_large_language_models_llms_.webp

LLMs Main Layers

جامعة الحسين التقنية
Al Hussein Technical University

1. Input Embedding Layer (tokenization + positional embeddings)
2. Multi-Head (Self-Attention) Attention Layer
3. Add & Normalization Layer
4. Feed-Forward Network (FFN / MLP Layer)
5. Repeated Transformer Blocks (N Layers Deep)
6. Output Layer (LM Head)
7. Softmax Layer


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
N×
N×
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

جامعة الحسين التقنية
Al Hussein Technical University

The GenAI Landscape

The GenAI Landscape

جامعة الحسين التقنية
Al Hussein Technical University

- Four Pillars of GenAI: Agents, Platforms, Models, Infrastructure
- A high-level framework for understanding the GenAI ecosystem

1. Agents

جامعة الحسين التقنية
Al Hussein Technical University

Definition:
GenAI **agents** are autonomous or semi-autonomous software components that use LLMs to perform tasks, make decisions, and interact with systems or users.

What they do:

- Interpret instructions (goal-driven)
- Plan multi-step actions
- Use tools (APIs, search, code execution)
- Maintain memory and context
- Execute workflows end-to-end

1. Agents

جامعة الحسين التقنية
Al Hussein Technical University

Examples:

- ChatGPT Agents, GitHub Copilot Workspace
- Customer support bots
- GenAI research assistants
- Autonomous coding/debugging agents

Agents represent *how we operationalize* models to perform real tasks.

2. Platforms

جامعة الحسين التقنية
Al Hussein Technical University

Definition:
Platforms are the **ecosystems** that provide APIs, orchestration, workflows, and developer tools for building GenAI applications.

What they provide:

- Model access (OpenAI, Gemini, Claude, etc.)
- RAG pipelines
- Evaluation & monitoring
- Guardrails & safety controls
- Multi-modal integration
- Model routing & switching

2. Platforms

جامعة الحسين التقنية
Al Hussein Technical University

Examples:

- OpenAI Platform / Azure OpenAI
- Google Vertex AI / Gemini API
- Anthropic Console / Claude API
- Hugging Face Hub
- LangChain, LlamaIndex (frameworks)

Platforms are *where GenAI is deployed, integrated, and managed*.

3. Models

جامعة الحسين التقنية
Al Hussein Technical University

Definition:
The core **foundation models** that power GenAI capabilities.

Types:

- **LLMs:** GPT-5.1, Claude 3.5, Gemini 1.5, Llama 3
- **Vision models:** GPT-Vision, Gemini Vision, CLIP
- **Multimodal models:** GPT-Omni, Gemini 2.0
- **Code models:** GPT-o1, DeepSeek Coder
- **Specialized models:** Diffusion, speech models, embeddings etc.

3. Models

جامعة الحسين التقنية
Al Hussein Technical University

- **Capabilities:**
  - Reasoning & problem-solving
  - Image/video understanding
  - Code generation
  - Text → speech & speech → text
  - Multimodal generation & understanding

Models are the *intelligence engine* behind GenAI agents and applications.

4. Infrastructure

جامعة الحسين التقنية
Al Hussein Technical University

Definition:
The hardware and system layer that makes GenAI compute possible.

- **Components:**
  - **Compute:** GPUs (A100, H100), TPUs, NPUs
  - **Cloud platforms:** AWS, Azure, GCP
  - **Inference optimization:** quantization, distillation
  - **Storage:** vector databases, data lakes, embeddings stores
  - **Networking:** high-speed interconnects (InfiniBand)
  - **Deployment pipelines:** containers, Kubernetes, ML Ops

Infrastructure is *the backbone* enabling scalable, fast, and reliable GenAI operations.

GenAI Pillars – A Summary

جامعة الحسين التقنية
Al Hussein Technical University

Pillar	Focus	Key Role	Examples
Agents	Autonomous LLM-powered components	Execute tasks & workflows	ChatGPT Agents, Copilot Workspace
Platforms	Ecosystems & APIs	Build, manage & deploy GenAI apps	OpenAI Platform, Azure, Vertex AI
Models	Foundation & fine-tuned models	Provide intelligence & capabilities	GPT-5.1, Claude 3.5, Gemini
Infrastructure	Hardware + system stack	Scaling, storage, performance	GPUs, Cloud ML, vector DBs

جامعة الحسين التقنية
Al Hussein Technical University

Key APIs and Platforms

What Is an API in GenAI?

