Gen AI with Element

Surendra Panpaliya

PREREQUISITES

Participants should have:

Basic knowledge of **Python programming**

Familiarity with **APIs**, **JSON**, and **HTTP requests**

General understanding of **Machine Learning** and **NLP concepts**

PREREQUISITES

- Awareness of **cloud platforms** (Azure or GCP preferred)
- Prior exposure to **Jupyter Notebooks** or **VS Code**
- Knowledge of **RESTful APIs**, **Docker**, and **Git**
- Some experience with **LLMs** or **prompt engineering**

LAB SETUP REQUIREMENTS

- **Python 3.10+** installed (preferably in a virtual environment)
- **JupyterLab** or **VS Code** with Python plugin
- Access to:
- **Azure OpenAI API key** (or)
- **Google GenAI credentials** (Vertex AI Studio, PaLM/Gemini API)

📌 LEARNING OUTCOMES

Explain key concepts in

Generative AI and

Transformer-based LLMs

Build and interact with

OpenAI/Gemini models using Python

📌 LEARNING OUTCOMES

Design **RAG pipelines** integrated with

LangChain + Milvus

Apply structured prompt

engineering strategies in LLM apps

📌 LEARNING OUTCOMES

Utilize Walmart's internal

LLM Gateway and **evaluation platforms**

Create simple **Agentic applications**

with planning, execution, and tools

📌 LEARNING OUTCOMES

Troubleshoot common LLM issues

such as hallucination or bias

Build and present a functional

Excel-based report builder GenAI app

Agenda

- DAY 1: GENAI FOUNDATION & ARCHITECTURE
- DAY 2: WALMART GENAI ECOSYSTEM
- DAY 3: APPLICATION DEVELOPMENT WITH GENAI
- DAY 4: HACKATHON & DEPLOYMENT

DAY 2: WALMART GENAI ECOSYSTEM

Objective:

Develop LLM-powered document generation and

retrieval applications using

Walmart’s internal GenAI infrastructure.

Agenda

- Prompt Engineering
- Prompting Strategies
- Structured Prompt Design
- RAG Architecture Overview
- LangChain Integration

Agenda

Walmart LLM Gateway

Embedding & Vector Store

LLM Use Cases @ Walmart

Hands-On Session

What is Prompt Engineering?

Art and science of crafting effective inputs (prompts)

that guide AI models

especially Large Language Models (LLMs)

to produce desired outputs.

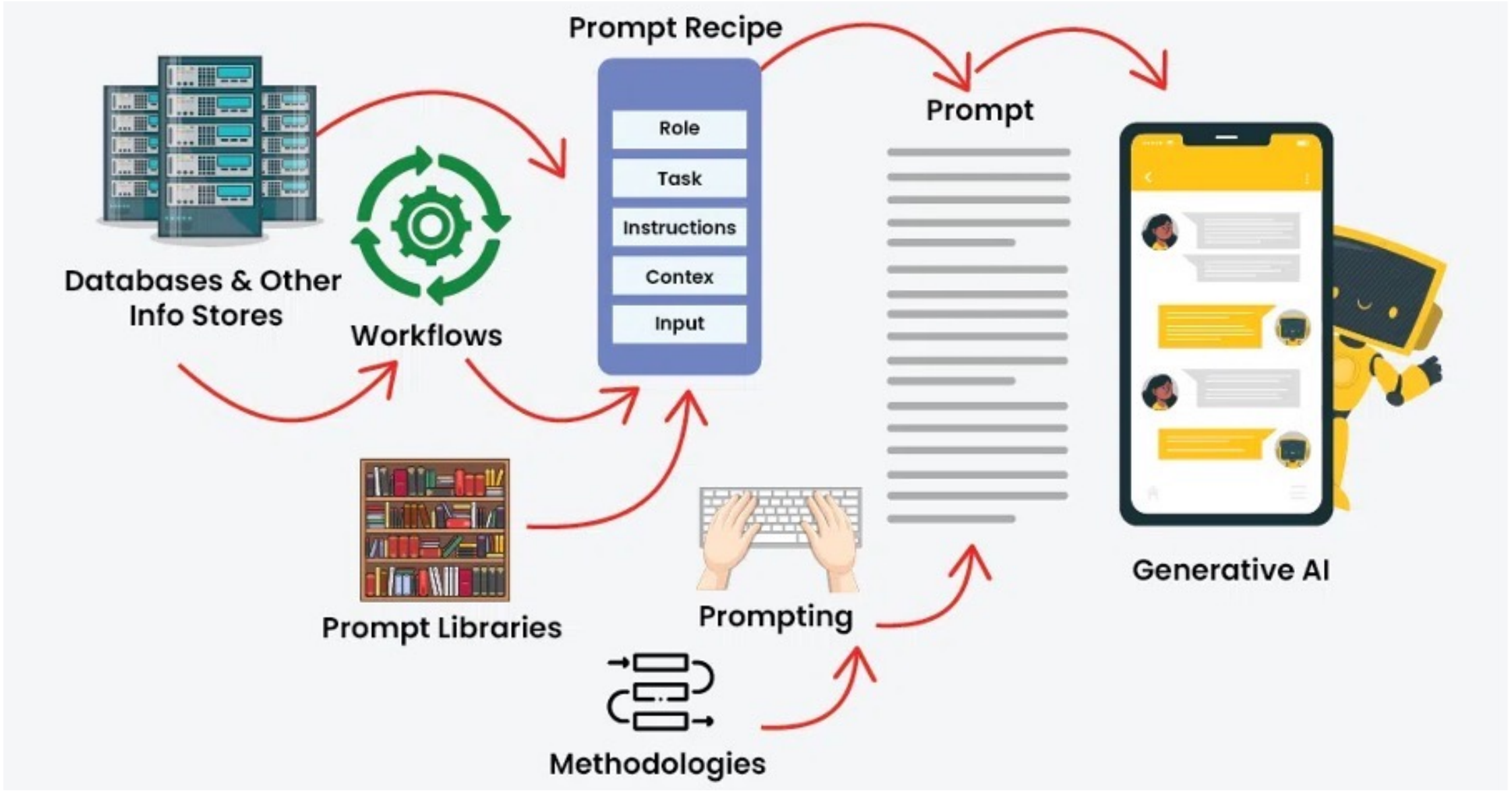
What is Prompt Engineering?

Prompts act as a bridge between

Human instructions and machine outputs,

allowing models to deliver more accurate,

contextual, and human-like responses.


Prompt Recipe
Role
Task
Instructions
Contex
Input
Databases & Other Info Stores
Workflows
Prompt Libraries
Prompting
Methodologies
Prompt
Generative AI

Why AI Prompt Engineering Matters?

Enhances Model Accuracy:

Well-crafted prompts improve AI output quality.

Tailors Outputs:

Custom prompts align responses to user-specific needs.

Why AI Prompt Engineering Matters?

Reduces Bias:

Ethical prompt design minimizes harmful or biased outputs.

Improves UX:

Better prompts = smoother and more relevant interactions.

Techniques Used

Contextual Prompts:

Add background for relevance.

Specificity:

Reduce vagueness for precise output.

Techniques Used

Iterative Refinement:

Test > Analyze > Modify > Repeat.

Techniques Used

Templates:

Reusable frameworks for consistency.

Experimentation:

Explore open vs. closed-ended prompts.

Prompt Engineering for GenAI

Goal: Understand Prompt Engineering

How Walmart developers can use it

To build high-impact, reliable, and

Enterprise-ready GenAI systems.

What is Prompt Engineering?

The practice of designing clear, structured,

Purposeful instructions (prompts)

To guide Large Language Models (LLMs)

like GPT, Gemini, or Claude

To generate accurate, context-aware, and useful outputs.

Why it matters for Walmart?

Automates customer support & content

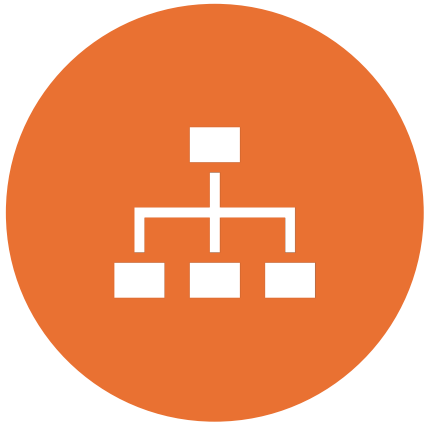
Reduces hallucination in responses

Accelerates GenAI adoption in retail

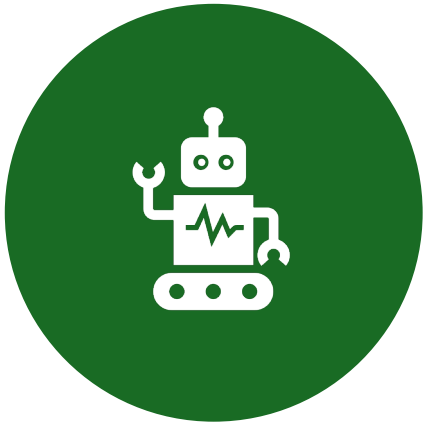
How Prompt Engineering Helps Developers?

