Multimodal and Multimodal RAG

Topics Covered

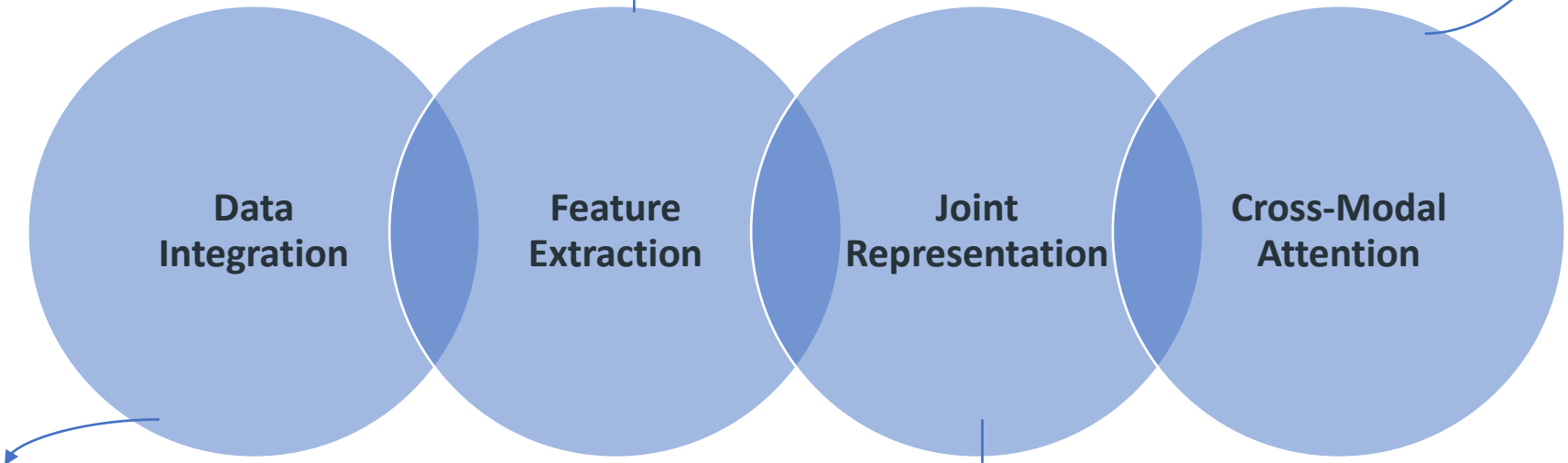
- Building blocks of Multimodal Large Language Models – # 4
- Multimodal LLM Processing Flow - #5
- Multimodal LLM Processing Steps – Critical Activities and Goal - #6
- Types of Multimodal LLM - #7 & 8
- Feature comparison - #9 & 10
- Multimodal RAG - #11
- Data Flow for Multimodal RAG – Data Flow - # 12
- Heart of RAG is Embedding & Various Multimodal Vector DB - #13 & 14
- Deployment of Multimodal RAG in Production consideration / Steps - #15, 16, 17
- Industry applications of Multimodal RAG - #18 & 19

Before Multimodal RAG – Let's understand Multimodal LLM (up to slide #10) first
and
later RAG with **MMLLM**

Building blocks of Multimodal Large Language Models

This technique extracts relevant features and identifies objects and their relationships in an image while understanding the context and meaning of the accompanying text

Focus on relevant parts of the data from different modalities, improving its ability to generate coherent and contextually appropriate responses