An **API (Application Programming Interface)** in Generative AI is a bridge that allows your application (Python script, web app, Colab notebook, etc.) to communicate with an AI model hosted on a cloud platform such as OpenAI, Azure OpenAI, Google Gemini, Anthropic, Hugging Face, and others.

Think of an API as the **messenger** that:

1. Receives your request → (your prompt + settings)
2. Sends it to the GenAI model in the cloud
3. Returns the model's output → (text, image, embedding, audio, code, etc.)

Why APIs Are Important in GenAI

جامعة الحسين التقنية
Al Hussein Technical University

APIs are essential because you do not download or run big models locally—they're too large.
Instead, you access them through APIs.

APIs allow you to:

- Use GPT-4/5, Claude, Gemini, LLaMA instantly
- Build apps (chatbots, summarizers, translators, RAG pipelines)
- Integrate AI into websites, mobile apps, dashboards, and business systems
- Run secure, scalable AI without expensive hardware

An API in GenAI is the gateway that lets your applications talk to powerful AI models hosted in the cloud—fast, secure, and without needing big hardware.

Commercial GenAI APIs vs. Open-Source Models

جامعة الحسين التقنية
Al Hussein Technical University

Commercial APIs (OpenAI, Gemini, Anthropic)

- **Fully managed services**: No need to handle infrastructure, scaling, or optimization.
- **High performance & safety**: State-of-the-art models with strong guardrails and compliance.
- **Faster to integrate**: Simple APIs, enterprise-grade SLAs, monitoring, and security.
- **Less control**: Model weights are closed; limited customization beyond fine-tuning and parameters.
- **Usage-based cost**: Pay-per-token or subscription billing.

Commercial GenAI APIs vs. Open-Source Models

جامعة الحسين التقنية
Al Hussein Technical University

Open-Source Models (Hugging Face, LLaMA, Mistral, etc.)

- **Full control & flexibility**: You can self-host, fine-tune deeply, modify, and inspect the model.
- **Lower long-term cost** once infrastructure is established, especially for high-volume workloads.
- **Customizable for domain-specific tasks** (medical, legal, telecom, Arabic NLP, etc.).
- **Higher responsibility**: You manage security, GPU infrastructure, updates, scaling, monitoring, and compliance.
- **Variable quality** depending on the model, training data, and fine-tuning effort.

How to Evaluate an LLM API Provider

جامعة الحسين التقنية
Al Hussein Technical University

- **Latency and throughput**: Evaluate the time-to-first-token (the duration until the initial output is observed) and the tokens-per-second rate — leading systems now achieve sub-0.5 seconds to first token and exceed 1,000 TPS in benchmarks.
- **Pricing**: Review input/output token expenses and decide whether they use flat tiers or actual pay-as-you-go pricing to avoid unexpected costs. Some providers charge as little as $0.10 per 1M input token while others may charge up to $40 for 1M output tokens.
- **Context window**: Check the maximum tokens permitted per request, most API leads provide between 32,000 and 128,000 tokens, although advanced models now extend to 1,000,000 or more.

How to Evaluate an LLM API Provider

جامعة الحسين التقنية
Al Hussein Technical University

- **Model quality**: Evaluate benchmark scores in reasoning, coding, summarization, and fact-checking, certain models surpass standardized tests in code generation and logical reasoning, while others show superior recall and grounding capabilities.
- **Enterprise features**: Validate Service Level Agreements, data compliance certifications, and alternatives for dedicated infrastructure to meet security criteria and uptime.
- **Ecosystem & integrations**: Support for MCP, advanced SDKs, and plugin systems will help to enable integration with present tools with low code requirements.

OpenAI/Azure

جامعة الحسين التقنية
Al Hussein Technical University

- **OpenAI / Azure OpenAI Service**
  - GPT models (GPT-4.1, GPT-4o, GPT-5 family), Assistants API, embeddings, multimodal inputs
  - Enterprise integration, safety, and deployment at scale
  - Azure governance, identity, and private endpoint options

Gemini (Google)

جامعة الحسين التقنية
Al Hussein Technical University

- **Google Gemini (via Google AI Studio & Vertex AI)**
  - Gemini 1.5 family, multimodal context windows
  - Model Garden and Vertex Pipelines

Anthropic Claude

- **Anthropic Claude**
  - Claude 3.x models, long context reasoning, constitutional AI

Hugging Face Hub

جامعة الحسين التقنية
Al Hussein Technical University

- **Hugging Face Hub & Inference API**
  - Open-source models (LLaMA, Mistral, Qwen, DeepSeek)
  - Spaces, deployment endpoints, fine-tuning tools

Group Task

	Open AI	Anthropic	Gemini	Microsoft (Azure OpenAI Service)	Amazon (Bedrock)
Models/examples					
Strengths					
Pricing tiers & free credits					
Deployment options					

Cohere (Command)
Mistral
Together AI
Fireworks AI
Hugging Face (Self-Hosted)
Replicate

جامعة الحسين التقنية
Al Hussein Technical University

Building LLM Applications

Why Frameworks?

