

ARCHITECTURAL APPROACH FOR BUILDING GENERATIVE AI APPLICATIONS



This is the second blog in my series following "[Requirements for Building an Enterprise Generative AI Strategy](#)," where I highlighted the significant challenges and expectations of enterprise customers for generative AI, with detailed requirements for building a strategy. My recommendations centered on being grounded in enterprise knowledge, integrating references for trust and verifiability, ensuring answers are based on user access control, and creating model flexibility.

In this blog, I introduce a reference architecture designed specifically for generative AI applications, demonstrate how this architecture effectively addresses generative AI enterprise challenges around trust, data privacy, security, and large language model (LLM) agility, and provide a brief overview on LLM operations (or LLMOps). As a refresher, [BMC HelixGPT](#) is our approach to generative AI integrated across the [BMC Helix for ServiceOps platform](#).

Reference architecture for generative AI applications

An application architecture describes the behavior of applications used in a business, focused on how they interact with each other and users. It is focused on the data consumed and produced by applications, rather than their internal structure. The industry has long recognized three prominent AI design patterns to build generative AI applications:

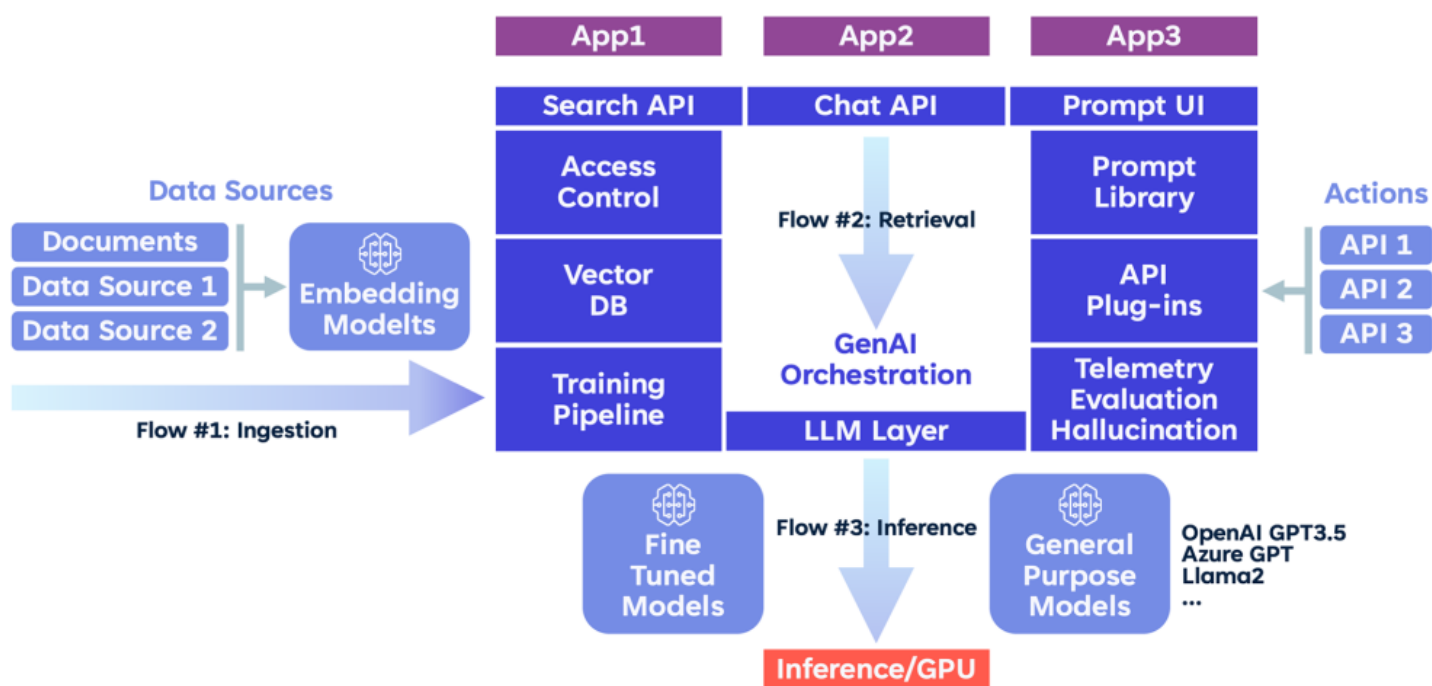
- Prompt engineering

- Retrieval augmented generation (RAG)
- Fine tuning pipelines

Instead of debating which approach is better, BMC HelixGPT seamlessly integrates all three.