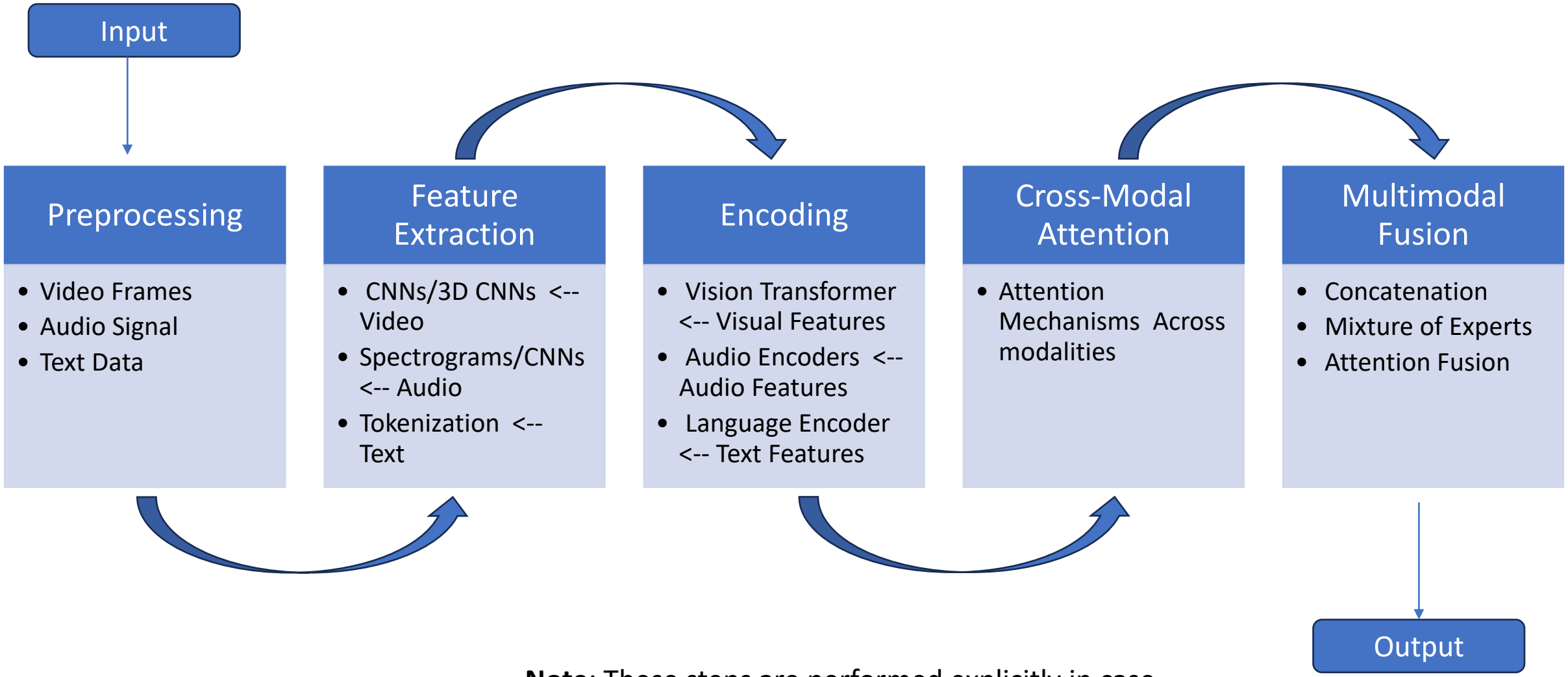


Combining data from multiple sources and information from each modality is accurately represented and integrated

The model can make inferences and generate outputs that consider all available information

Reference: www.geeksforgeeks.org

Multimodal LLM Processing Flow



Note: These steps are performed explicitly in case need to tweak the MMLLM

Reference: www.geeksforgeeks.org

Multimodal LLM Processing Steps – Critical Activities and Goal

Preprocessing

Goal: Prepare raw multimodal data for analysis by ensuring consistency and quality

Critical Activities: Ensuring synchronization and preserving modality-specific characteristics during cleaning and alignment

Feature Extraction

Goal: Extract meaningful representations from raw data for each modality.

Critical Activities: Capturing task-relevant features while minimizing noise and redundancy

Encoding

Goal: Convert extracted features into a unified, machine-readable format suitable for multimodal integration

Critical Aspect: Balancing modality-specific details with compatibility across modalities

Cross-Modal Attention

Goal: Enable the model to focus on relevant information across modalities by attending to their interactions

Critical Aspect: Balancing attention across modalities to avoid dominance by any single modality.