جامعة الحسين التقنية
Al Hussein Technical University

LLM applications usually require more than a single prompt–response:

- **Prompt management**
  - Reusing prompts across many features (e.g., “summarize email,” “extract tasks,” “draft reply…”).
  - Versioning and testing prompts (A/B, prompt templates, parameters).
  - Injecting dynamic data into prompts safely (variables, user inputs, retrieved documents).

Why Frameworks?

جامعة الحسين التقنية
Al Hussein Technical University

LLM applications usually require more than a single prompt–response:

- **Chaining components**
  - Combining multiple steps:
    e.g., User query → retrieve documents → summarize → generate final answer → log result.
  - Handling control flow: branching, loops, retries, fallbacks when the model fails.
  - Orchestrating different tools: LLM + embeddings + vector DB + external APIs + custom Python functions.

Why Frameworks?

جامعة الحسين التقنية
Al Hussein Technical University

LLM applications usually require more than a single prompt–response:

- **Integrating data sources**
  - Connecting to **databases, vector stores, files, APIs, enterprise systems**.
  - Implementing **RAG pipelines**: chunk → embed → store → retrieve → answer.
  - Managing credentials, configs, and environment (dev vs prod).

LangChain

جامعة الحسين التقنية
Al Hussein Technical University

What it is: A Python/TypeScript framework for building LLM-powered apps.

Key ideas:

- Prompt templates – reusable prompts with variables.
- Chains – sequences of steps (e.g., “retrieve → summarize → answer”).
- Memory – storing conversation state or user context across calls.
- Tooling integration – connectors to vector DBs (Pinecone, Chroma, etc.), SQL DBs, web search, file loaders.
- Agents (optional) – LLMs that decide which tool/chain to call next based on goals.

Semantic Kernel

جامعة الحسين التقنية
Al Hussein Technical University

What it is: A Microsoft framework (C#, Python, JS/TS) for building AI apps that combine LLMs with traditional code.

Key ideas:

- Skills / Plugins – modular collections of “functions” (prompt-based or native code) that you can call.
- Planners – the LLM plans how to achieve a goal by sequencing available functions.
- Memory & connectors – storing information and integrating external data (files, graphs, etc.).
- Deep integration with .NET and the broader Microsoft ecosystem (good fit for enterprise apps on Azure).

Agents and Chains

جامعة الحسين التقنية
Al Hussein Technical University

Chains

- A **fixed pipeline** of steps defined by the developer.
- Example chain:
  - Take user question.
  - Retrieve relevant documents from a vector store.
  - Ask the LLM to summarize the retrieved docs.
  - Format the final answer in a specific style.
- Deterministic structure: same steps every time, easier to test and debug.

Agents and Chains

- **Agents**
- An **LLM-driven decision-maker** that:
  - Receives a goal (“Book me a flight and summarize the options”).
  - Chooses **which tools/skills** to call (DB, web search, code function).
  - Decides the order and when to stop.
- More flexible but also more complex (need safety, tool restrictions, logging).

Chains = you control the flow.

Agents = the model controls the flow using tools you provide.

LangChain vs. Semantic Kernel

جامعة الحسين التقنية
Al Hussein Technical University

Feature	LangChain (Open Source)	Semantic Kernel (Microsoft SDK)
Primary Philosophy	Flexibility and rapid prototyping. A "Swiss Army knife" for LLM application development.	Structured orchestration and integration with existing business logic/code.
Key Components	Chains (sequential component calls), Agents (dynamic decision-makers), Tools (external systems/APIs).	Kernel (central orchestrator), Plugins/Skills (modular functions), Planner (automated workflow generator).
Language Support	Strongest support in Python and JavaScript/TypeScript.	Native support for C#/.NET (Microsoft's primary stack), along with Python and Java.