- Precise control over LLM behavior
- Faster prototyping of GenAI workflows
- More aligned, relevant, and reliable responses

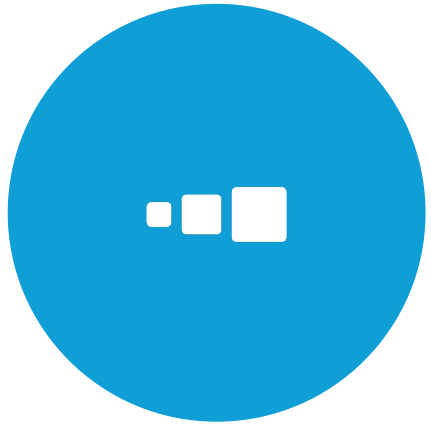
Walmart Use Case



GENERATE STRUCTURED PRODUCT DESCRIPTIONS,



AUTOMATE RETURN POLICY ANSWERS, AND



REDUCE AGENT LOAD.

Prompting Strategies

Zero-shot prompting

Few-shot prompting

Role-based prompting

Zero-Shot Prompting

Asking the model to perform a task

without any examples

leveraging its general knowledge base

Example

Calculate the estimated delivery date

for an order placed today in Pune.

Walmart Use Case

Customer chatbot offers real-time answers like:

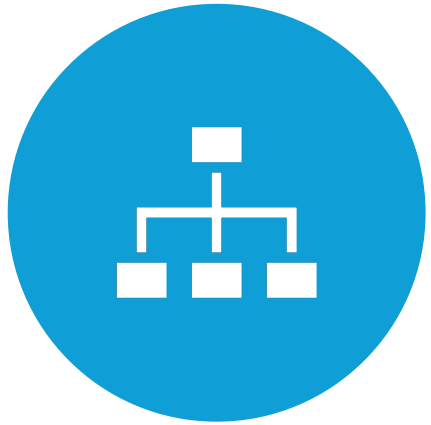
Your order placed today in Pune

should arrive by Monday.

Why It Works?

QUICK TO DEPLOY,

USES CLEAR INSTRUCTIONS,



IDEAL FOR STRAIGHTFORWARD TASKS.

Few-Shot Prompting

Provide a few example

input-output pairs

to teach the pattern in context.

Example 1

Q: Product: ₹1999 headphones,

Features: noise-cancelling

A: Clear audio, Blocks noise for focus

Example 1

Now format for:

Q: Product: ₹999 kids backpack,

Features: cartoon print, adjustable straps

A:

Walmart Use Case

Automatically generate consistent

brand-aligned bullet points.

Why It Works?

Reduces hallucinations and

improves output consistency.

Role-Based Prompting

Assign	Assign a persona or role
Tone/style	to shape tone/style and
Ensure	ensure domain-specific alignment

Example

You are a Walmart logistics strategist.

Identify three ways to reduce delivery delays

in Pune during monsoon season.

Walmart Use Case

Produces professional,

strategy-focused insights

for internal teams.

Why It Works

Controls tone, style, and

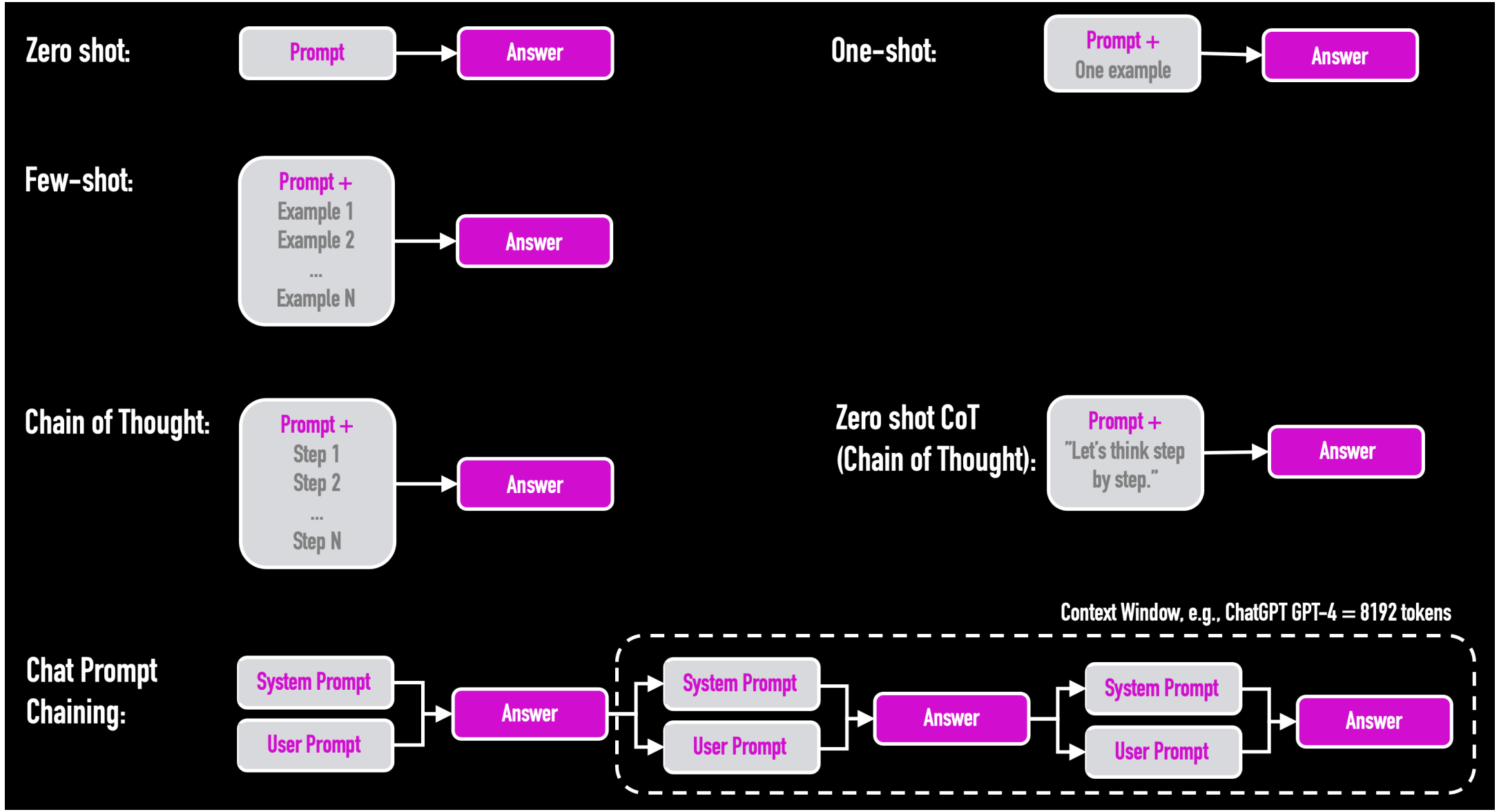
expertise level

critical for business and

internal applications.

Summary

Strategy	Key Trick	Developer Example	Walmart Use Case
Zero-Shot	Direct instruction	Check ₹999 kids backpack stock in Mumbai.	Quick stock check in chatbots
Few-Shot	Teach via examples	Provide product examples -> generate new bullets	Uniform product content generation
Role-Based	Define persona/tone	You are a product specialist. Explain return policy.	Polished responses in policy docs or training bots


Zero shot:
Prompt
Answer
One-shot:
Prompt +
One example
Answer
Few-shot:
Prompt +
Example 1
Example 2
...
Example N
Answer
Chain of Thought:
Prompt +
Step 1
Step 2
...
Step N
Answer
Zero shot CoT
(Chain of Thought):
Prompt +
"Let's think step
by step."
Answer
Context Window, e.g., ChatGPT GPT-4 = 8192 tokens
Chat Prompt
Chaining:
System Prompt
User Prompt
Answer
System Prompt
User Prompt
Answer
System Prompt
User Prompt
Answer

Structured Prompt Design

Crafting prompts for document generation

Principles of clarity,

structure,

context preservation

What is Structured Prompt Design?

Crafting prompts that are

clear, contextual, and goal-oriented,

helping LLMs like GPT or Gemini

generate precise, useful outputs.

Components of a Well-Structured Prompt

Role/Persona

Define who the AI is:

“You are a Walmart product specialist.”

Components of a Well-Structured Prompt

Task

Clearly state the job:

“Generate a product summary.”

Components of a Well-Structured Prompt

Context

Provide necessary info:

product specs,

SKU (Stock Keeping Unit) details,

target audience.

Components of a Well-Structured Prompt

Instructions/Constraints

Specify format,

tone, length, style, or

forbidden elements.

Components of a Well-Structured Prompt

Examples (Few-Shot)

Show the model

what good output looks like.