Multimodal Fusion

Goal: Combine information from different modalities into a unified representation for decision-making or downstream tasks

Critical Aspect: Choosing the right fusion strategy and ensuring no loss of critical information during integration.

Types of Multimodal LLM (1/2)

Name	Description
CLIP (Contrastive Language–Image Pre-training)	Designed to understand images and text by learning a wide variety of visual concepts from natural language descriptions.
DALL-E	Generates images from textual descriptions, showcasing the ability to create visual content based on detailed text prompts.
Florence	Florence is a foundation model designed for computer vision tasks. It integrates textual descriptions with visual data to perform various tasks, including image captioning and visual question answering.
ALIGN (Vision-Language Pre-training)	Model trained to understand and generate text from images by aligning visual and linguistic representations. It can perform cross-modal retrieval and zero-shot image classification.
ViLBERT (Vision-and-Language BERT)	ViLBERT extends the BERT architecture to handle visual and textual data simultaneously. It can be used for tasks such as visual question answering and visual commonsense reasoning.
VisualBERT	VisualBERT integrates visual and textual information using a unified BERT-like architecture. It is applied to tasks like image-caption matching and visual question answering.
LXMERT (Learning Cross-Modality Encoder Representations from Transformers)	LXMERT is a model that encodes visual and textual data using separate transformers and then merges the information for tasks like visual question answering and image captioning.

Types of Multimodal LLM (2/2)

Name	Description
UNITER (Universal Image-Text Representation Learning)	UNITER learns joint representations of images and text, achieving state-of-the-art results on several vision-and-language tasks, such as visual question answering and image-text retrieval
ERNIE-ViL (Enhanced Representation through Knowledge Integration)	Description: ERNIE-ViL enhances visual-linguistic pre-training by integrating structured knowledge, improving performance on tasks such as visual question answering and image captioning
M6 (Multi-Modality to Multi-Modality Multilingual Pre-training)	Description: M6 is designed to handle multimodal data across multiple languages, integrating text and images for tasks like cross-lingual image captioning and visual question answering.

Feature comparison (1/2)

Key Activities	Preprocessing	Feature extraction	Encoding	Cross-modal attention	Multimodal fusion
CLIP (Contrastive Language–Image Pre-training)	Yes	Yes	Yes	Not Done Explicitly	Yes
DALL-E	Yes	Yes	Yes	Yes	Not Done Explicitly
Florence	Yes	Yes	Yes	Not Done Explicitly	Not Done Explicitly
ALIGN (Vision-Language Pre-training)	Yes	Yes	Yes	Not Done Explicitly	Not Done Explicitly
ViLBERT (Vision-and-Language BERT)	Yes	Yes	Yes	Yes	Yes
VisualBERT	Yes	Yes	Yes	Yes	Yes

Note Done Explicitly: This requires additional code and dependency/module to be considered

Feature comparison (2/2)

Key Activities	Preprocessing	Feature extraction	Encoding	Cross-modal attention	Multimodal fusion
LXMERT (Learning Cross-Modality Encoder Representations from Transformers):	Yes	Yes	Yes	Yes	Yes
UNITER (Universal Image-Text Representation Learning)	Yes	Yes	Yes	Yes	Yes
ERNIE-ViL (Enhanced Representation through Knowledge Integration)	Yes	Yes	Yes	Yes	Yes
M6 (Multi-Modality to Multi-Modality Multilingual Pre-training)	Yes	Yes	Yes	Yes	Yes

Note Done Explicitly: This requires additional code and dependency/module to be considered