LangChain vs. Semantic Kernel

جامعة الحسين التقنية
Al Hussein Technical University

Feature	LangChain (Open Source)	Semantic Kernel (Microsoft SDK)
Ecosystem & Integrations	Broad and model-agnostic. Massive community-driven ecosystem with many third-party integrations (diverse LLMs, vector DBs, APIs).	Strong integration with the Microsoft Ecosystem (Azure, Azure OpenAI, MS 365, etc.).
Target Use Case	Startups, research, data science, rapid experimentation, dynamic interactive applications, Retrieval-Augmented Generation (RAG).	Enterprises, internal copilots, complex process automation, projects requiring robust governance and scalability, integrating AI with existing business systems.

جامعة الحسين التقنية
Al Hussein Technical University

The Power of Retrieval Augmented Generation (RAG)

The RAG Problem & Solution

جامعة الحسين التقنية
Al Hussein Technical University

The Problem: LLMs Only Know Their Training Data

LLMs are trained on massive datasets, but they have **two fundamental limitations**:

a) Fixed Knowledge (Knowledge Cutoff)

- They cannot know anything that happened after their last training update.
- They cannot access real-time information (unless explicitly connected to external tools).

The RAG Problem & Solution

جامعة الحسين التقنية
Al Hussein Technical University

The Problem: LLMs Only Know Their Training Data

LLMs are trained on massive datasets, but they have **two fundamental limitations**:

b) No Access to Private or Proprietary Data

LLMs **cannot** see your:

- internal documents
- PDFs, emails, reports
- database records
- company policies

- Unless this data is **explicitly provided**, the model cannot use it.

Result:
LLMs may hallucinate, give outdated answers, or miss important enterprise context.

The RAG Problem & Solution

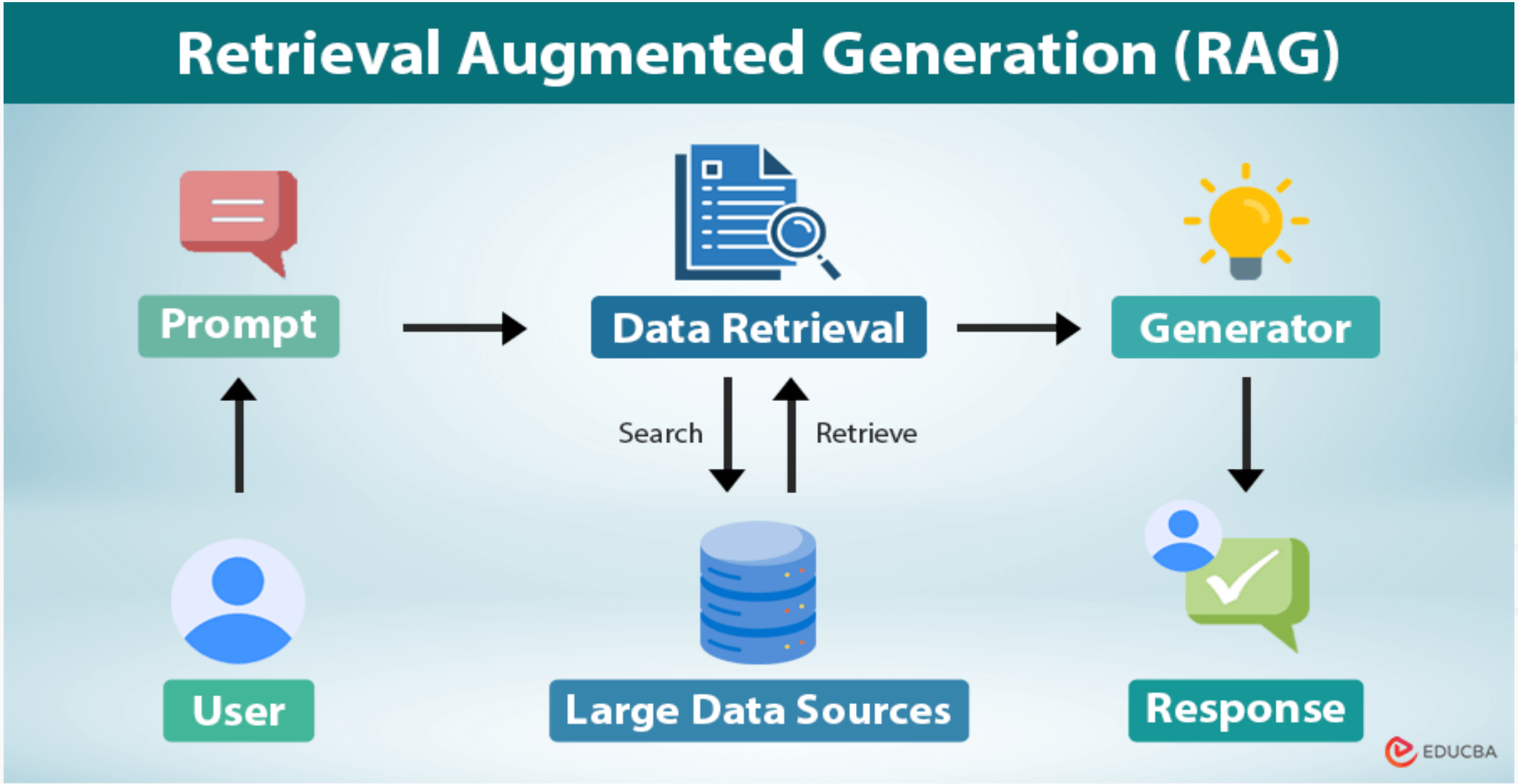
جامعة الحسين التقنية
Al Hussein Technical University