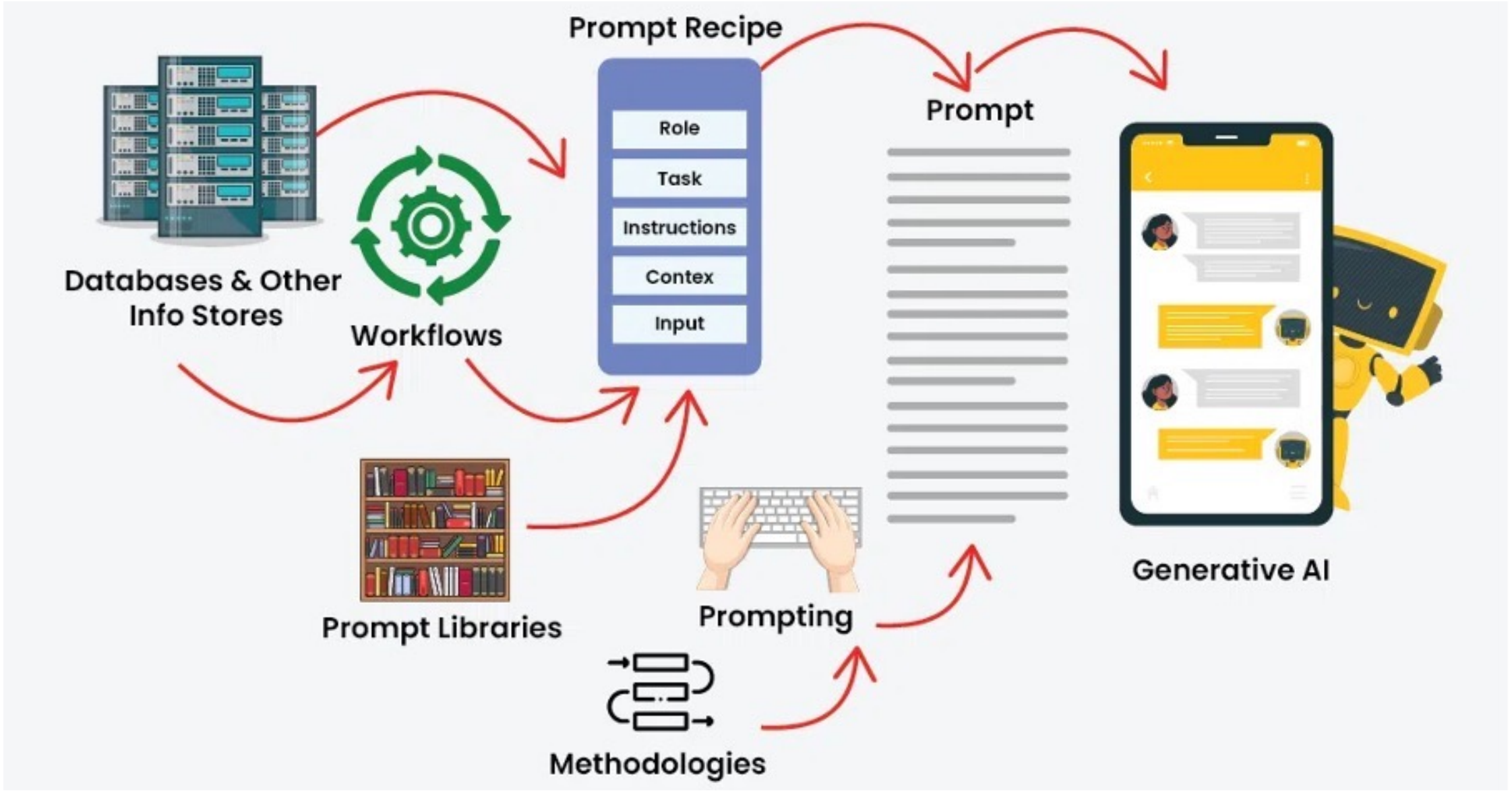
Components of a Well-Structured Prompt

Output Format

Define structure:

bullet points,

HTML snippet, table


Prompt Recipe
Role
Task
Instructions
Contex
Input
Databases & Other Info Stores
Workflows
Prompt Libraries
Prompting
Methodologies
Prompt
Generative AI

Walmart Prompt

Generating a Product Description

Generating a Product Description

You are a Walmart product specialist.

Task:

Create a short, persuasive product description.

Context:

Product: Kids’ backpack

Price: ₹999

Features: durable nylon,

two zip pockets,

adjustable straps, cartoon print

Constraints:

- Max 2 sentences
- Tone: friendly, helpful to parents

Please provide:

1 catchy headline

2 bullet points focused on quality and kid appeal

Example:

Headline: Durable Backpack for School Adventures!

- Made with tough nylon...
- Comfortable straps...

Begin:

Headline:

-
-

Types of Prompts with Walmart-Specific Examples

Surendra Panpaliya

GKTCS Innovations

1. ✍️ Instruction Prompt

Give the LLM a direct, clear instruction.

Example Prompt:

“Summarize the return policy for electronic items in 2 bullet points.”

Walmart Use Case:

Used in chatbots to explain policies clearly to customers.

2. 🎭 Persona Prompt

Assign a specific role or identity to the LLM for tone and context.

Example Prompt:

“You are a Walmart customer service executive.

Write a friendly email to a customer whose refund

is delayed by 3 days.”

2. 🎭 Persona Prompt

Walmart Use Case:

Customer communication automation

that aligns with brand voice.

3. Few-Shot Prompt

Provide **examples** to teach

the model what kind of output you expect.

Example Prompt

- Input: Product - Wireless Earbuds,
- Price - ₹1,499,
- Feature - Long battery life
- Output: Crystal Clear Audio for Every Beat!
- 20-hour battery life
- Sleek & snug fit

Input: Product - Kids’ backpack, Price - ₹999, Feature - Cartoon print Output:

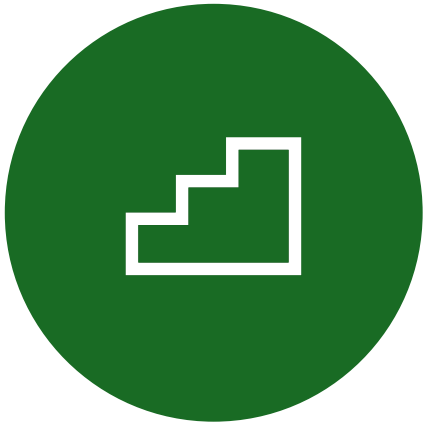
Walmart Use Case

Auto-generate product descriptions or

marketing copy with brand tone.

4. Chain-of-Thought Prompt

ENCOURAGE THE MODEL



TO **SHOW ITS REASONING STEP-BY-STEP**

BEFORE GIVING AN ANSWER.

Example Prompt

“A blender was purchased 40 days ago.
Walmart’s return policy allows 45-day returns.
Should the customer be allowed to return it?
Explain.”

Walmart Use Case

Used for building GenAI agents

in compliance reasoning or

policy validation.

5 best practices for prompt engineering

Be clear and specific

To generate accurate responses avoid using vague language.

Provide context

Use relevant examples to help the model generate appropriate responses.

Experiment with different prompts

Use a variety of prompts to test the limits of the model.

Use relevant keywords

To help the model focus on the specific task and improve the relevance of the responses.

Refine the prompt

Use feedback to improve the prompt and optimize the model's performance.

RAG Architecture Overview

Introduction to Retrieval-Augmented Generation

Why RAG is critical for enterprise GenAI adoption

What is RAG?

RAG combines traditional **retrieval systems**

(like document or vector search) with **LLMs**

to produce answers grounded in real,

up-to-date data

What is RAG?

Retrieves relevant content

from enterprise data

policies, product manuals,

inventory logs

What is RAG?

Augments

the user query

by combining it

with this retrieved context.