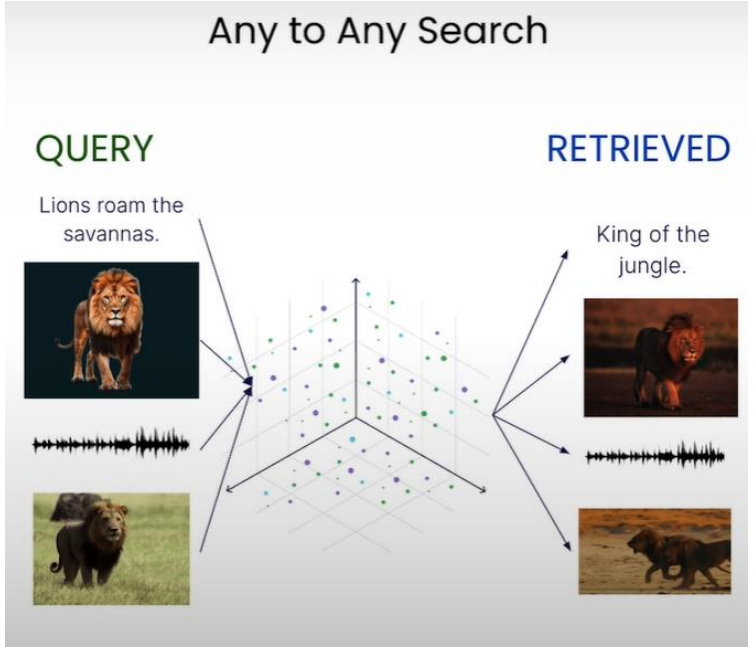
Multimodal RAG

Multimodal RAG extends the RAG framework to work with multimodal inputs and external multimodal knowledge sources.

It fits into **multimodal LLMs** by providing **retrieval and augmentation capabilities** that enhance the model's ability to reason, generate, and interact with multimodal data. Together, they create a powerful system capable of solving complex tasks that span multiple modalities.

Why **Multimodal** RAG:

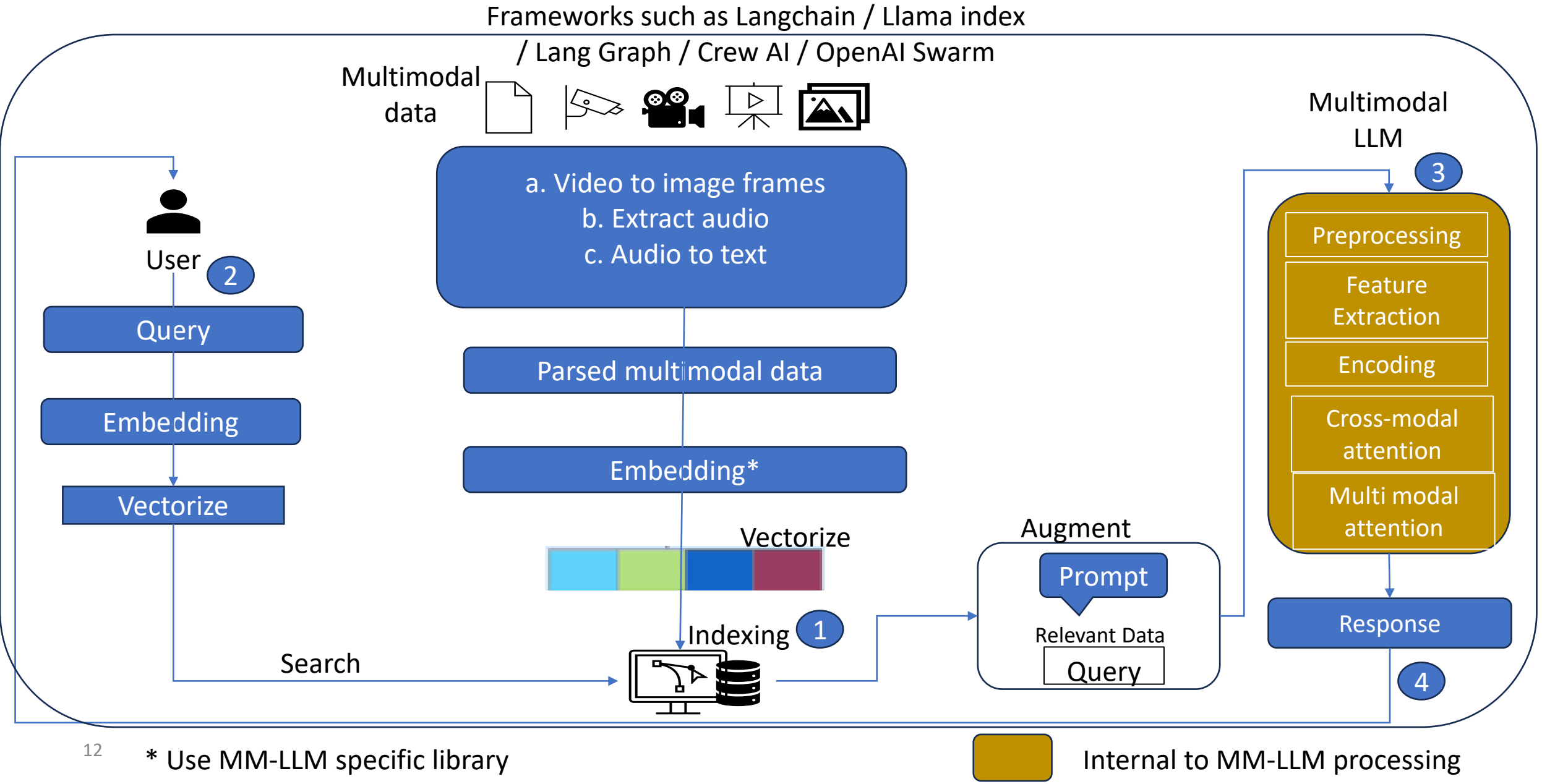
- Rich Data sources
- Better context understanding
- Complex query handling
- Real-world scenarios:



Creating common **spatial** space – irrespective of modality

Reference: https://www.deeplearning.ai/

Data Flow for Multimodal RAG – submodules



* Use MM-LLM specific library

Internal to MM-LLM processing

Heart of RAG is Embedding….

Vectorize knowledge base

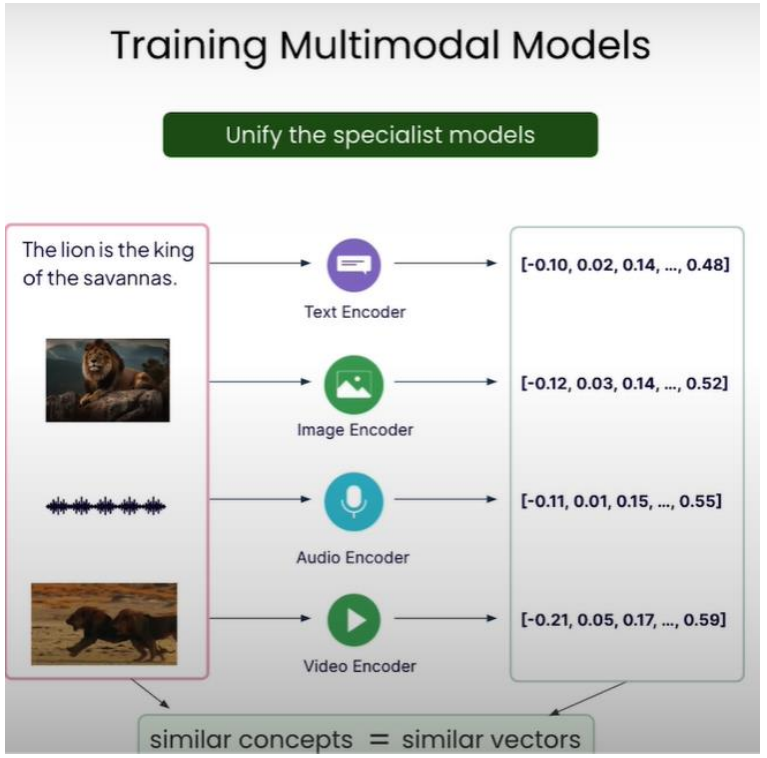
The Purpose: Creating common vector space for grounding and context generation
Such as :
Text, Images, Audio

Feature Extraction

The Purpose: Embeddings act as compact and meaningful representations of raw modality-specific data.
Such as :
Text: Word embeddings (e.g., Word2Vec, BERT).
Images: Feature vectors from CNNs (e.g., ResNet).
Audio: Embeddings from models like Wav2Vec.