The Solution: Retrieval Augmented Generation (RAG)

What RAG Does

RAG enhances an LLM by **feeding it relevant, external, up-to-date information** retrieved from your own data sources.

Instead of trying to make the model “know more,” we give it the knowledge it needs at query time.



content/uploads/2024/12/Retrieval-Augmented-Generation-RAG.png

How RAG Works

جامعة الحسين التقنية
Al Hussein Technical University

1. Index your data

- Split documents into chunks
- Generate embeddings
- Store in a vector database (e.g., Pinecone, Chroma, Weaviate)

2. Retrieve relevant chunks

- User asks a question
- System finds the most similar document sections
- This provides context the model never had during training

How RAG Works

جامعة الحسين التقنية
Al Hussein Technical University

3. Augment the prompt

Retrieved text is added to the LLM prompt as grounding context

4. Generate answer

The LLM produces an output that is:

- grounded in real data
- Factual
- up-to-date
- aligned with your internal knowledge sources

RAG for Enterprise Use Cases

جامعة الحسين التقنية
Al Hussein Technical University

Up-to-date information

- LLMs can respond with information from “right now,” not last year’s training data.

Uses proprietary data securely

- Your private data stays inside your systems (vector DB, storage).

Reduces hallucinations

- The model answers based on **real documents**, not guesses.

Cheaper than fine-tuning

- You do not need to retrain the model every time data changes.

جامعة الحسين التقنية
Al Hussein Technical University

Model Deployment on Cloud Platforms

Main Cloud Platforms for LLMs Deployment

جامعة الحسين التقنية
Al Hussein Technical University

Here are the main cloud platforms used today for deploying Large Language Models (LLMs)—including both :

- managed services and infrastructure-level options

suitable for enterprise or academic GenAI projects

AWS – Amazon Web Services

جامعة الحسين التقنية
Al Hussein Technical University

Key Services:

- **Amazon SageMaker**
  Full MLOps platform for training, fine-tuning, and deploying LLMs.
  Supports custom models, Hugging Face, Llama, Mistral, etc.
- **Amazon Bedrock**
  Managed *foundation model* service with access to **Anthropic Claude, Amazon Titan, Meta Llama, Cohere**, etc.
- **EC2 GPU Instances**
  For custom training/deployment (NVIDIA A100, H100, etc.)

Suitable for:

- Enterprise RAG, secure model hosting, custom fine-tuning, private VPC deployments.

Google Cloud Platform - GCP

Key Services:

- **Vertex AI**
  End-to-end GenAI platform for model training, evaluation, deployment, and pipelines.
- **Gemini Models API**
  Access to Google's Gemini 1.5, 2.0, and multimodal models.
- **GKE (Google Kubernetes Engine)**
  For large-scale distributed training/inference on GPUs/TPUs.

Suitable for:

- Multimodal apps, real-time inference, scalable enterprise pipelines, TPU training.

Microsoft Azure

جامعة الحسين التقنية
Al Hussein Technical University

Key Services:

- **Azure OpenAI Service**
  Gives enterprise access to **GPT-4, GPT-4.1, GPT-4o, DALL·E, Whisper**, etc.
- **Azure ML**
  Full lifecycle MLOps environment for training and deploying your own LLMs.
- **Azure Kubernetes Service (AKS)**
  For container-based LLM deployments using GPUs.