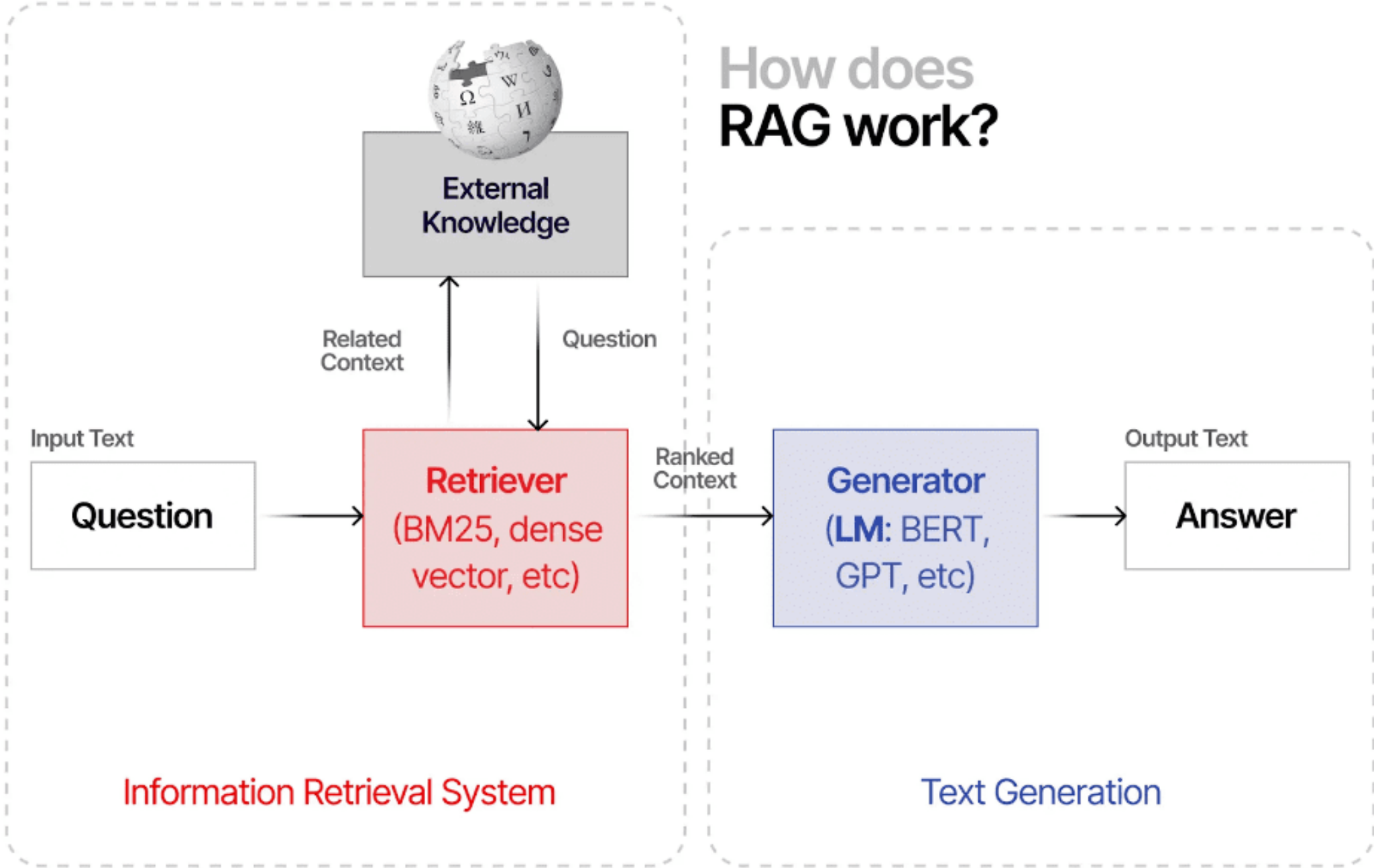
What is RAG?

Generates

a precise,

accurate response

using the LLM.


How does
RAG work?
External Knowledge
Related Context
Question
Input Text
Question
Retriever (BM25, dense vector, etc)
Ranked Context
Generator (LM: BERT, GPT, etc)
Output Text
Answer
Information Retrieval System
Text Generation

Why RAG is Critical for Enterprise GenAI ?

Reduces hallucinations,

ensuring accuracy and trust

Why RAG is Critical for Enterprise GenAI ?

Keeps responses current

by accessing live

enterprise data

no retraining needed

Why RAG is Critical for Enterprise GenAI ?