The diagram below shows that our BMC HelixGPT application reference architecture for generative AI. The architecture consists of several layers: API plug-ins, prompt library, vector data source ingestion, access processing control, model-training pipeline, and assessment layer to assess hallucination/telemetry/evaluations, a "bring your own model" embedding layer, and an LLM orchestration layer. BMC HelixGPT extensively uses LangChain as the engine to orchestrate and trigger LLM chains.

Reference Architecture for GenAI



Reference Architecture for GenAI

The BMC HelixGPT proprietary generative AI technology, combined with LangChain open source models, provide "bring your own model" flexibility for our customers. There are also retrieval plug-ins, access control plug-ins, and API plug-ins that integrate into enterprise systems. Like the holistic design explained in this [June 2023 blog by Andressen Horowitz](#), we have three main flows:

- Data ingestion and training flow:** Data is read from multiple data stores, preprocessed, chunked, and trained through an embedding model (RAG) and training pipeline (fine-tuning). VectorDB stores the chunked document embeddings that allow for better semantic, similarity-based data retrievals.
- Prompt augmentation using data retrieval:** Once a user query arrives at the API layer, the prompt is selected, followed by data retrievals through VectorDB or API plug-ins to get the right contextual data before the prompt is passed to the LLM layer.
- LLM inference:** This is where there is a choice to use general purpose foundation models from OpenAI, Azure GPT models, or the self-hosted foundation model in BMC HelixGPT. Fine-tuned models are used when tuned for a specific task or use case. The response is evaluated for

accuracy and other metrics, including hallucinations.

Now, let us look at how this reference architecture addresses the challenges of generative AI for enterprises and facilitates the rapid development of generative AI applications.

Overcoming common enterprise challenges with generative AI deployments

Enterprise versus world knowledge: accuracy and trust

Enterprises seek answers across diverse internal and external enterprise data sources such as articles, web pages, how-to guides, and more. Further, data can be contained in both unstructured and structured databases. BMC HelixGPT ingests, chunks, and embeds these sources through LangChain data loaders using embedding transformer-based models. LangChain provides a rich set of document loaders that it leverages. When a user question is received, we augment the prompts with document retrievals from VectorDB or APIs and use the LLM's in-context learning to generate a response. This method anchors the LLM's response to the retrieved documents or data, reducing the risk of hallucinations. BMC HelixGPT also provides the retrieved documents as citations, allowing users to verify the responses. To realize this advanced capability, our strategy integrates various LangChain capabilities, such as retrieval QA chains with sources and conversation history chains.

Access control, security, and data privacy

During the retrieval of document flow, BMC HelixGPT validates that the user has access permissions to read the documents and removes those documents from the prompt context that the user doesn't have access to. This ensures that LLM-generated answers are always from only those documents that a user has read access to. Hence, the same question will generate two different answers aligned to the user's role and permission model.

Model flexibility and security

The BMC HelixGPT reference architecture is based on a model abstraction layer that LangChain provides. This capability enables seamless integration of foundational general-purpose models, whether hosted or behind APIs such as OpenAI and Azure or open-source models running in customers' centers. There are over 50 connectors to different model providers in LangChain, making it easy to add new providers or models modularly. Customers who prioritize data security have the option to host and run a foundational model in the datacenter. This model architecture caters to diverse enterprise customers and prevents vendor lock-in, including implementations that provide the strongest privacy and security guarantees.

An Introduction to LLMOps

Machine learning for IT operations (MLOps) for LLM is called LLMOps. LLMOps is a new set of tools and best practices to manage the lifecycle of LLM-powered applications, including data management, model management, and LLM training, monitoring and governance LLMOps is the driving force to build generative AI applications for BMC HelixGPT.

BMC HelixGPT is a platform that provides models and services that allow applications to harness the power of generative AI. It also provides LLMOps foundational services such as prompt engineering and RAG to power a spectrum of use cases ranging from summarization to content generation and

conversations.

LLMOps is distinct from MLOps because it introduces three new paradigms for training LLMs:

- Prompt engineering
- Retrieval augmented generation
- Fine-tuning pipelines

My third and final installment in this blog series will dive deeper into BMC HelixGPT's LLMOps capabilities.