Suitable for:

- Enterprises needing security, compliance, and high-trust deployments (e.g., banking, government).

Hugging Face

جامعة الحسين التقنية
Al Hussein Technical University

Key Services:

- **Inference Endpoints (Serverless/GPU)**
  One-click deployment for thousands of open-source models.
- **Text Generation Inference (TGI)**
  High-performance LLM inference framework.
- **Hugging Face on AWS/Azure/GCP**
  Pre-built integrations for enterprise deployment.

Suitable for:

- Fast prototyping, open-source models, custom fine-tuning workflows.

IBM Watsonx

جامعة الحسين التقنية
Al Hussein Technical University

Key Services:

- **Watsonx.ai**
  IBM’s enterprise-grade AI studio for training and deploying LLMs.
- **Watsonx.data** and **Watsonx.governance**
  Strong focus on trust, compliance, and responsible AI.

Suitable for:

- Highly regulated industries and organizations requiring enterprise governance.

Oracle Cloud (OCI)

جامعة الحسين التقنية
Al Hussein Technical University

Key Services:

- **OCI Generative AI**
  Cohere, Meta Llama, and Oracle models.
- **OCI Data Science**
  End-to-end deployment for custom models.
- **High-performance GPU clusters** at lower cost than AWS/GCP.

Suitable for:

- Organizations requiring cost-efficient GPU infrastructure at scale.

NVIDIA AI Cloud (NIM, DGX Cloud)

جامعة الحسين التقنية
Al Hussein Technical University

Key Services:

- **DGX Cloud**
  Train and deploy LLMs on NVIDIA's full GPU stack.
- **NVIDIA NIM**
  Pre-built inference microservices for LLMs.
- **AI Enterprise** suite for optimized deployment.

Suitable for:

- High-performance LLM training, multi-GPU clusters, enterprise RAG.

Cloud Platforms Benefits

جامعة الحسين التقنية
Al Hussein Technical University

Scalability and Performance

Cloud platforms provide the core infrastructure for scaling ML models both during **training** and **inference** (serving predictions).

- **Elastic Compute:** They offer on-demand, specialized hardware like **GPUs** and **TPUs (Google)**, which can be provisioned in massive, distributed clusters for large-scale model training (**horizontal scaling**). Resources automatically scale up and down (elasticity) to match the workload, ensuring models can handle sudden spikes in user traffic during real-time inference.
- **Deployment Strategies:** All platforms support advanced deployment methods like **A/B testing**, **Canary deployments**, and **Blue/Green deployments** to roll out new model versions safely without impacting end-users.

Cloud Platforms Benefits

جامعة الحسين التقنية
Al Hussein Technical University

- **Scalability and Performance**
  - **AWS:** Uses **Amazon EC2** (with auto-scaling) and specialized inference chips like **AWS Inferentia**.
  - **Azure:** Leverages **Azure Kubernetes Service (AKS)** for scalable, containerized deployments and **Azure Functions** for serverless inference.
  - **GCP:** Excels with **Cloud Run** for serverless container deployment and its custom **TPU Pods** for high-performance deep learning.

Cloud Platforms Benefits

جامعة الحسين التقنية
Al Hussein Technical University

Security and Governance

Security in ML deployment involves protecting the data, the model artifacts, and the inference endpoints. Cloud platforms offer robust, enterprise-grade security features:

- **Identity and Access Management (IAM):** All major clouds use **Role-Based Access Control (RBAC)** (e.g., AWS IAM, Azure Active Directory, Google IAM) to control who can access and modify models, data, and pipelines.
- **Data and Model Encryption:** Data is encrypted both **at rest** (e.g., in S3, Azure Blob Storage, Cloud Storage) and **in transit** (via SSL/TLS). Models and artifacts are stored securely in managed registries.

Cloud Platforms Benefits

جامعة الحسين التقنية
Al Hussein Technical University

Security and Governance

Security in ML deployment involves protecting the data, the model artifacts, and the inference endpoints. Cloud platforms offer robust, enterprise-grade security features:

- **Networking and Isolation:** They allow deploying models to **private endpoints** within a virtual private cloud (VPC/VNet), preventing public internet access and ensuring compliance with regulatory standards (e.g., HIPAA, GDPR).
- **Compliance:** Cloud providers maintain a wide array of compliance certifications, which is crucial for regulated industries like finance and healthcare.

Cloud Platforms Benefits

جامعة الحسين التقنية
Al Hussein Technical University

MLOps Tools and Automation

MLOps is the key differentiator for these integrated platforms, automating and standardizing the entire ML lifecycle.