ENABLES DOMAIN-SPECIFIC ANSWERS,

IMPORTANT FOR POLICY,

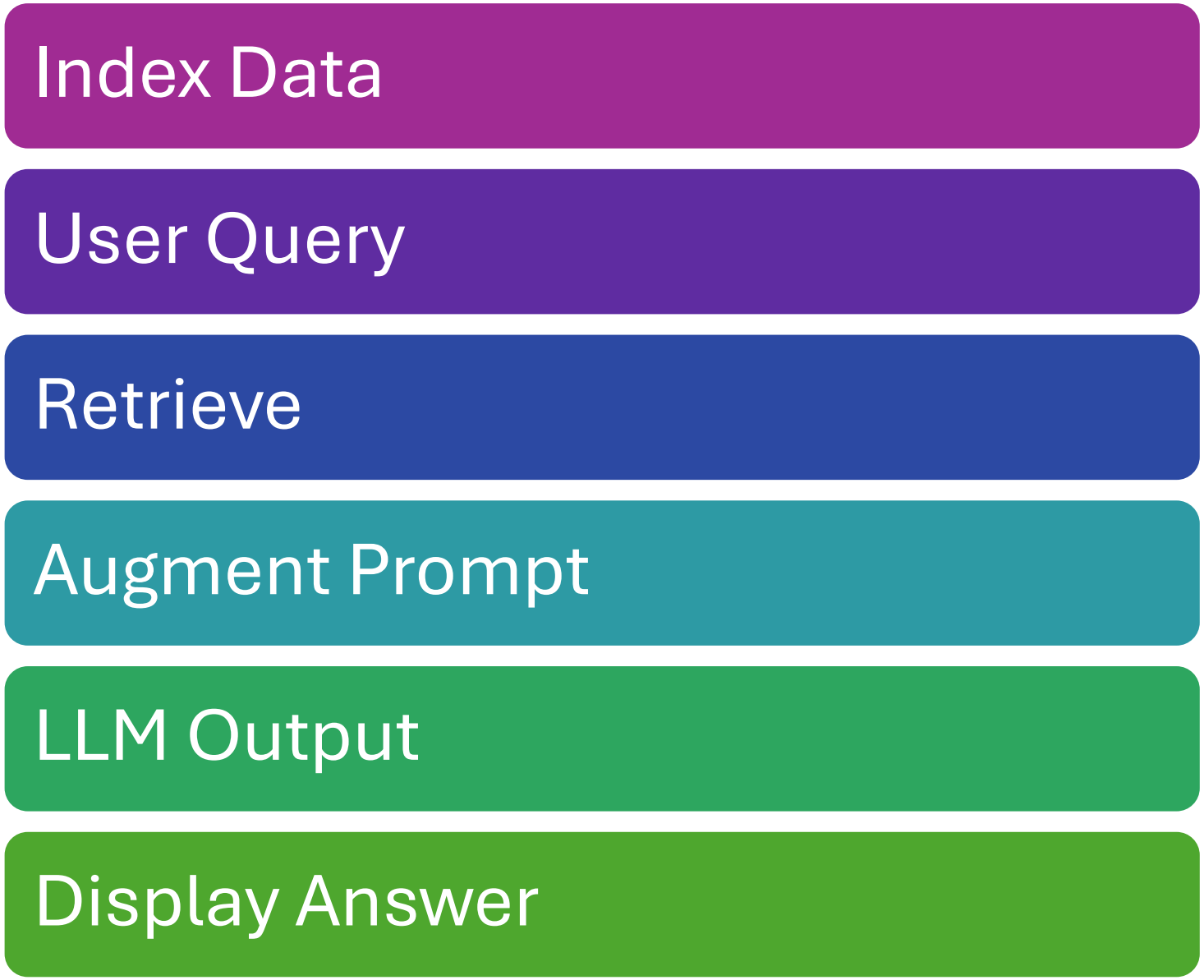
INVENTORY, LEGAL, OR

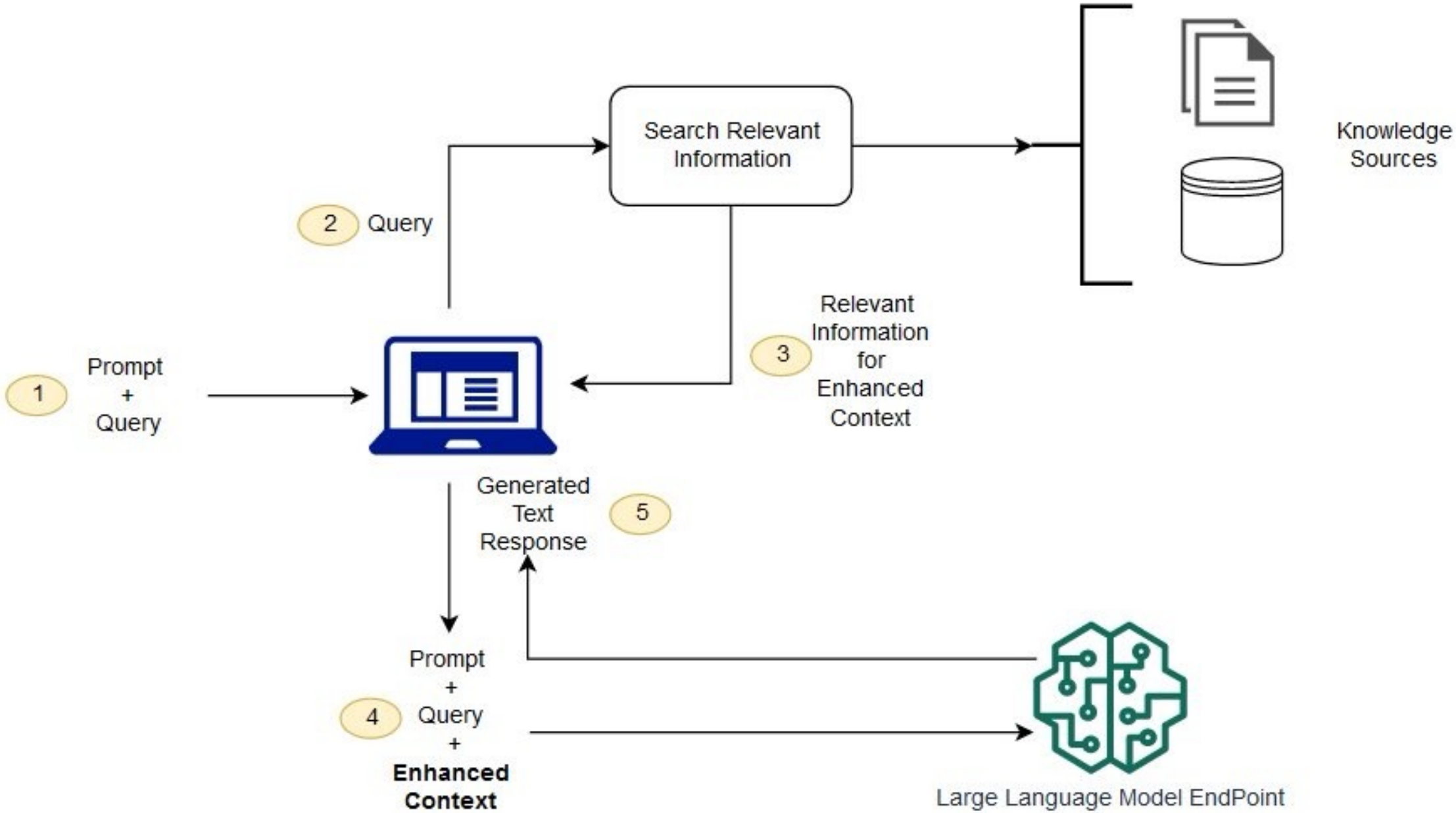
OPERATIONAL QUERIES

Why RAG is Critical for Enterprise GenAI ?

Faster and cheaper to implement than full model retraining

RAG Workflow




Search Relevant Information
Knowledge Sources
2 Query
1 Prompt + Query
3 Relevant Information for Enhanced Context
Generated Text Response
5
4 Prompt + Query + Enhanced Context
Large Language Model EndPoint

Index Data

Convert Walmart internal documents

e.g., return policy PDFs, inventory logs

into embeddings and

store them in a vector database.

User Query

Customer asks via chatbot:

“Can I return a blender bought 40 days ago?”

Retrieve

RAG searches the vector database,

finds recent “Return Policy” doc,

30–45 day rules.

Augment Prompt

Q: Can I return a blender bought 40 days ago?

Context: “Blender returns accepted within 45 days

of purchase with receipt.”

LLM Output

“Yes – you can return the blender within

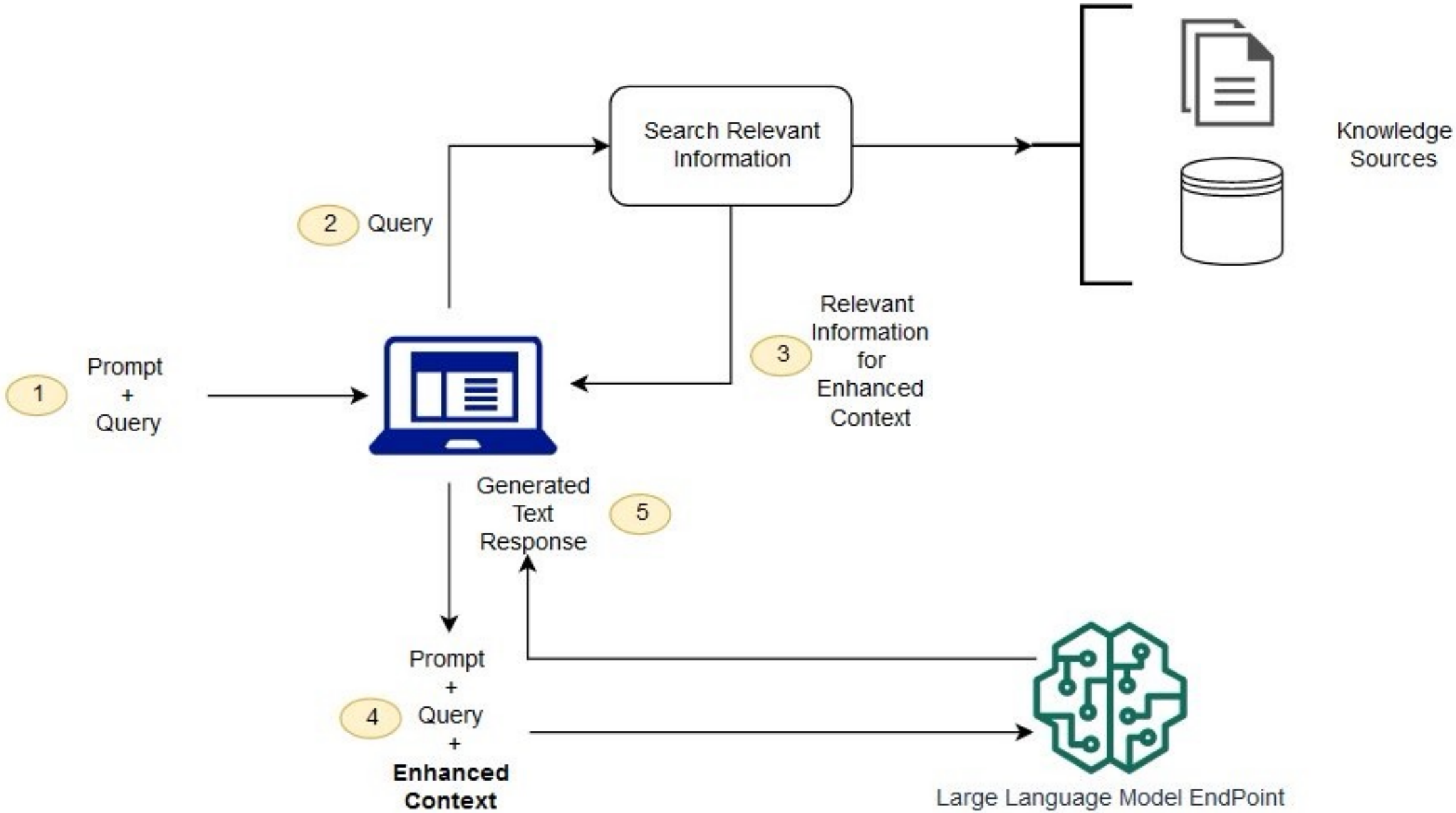
45 days of purchase with receipt.”

Display Answer

Chatbot responds

confidently,

grounded in live data.


Search Relevant Information
Knowledge Sources
2 Query
1 Prompt + Query
3 Relevant Information for Enhanced Context
Generated Text Response
5
4 Prompt + Query + Enhanced Context
Large Language Model EndPoint

Benefits for Walmart GenAI

- Customer Confidence
- Efficiency
- Auditability
- Scalability

Customer Confidence

REPLIES ARE

FACT-BASED AND

POLICY-GROUNDED,

REDUCING SUPPORT ISSUES.

Efficiency

Real-time,

accurate assistance

without extensive model retraining.

Auditability

Context can include citations

Based on Walmart’s Return Policy doc

Scalability

CAN BE APPLIED TO OTHER AREAS

PRICING, STOCK QUERIES,

LEGAL GUIDELINES.

Why Walmart Needs RAG?

Without RAG:

LLMs might **hallucinate** or provide outdated info.

Sensitive enterprise data stays siloed.

Why Walmart Needs RAG?

With RAG:

Answers draw from real-time **internal data**, boosting accuracy.

No full retraining—just update your vector store when policies change.

Enables **trusted, compliant, enterprise-grade AI experiences**.

LangChain Integration

Surendra Panpaliya

GKTCS Innovations

https://www.gktcs.com

Agenda

LangChain with OpenAI / Gemini

Document loading,

Retriever setup,

Chain creation

What is LangChain?