Encoding

The purpose involves mapping extracted embeddings into a shared latent space or transforming them for compatibility across modalities.
Such as: Mapping text and image embeddings into a unified vector space using a multimodal encoder (e.g., CLIP).



Thus, embeddings bridge feature extraction and encoding as foundational building blocks for downstream tasks like attention and fusion

Reference: https://www.deeplearning.ai/

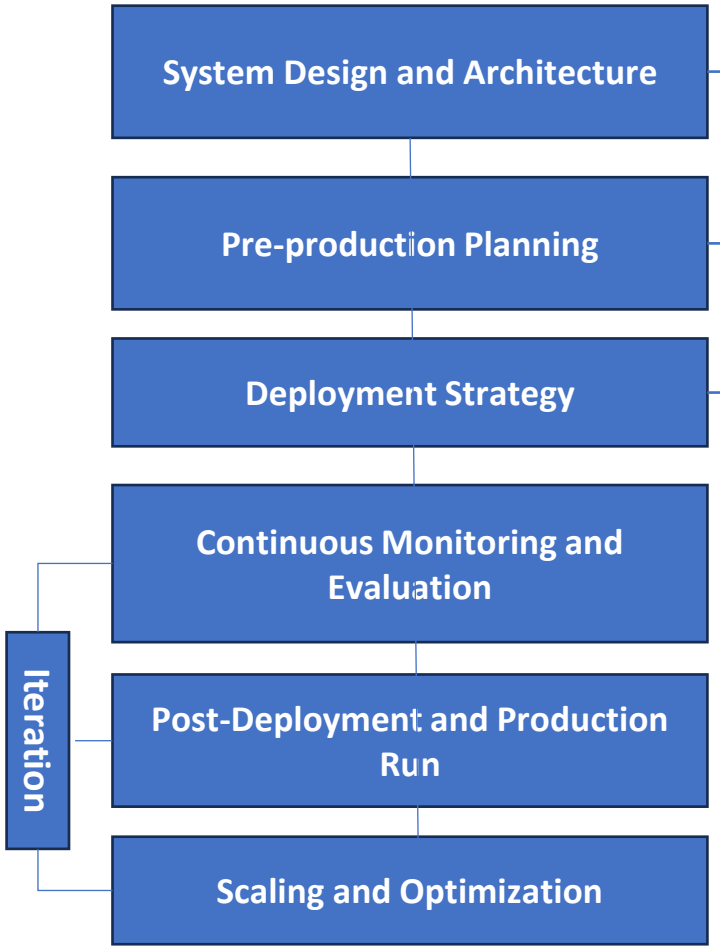
Various Multimodal Vector DB

Below is a comparison table highlighting the features of the top vector databases discussed before:

Feature	Chroma	Pinecone	Weaviate	Faiss	Qdrant	Milvus	PGVector
Open-source	✅	❎	✅	✅	✅	✅	✅
Primary Use Case	LLM Apps Development	Managed Vector Database for ML	Scalable Vector Storage and Search	High-Speed Similarity Search and Clustering	Vector Similarity Search	High-Performance AI Search	Adding Vector Search to PostgreSQL
Integration	LangChain, LlamaIndex	LangChain	OpenAI, Cohere, HuggingFace	Python/NumPy, GPU Execution	OpenAPI v3, Various Language Clients	TensorFlow, PyTorch, HuggingFace	Built into PostgreSQL ecosystem
Scalability	Scales from Python notebooks to clusters	Highly scalable	Seamless scaling to billions of objects	Capable of handling sets larger than RAM	Cloud-native with horizontal scaling	Scales to billions of vectors	Depends on PostgreSQL setup
Search Speed	Fast similarity searches	Low-latency search	Milliseconds for millions of objects	Fast, supports GPU	Custom HNSW algorithm for rapid search	Optimized for low-latency search	Approximate Nearest Neighbor (ANN)
Data Privacy	Supports multi-user with data isolation	Fully managed service	Emphasizes security and replication	Primarily for research and development	Advanced filtering on vector payloads	Secure multi-tenant architecture	Inherits PostgreSQL's security
Programming Language	Python, JavaScript	Python	Python, Java, Go, others	C++, Python	Rust	C++, Python, Go	PostgreSQL extension (SQL-based)