MLOps Component	AWS SageMaker	Azure Machine Learning	Google Vertex AI
Pipeline Automation	SageMaker Pipelines	Azure ML Pipelines (integrated with Azure DevOps/GitHub Actions)	Vertex AI Pipelines (Kubeflow-based)
Model/Experiment Tracking	SageMaker Experiments and Model Registry	Azure ML Studio	Vertex AI Experiments and Model Registry

MLOps Component	AWS SageMaker	Azure Machine Learning	Google Vertex AI
Monitoring	SageMaker Model Monitor (for data/model drift)	Azure Monitor integration (for performance and drift)	Vertex AI Model Monitoring (for drift and anomalies)
Feature Management	SageMaker Feature Store	Azure ML Feature Store	Vertex AI Feature Store
Generative AI	Amazon Bedrock & SageMaker JumpStart	Azure OpenAI Service & Prompt Flow	Vertex AI Model Garden & Generative AI Studio

Deployment Considerations

جامعة الحسين التقنية
Al Hussein Technical University

Key Factors for Production-Grade LLM Deployment

Deploying an LLM in production is very different from running experiments or prototypes. Production systems must balance performance, reliability, and cost while ensuring compliance and user safety. The four most critical factors are:

- Cost
- Latency
- Security
- Monitoring

Cost (Inference Cost)

Inference cost becomes the **largest ongoing expense** in LLM applications.

Why inference is expensive:

- LLMs require **GPU/TPU** acceleration
- Larger models → more VRAM → higher compute cost
- Interactive workloads keep GPUs running 24/7
- Token generation is sequential → harder to batch

What determines inference cost:

- **Model size** (e.g., 70B vs 8B parameters)
- **Sequence length** (longer prompts = more tokens = higher cost)
- **Batching efficiency**
- **Autoscaling strategy** (idle GPUs are costly)
- **Cloud provider pricing** (A100, H100, TPU v5 are high-cost resources)

Cost (Inference Cost)

جامعة الحسين التقنية
Al Hussein Technical University

Cost-reduction techniques:

- Use **smaller distilled models** (e.g., 7B–13B) when possible
- Apply **quantization** (4-bit, 8-bit)
- Load models with **tensor or speculative decoding**
- Use serverless inference (pay per request)
- Cache frequent responses or use embedding-based retrieval

Latency

Latency directly impacts user experience, especially in chatbots, RAG applications, or real-time analytics.

Factors that affect latency:

- **Model size** → Larger models respond slower
- **GPU type** → H100 < A100 < T4 in speed
- **Token generation speed** → 10–200 tokens/sec depending on hardware
- **Network overhead**
- **Batching vs real-time trade-offs**

Latency

Latency directly impacts user experience, especially in chatbots, RAG applications, or real-time analytics.

Techniques to reduce latency:

- Use **smaller models** for interactive use-cases
- Use **speculative decoding** or **quantized models**
- Warm pools of GPU instances to avoid cold starts
- Region selection: deploy model **close to users**
- Use optimized inference frameworks:
  - NVIDIA TensorRT-LLM
  - Hugging Face TGI
  - vLLM (very fast due to efficient memory paging)

Security

Security is central in enterprise deployments, especially for sectors like finance, health, or government.

Key security requirements:

- **Data isolation**
  - Customer data must not train the model
  - No cross-tenant leakage
- **Private networking**
  - VPC, VNet, Private Service Connect
- **Encryption**
  - Data in transit (TLS 1.2+)
  - Data at rest (cloud KMS)

Security

Security is central in enterprise deployments, especially for sectors like finance, health, or government.

Key security requirements:

- **Identity and access control**
  - IAM roles, OAuth, Azure AD, GCP IAM
- **Content safety**
  - Input/output filtering
  - Guardrails and moderation services
- **Auditability and governance**
  - Logging, model lineage, policy tracking

Security

Security is central in enterprise deployments, especially for sectors like finance, health, or government.

Enterprise platforms supporting this:

- **Azure OpenAI** → strict isolation, no training on customer data
- **AWS Bedrock** → “no customer data retention” security guarantees
- **GCP Vertex AI** → VPC Service Controls + CMEK encryption
- **On-prem NVIDIA DGX** → full data control inside private datacenters

Monitoring

Monitoring ensures reliability, performance, and ethical behavior of the deployed LLM.