A powerful **framework**

for building applications

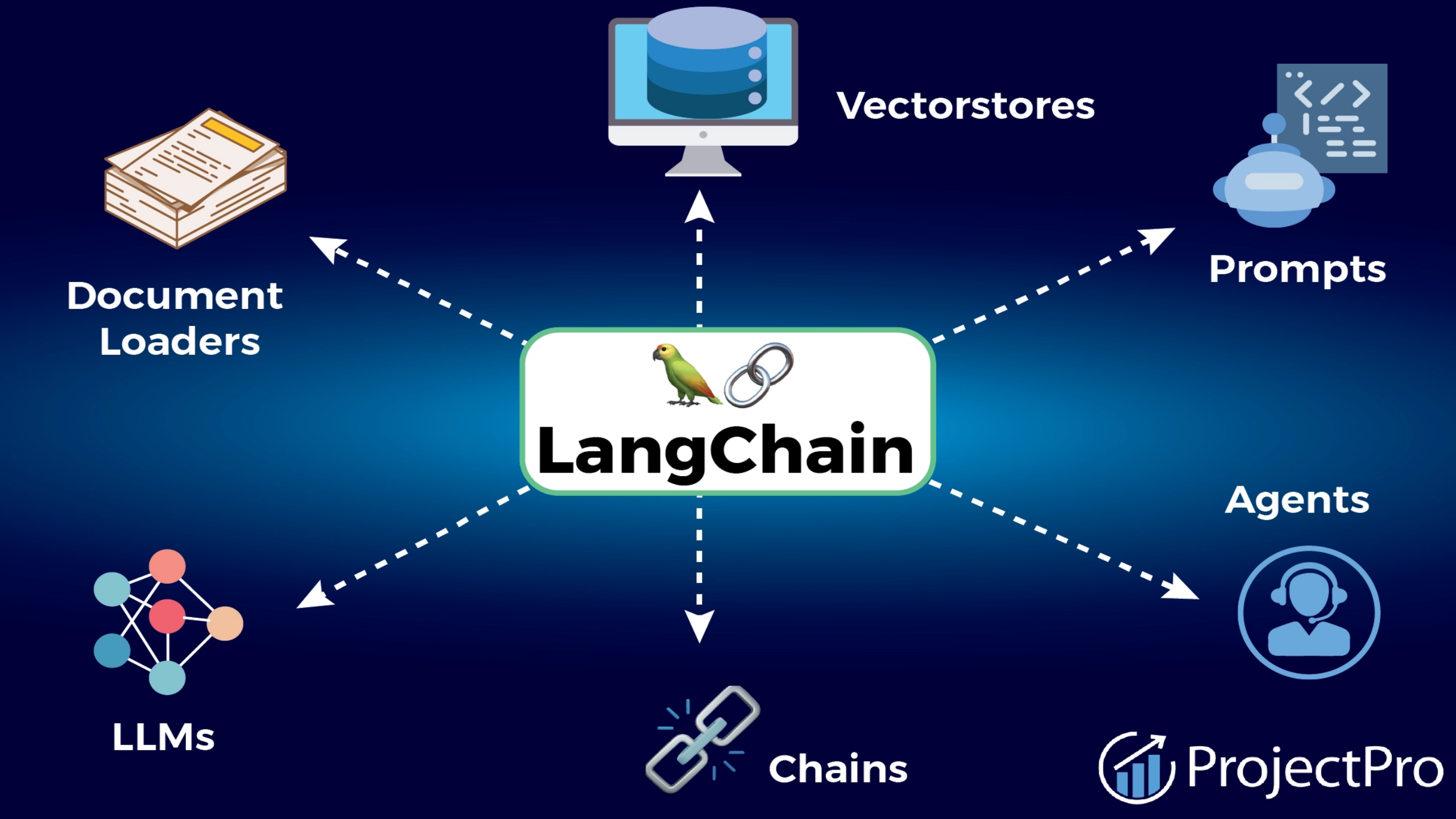
using Large Language Models (LLMs).

What is LangChain?

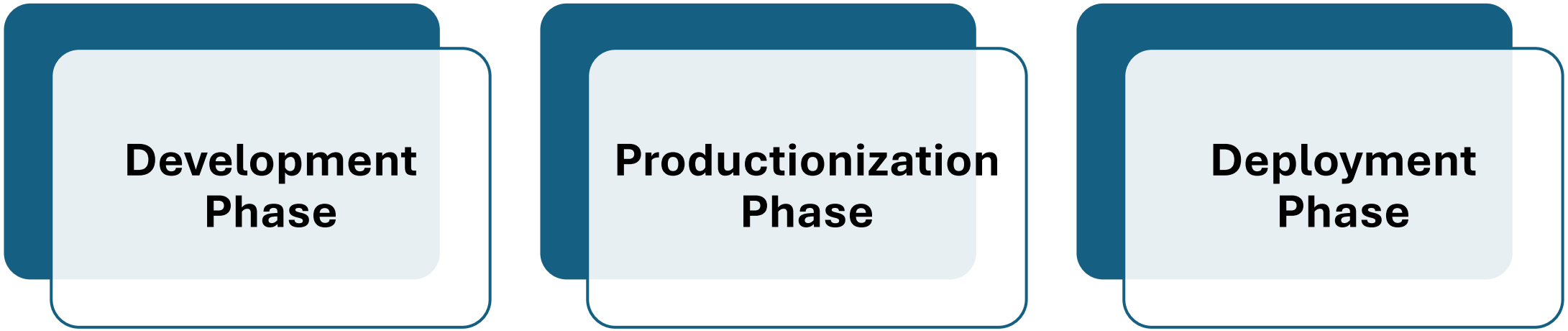
SIMPLIFIES
EVERYTHING

FROM DEVELOPMENT
TO DEPLOYMENT

OF LLM-BASED
SOFTWARE SYSTEMS.


Vectorstores
Prompts
Document Loaders
LangChain
Agents
LLMs
Chains
ProjectPro

Key Capabilities at a Glance



1. Development Phase

USE **OPEN-SOURCE COMPONENTS**

THIRD-PARTY TOOLS

TO BUILD **LLM APPLICATIONS.**

1. Development Phase

LangGraph

enables

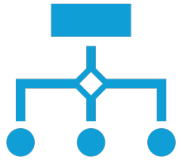
to create

stateful agents

1. Development Phase

Support

Streaming outputs



Human-in-the-loop feedback

Tool usage coordination

2. Productionization Phase

LangSmith

Inspect and debug LLM applications

Monitor performance

Evaluate accuracy & quality

Continuously optimize models and prompts

3. Deployment Phase

Turn LangGraph apps

Into APIs

Production-ready AI Assistants

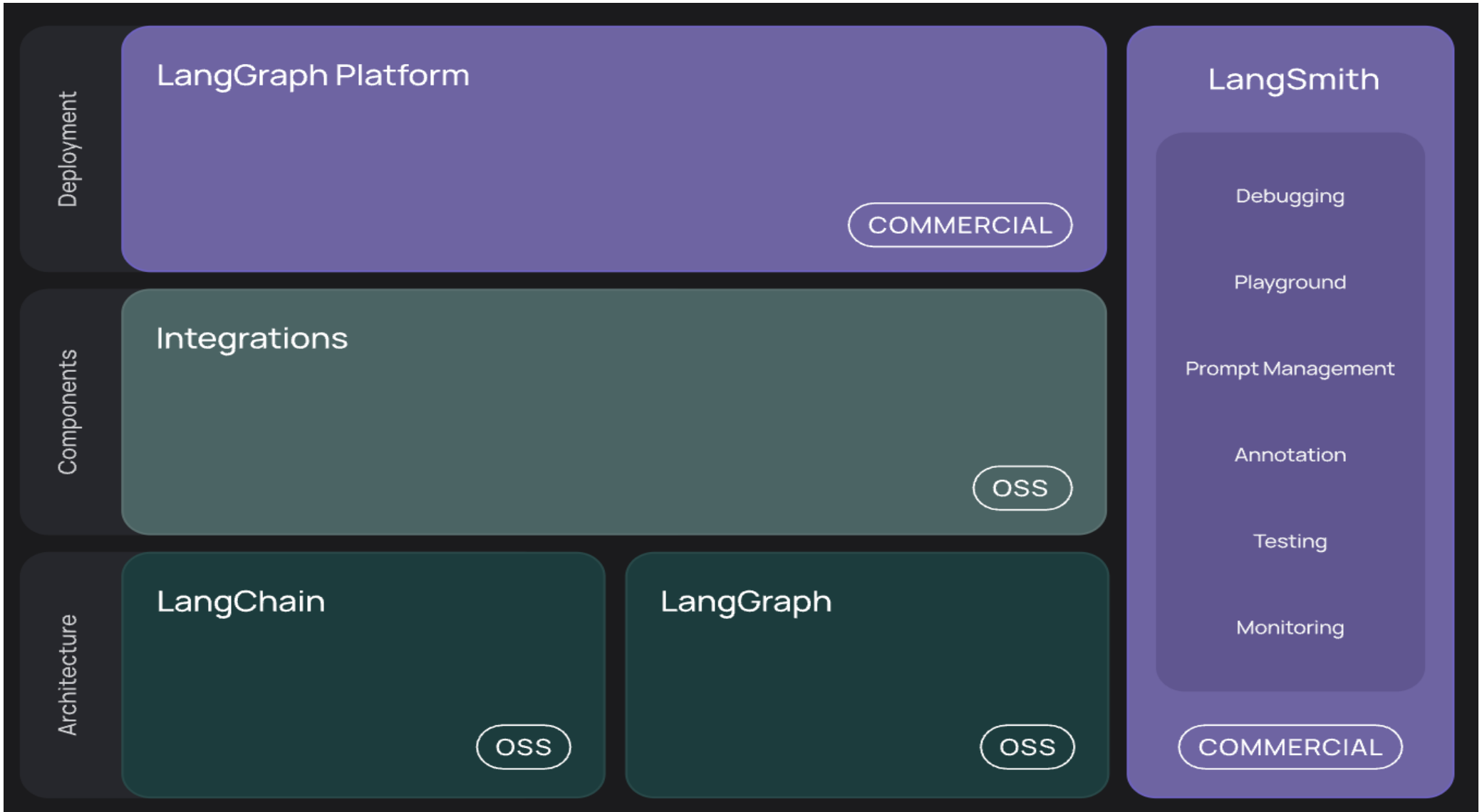
3. Deployment Phase

USE **LANGGRAPH PLATFORM**

FOR CLOUD-BASED HOSTING

AND ORCHESTRATION

LangChain + LangGraph Architecture



Architecture Overview

Module	Purpose
langchain-core	Base interfaces & abstractions for LLMs, tools, and prompts
langchain	Contains higher-level components like agents, chains, and retrieval strategies