What to monitor in production:

Technical Metrics

- **Latency** (P50/P90/P99 response time)
- **Throughput** (tokens/sec)
- **GPU/TPU utilization**
- **Error rates** (timeouts, crashes)
- **Autoscaling triggers**

Model Quality Metrics

- Response relevance
- Hallucination rate
- Toxicity / safety violations
- Prompt injection attempts

Monitoring

Monitoring ensures reliability, performance, and ethical behavior of the deployed LLM.

What to monitor in production:

Business-Level Metrics

- Cost per request
- User satisfaction
- Conversion or task completion rate

Tools and methods:

- **MLOps platforms**: SageMaker, Vertex AI, Azure ML
- **Model-specific logging**: token usage, input/output samples
- **Observability stacks**: Prometheus, Grafana, Elastic
- **Automatic guardrails**: Azure AI Content Safety, Bedrock Guardrails
- **A/B testing and shadow deployments**

GenAI MLOps

جامعة الحسين التقنية
Al Hussein Technical University

1. Monitoring Prompt Drift

- In traditional ML, you track **input feature drift**; in GenAI, you also need to monitor how **prompts evolve over time**.
- Drift may come from:
  - Users changing writing styles or instructions
  - New business rules altering the context
  - Prompts becoming longer, ambiguous, or misaligned with expected task formats
- Tools track:
  - Prompt similarity embedding distance
  - Frequency of certain instructions
  - Changes in system vs. user instructions in production

Goal → **Detect when prompt patterns change**, as this often predicts performance degradation or hallucinations.

GenAI MLOps

جامعة الحسين التقنية
Al Hussein Technical University

2. Evaluating Output Quality

GenAI requires continuous evaluation because outputs are **open-ended**, not fixed numeric predictions.

Key evaluation methods:

- **LLM-as-a-judge** scoring (e.g., correctness, coherence, style)
- **Similarity metrics**: BLEU/ROUGE for summarization, BERTScore, embedding cosine similarity
- **Task-specific validators**: JSON schema validators, toxicity detectors, safety filters
- **Human-in-the-loop evaluation** for high-risk workflows
- **Hallucination detection** using grounding checks, retrieval accuracy, and fact-consistency modules

Goal → Ensure **factuality**, **consistency**, **safety**, and **task alignment** in outputs.

GenAI MLOps

جامعة الحسين التقنية
Al Hussein Technical University

3. Managing Data and Model Versions

GenAI requires versioning of **three assets**, not just one:

A. Prompt Templates

- Versioning system prompts, role prompts, chain-of-thought visibility rules, etc.
- Prompt versioning is critical because *small changes can change model behavior dramatically*.

B. Knowledge/RAG Data

- Need version control for:
  - Vector store snapshots
  - Document ingestion pipelines
  - “Knowledge freshness” timestamps
- Helps guarantee reproducibility and explainability of generated content.

GenAI MLOps

جامعة الحسين التقنية
Al Hussein Technical University

3. Managing Data and Model Versions

GenAI requires versioning of **three assets**, not just one:

C. Model Versions

- Track:
  - Base model version (e.g., GPT-4.1 vs 4.1-mini)
  - Fine-tuned model revision
  - Quantized or distilled deployment variants
- Use MLOps tools to maintain:
  - Model lineage
  - Rollback capability
  - Performance comparisons across versions

Goal → **Reproducible GenAI behavior** across updates, deployments, and experiments.

Module 6 Summary & Transition

جامعة الحسين التقنية
Al Hussein Technical University

Module 6 introduces learners to the modern GenAI technology ecosystem, covering leading APIs and AI platforms (OpenAI/Azure, Gemini, Anthropic, Hugging Face), practical frameworks for building applications (LangChain, Semantic Kernel), enterprise-grade Retrieval Augmented Generation (RAG), and real-world deployment workflows using cloud ML platforms (AWS SageMaker, Google Vertex AI, Azure ML). By the end of the module, students will be able to integrate, orchestrate, and deploy LLM-based applications across common tools and cloud environments.

Module 6 Summary & Transition

جامعة الحسين التقنية
Al Hussein Technical University

As we conclude Module 6 on GenAI tools and deployment ecosystems, it becomes essential to move beyond technical implementation and consider the broader responsibilities that come with building real-world AI systems. **Module 7** shifts our focus to **Responsible AI and Bias Mitigation**, where we will explore core ethical principles, methods for detecting and reducing bias, and the safety and governance frameworks needed to ensure that GenAI solutions are developed and deployed responsibly.