Architecture Overview

Module	Purpose
langchain-openai, langchain-anthropic,	Integration packages maintained by LangChain + provider teams
langchain-community	Third-party and community-contributed integrations

Architecture Overview

Module	Purpose
langgraph	Agent orchestration framework with support for memory, streaming, and persistence
LangSmith	Observability and evaluation platform for LangChain apps
LangGraph Platform	Cloud deployment & lifecycle management of LangChain applications

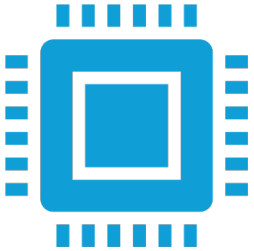
Reference

- https://python.langchain.com/docs/introduction/

Ecosystem Integration

LangChain integrates

with **hundreds of tools**,



models, and providers

GenAI Tools Providers

OpenAI,

Anthropic,

Hugging Face,

Cohere,

Google Vertex

Vector stores

Pinecone,

Chroma,

FAISS

Tools

Search APIs,

SQL databases,

file readers,

Python REPL

Cognitive Architecture Features

LangChain’s design

encourages building apps

that simulate thinking agents with:

Memory, Tool usage

Multi-step planning,

Retrieval-Augmented Generation (RAG)

Cognitive Architecture Features

Memory: Store past interactions

Tool usage: Perform calculations, search, look up info

Multi-step planning: Use planners, routers, chains

Retrieval-Augmented Generation (RAG)

Fetch relevant knowledge before answering

Summary of Key Points

Category	Summary
Framework Type	Agentic LLM application development
Core Components	Chains, Agents, Tools, LangGraph
Lifecycle Coverage	Build → Monitor (LangSmith) → Deploy (LangGraph Platform)

Summary of Key Points

Category	Summary
Extensibility	Integrates with major LLM providers and tools
Production Readiness	Enables streaming, persistence, tracing, and evaluation

Why LangChain?

LangChain makes it easy

to build complex GenAI pipelines using:

Document Loaders

Retrievers

Chains/Agents

Why LangChain?

Document Loaders:

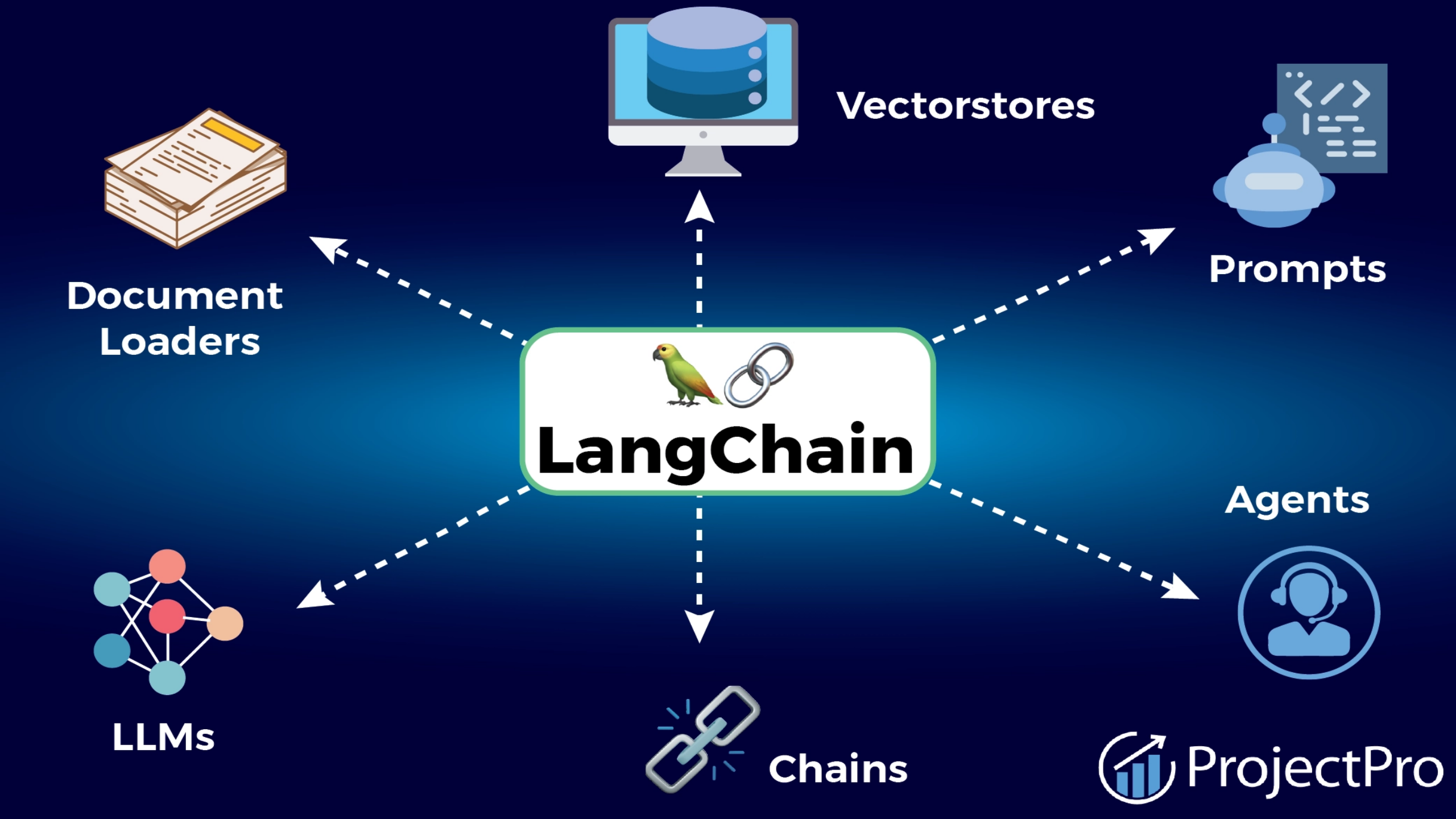
bring in PDFs, Excel, web pages

Retrievers:

fetch relevant info via embeddings

Chains/Agents: orchestrate workflows

e.g., search → generate → respond


Vectorstores
Prompts
Document Loaders
LangChain
Agents
LLMs
Chains
ProjectPro

Loading Walmart Documents

```
from langchain.document_loaders import PDFLoader

loader = PDFLoader("/data/walmart_return_policy.pdf")

docs = loader.load()  # 🌈 policies, manuals, docs
```

Creating Embeddings + Retriever

```
#Using OpenAI embeddings


from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores import Chroma


embeddings = OpenAIEmbeddings()
vectordb = Chroma.from_documents(docs, embeddings)
retriever = vectordb.as_retriever()
```

How it Ties to Walmart?

Document loaders ingest manuals, guidances, product info

Retrievers pull only relevant slices using vector search

Chains inject context to generate reliable replies in live chat or voice kiosks

Benefits for Walmart

Accurate, grounded answers — reduces hallucination

Real-time updates — policies, inventory stay fresh in vectordb

Modular and scalable — swap LLMs (OpenAI, Gemini) as needed

Multimodal possibilities — e.g., image + text via Gemini

Walmart LLM Gateway

- Access control and internal model orchestration
- Capabilities and supported use cases
- Comparison:
- Azure OpenAI
- Google GenAI
- Open Source (LLaMA, Mistral)

Embedding & Vector Store

Managed Milvus:

Vector DB-as-a-Service

Embedding generation pipeline:

Sentence Transformers, HuggingFace

Document ingestion and search

What is Managed Milvus?

A vector database-as-a-service

optimized for similarity search

large-scale embeddings .

What is Managed Milvus?

Built for storage,

query,

indexing,

scaling across billions of vectors

What is Managed Milvus?

Offers both

open-source standalone and

fully **managed cloud service**

options via Zilliz Cloud .

Walmart uses case

Multilingual product search and

internal document retrieval.

Embedding Generation Pipeline

Use Sentence Transformers or

HuggingFace models to convert text

(product titles, reviews, policies)

into dense vectors

Embedding Generation Pipeline

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('all-MiniLM-L6-v2')
vector = model.encode("₹999 kids backpack with cartoon print")
```

Document Ingestion & Vector DB Setup

Load documents: CSVs of product data,

PDFs of policy docs

Generate embeddings

Insert into Milvus

Document Ingestion & Vector DB Setup

```
from pymilvus import Collection, connections


connections.connect()
collection = Collection("walmart_products", schema=...)
collection.insert([vectors, product_ids])
```

Document Ingestion & Vector DB Setup

Supports **auto-scaling**, **sharding**, and **fault tolerance** .

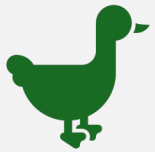
Zilliz Cloud offers fully managed infrastructure for ease and performance .

Semantic Search Workflow

- Customer queries: “cartoon school bag under ₹1000.”
- Convert query to embedding using the same model.
- Use collection.search in Milvus to find top-k similar product vectors.
- Load and display matching product details in app/website/chat.

Walmart use

Enhances search relevance drastically



especially for “long-tail” or

descriptive queries

Why This Matters for Walmart?

Improves **search relevance** and

product discovery on Walmart.com.

Enables **fast retrieval**

from internal documents

like inventory or policy manuals.

Why This Matters for Walmart?

Fully **scalable**, secure, and

integrates well with existing GenAI stack

using APIs like **LangChain retrievers** .

Managed Milvus reduces ops burden

while supporting enterprise-scale queries

Summary

Stage	What Happens	Walmart Impact
Embedding Generation	Convert text/images → numeric vectors	Meaningful search, similarity matching
Vector Ingestion	Store embeddings in Milvus database	Scalable, low-latency retrieval
Semantic Search	Find similar vectors using nearest neighbor	Better product recommendations, search UX
RAG + LLM Integration	Use retrieved docs as context for LLM responses	Accurate chatbot answers, grounded responses

LLM Use Cases @ Walmart

Customer-facing chatbots

Supplier onboarding assistants

Buyer enablement tools

Internal process automation

Customer-Facing Chatbots

What it is?

AI-powered assistants

that help customers with queries

about orders, returns, and product info.

Walmart Example

Chat with Walmart

chatbot helps users with checkout,

order tracking, and refunds

Why it matters?

Provides 24/7 support and reduces live-agent volume.

Personalized and context-aware assistance enhances customer experience.

Supplier Onboarding Assistants

WHAT IT IS?

CHATBOTS THAT AUTOMATE

SUPPLIER INTERACTIONS

FROM DOCUMENT SUBMISSION TO NEGOTIATION.

Walmart Example

AI-powered negotiation bot achieved a

64% negotiation success with suppliers,

surpassing the 20% target

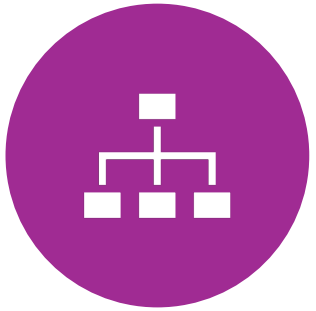
AI automates routine supplier data entries and contracts

Why it matters?

SPEEDS UP SUPPLIER ONBOARDING,

IMPROVES TERMS, AND

FREES PROCUREMENT STAFF



FOR STRATEGIC TASKS.

Buyer Enablement Tools

What it is:

Tools that assist Walmart buyers and merchandisers with analytics and product decisions.

Walmart Example

“TREND-TO-PRODUCT” AGENT USES GENAI

TO IDENTIFY TRENDING ITEMS AND

STREAMLINE PRODUCT SELECTION

Walmart Example

“Wallaby”—Walmart’s proprietary LLM

trained on internal data

enhances context-aware product insights

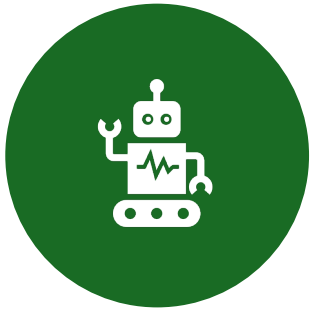
Why it matters?

Improves speed and quality of product

assortments and pricing decisions.

Internal Process Automation

WHAT IT IS?



GENAI-POWERED ASSISTANTS THAT

IMPROVE INTERNAL OPERATIONS

FOR ASSOCIATES AND CORPORATE TEAMS.

Walmart Example

Ask Sam:

Voice assistant helps store associates

locate products, check inventory,

view schedules

Walmart Example

My Assistant:

Summarizes documents,

supports HR

streamlines developer queries,

improving onboarding and troubleshooting

Why it matters

Boosts associate productivity

and job satisfaction.

Automates repetitive internal tasks,

saving time and improving accuracy.

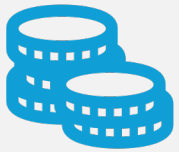
Summary

Use Case	Description	Walmart Benefit
Customer Chatbots	AI for orders, refunds, questions	24/7 support, personalized experiences
Supplier Assistance	Automating onboarding and negotiations	Faster onboarding, better terms
Buyer Tools	Trend detection and product analytics	Data-driven purchasing and assortment planning
Internal Automation	Voice/data assistants for employees	Faster access to inventory info, HR support, and developer help

ROI & Strategic Impact

Satisfying AI-powered consumer experiences

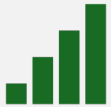
can **increase satisfaction by 20%** and



reduce costs by 10–20% .

ROI & Strategic Impact

Automating supplier negotiations



leads to **higher win rates (64%)**,

reduced negotiations time, and

cost efficiency

ROI & Strategic Impact

Internal GenAI tools

empower **50,000+ employees**,

increasing developer efficiency and

reducing manual work

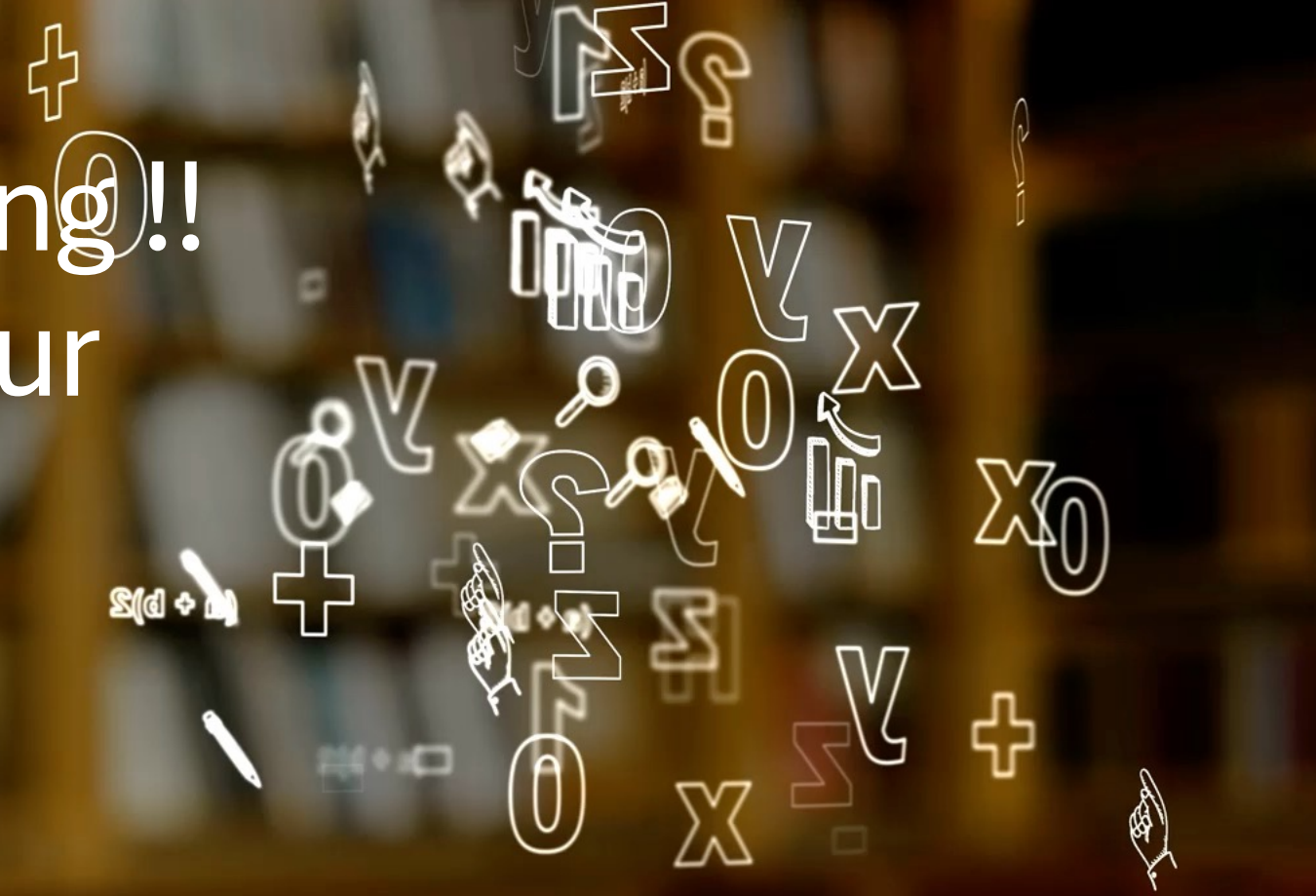
Hands-On Session

Generate embeddings from text documents

Store and query documents using Milvus vector DB

Perform similarity-based document retrieval

Happy Learning!! Thanks for Your Patience ☺



Surendra Panpaliya

GKTCS Innovations