Reference: https://www.datacamp.com

Deploying a Multimodal **RAG (Retrieval-Augmented Generation)** system in production (1 /2)

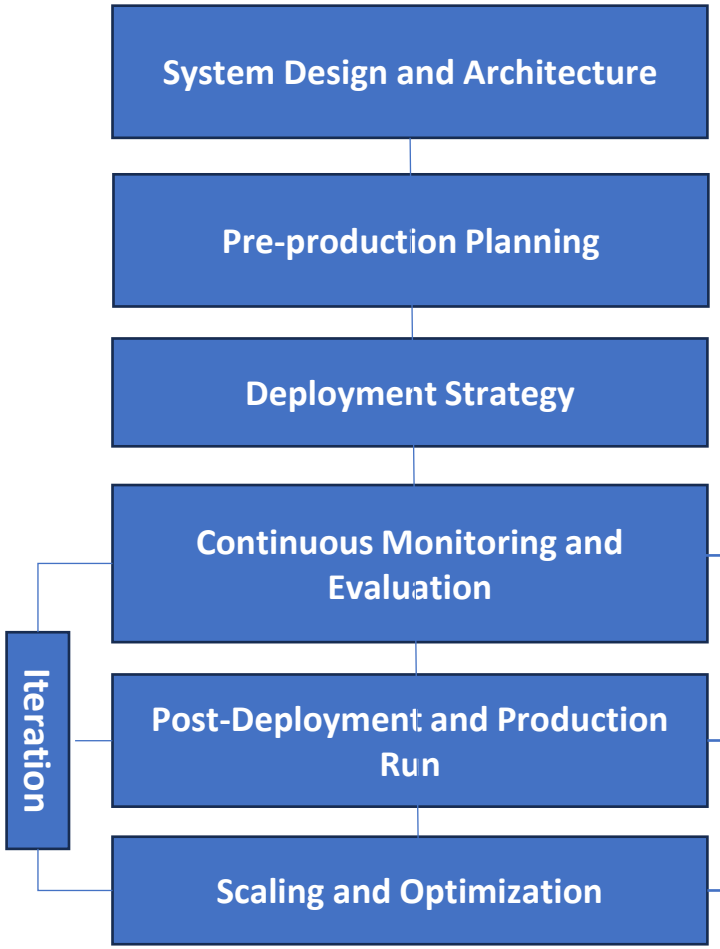


- **Core Components**: MMLLM & Retriever
- **Workflow Integration :** Input, Processing ,Retrieval, and Generation

- **Data Collection and Preprocessing**: preprocess the data sources (images, text, videos, etc.)
- **Model Training and Fine-tuning:** Train the multimodal LLM , Fine-tune models
- **Integration with Existing Infrastructure:** existing back-end systems like e-commerce platforms, CRM systems, APIs for real-time data access

- **Infrastructure Considerations**: Cloud vs On-premises, **Containerization**: Use **Docker** or **Kubernetes** , **Scaling**
- **Model Deployment:** Retrieval Model (FAISS, Elasticsearch) , **MMLLM on GPU , Inference API** (create an API layer that interacts with the multimodal model)
- **Data Storage & Indexing:** AWS S3 for images, or a vector database like Pinecone for text/image embeddings, Indexing - e.g., FAISS, Pinecone, Chroma

Deploying a Multimodal **RAG (Retrieval-Augmented Generation)** system in production (2 /2)



- **Monitoring**: Prometheus, Grafana) to track the health of the retrieval system, model inference times, and resource usage
- **Latency:** Monitor the time it takes for the system to generate a response
- **Accuracy & Relevance:** Continuously track the relevance and accuracy of the responses
- **Logging and Debugging: C**entralized logging systems (e.g., ELK stack, Splunk) to capture logs from various components of the system for real-time debugging and error resolution

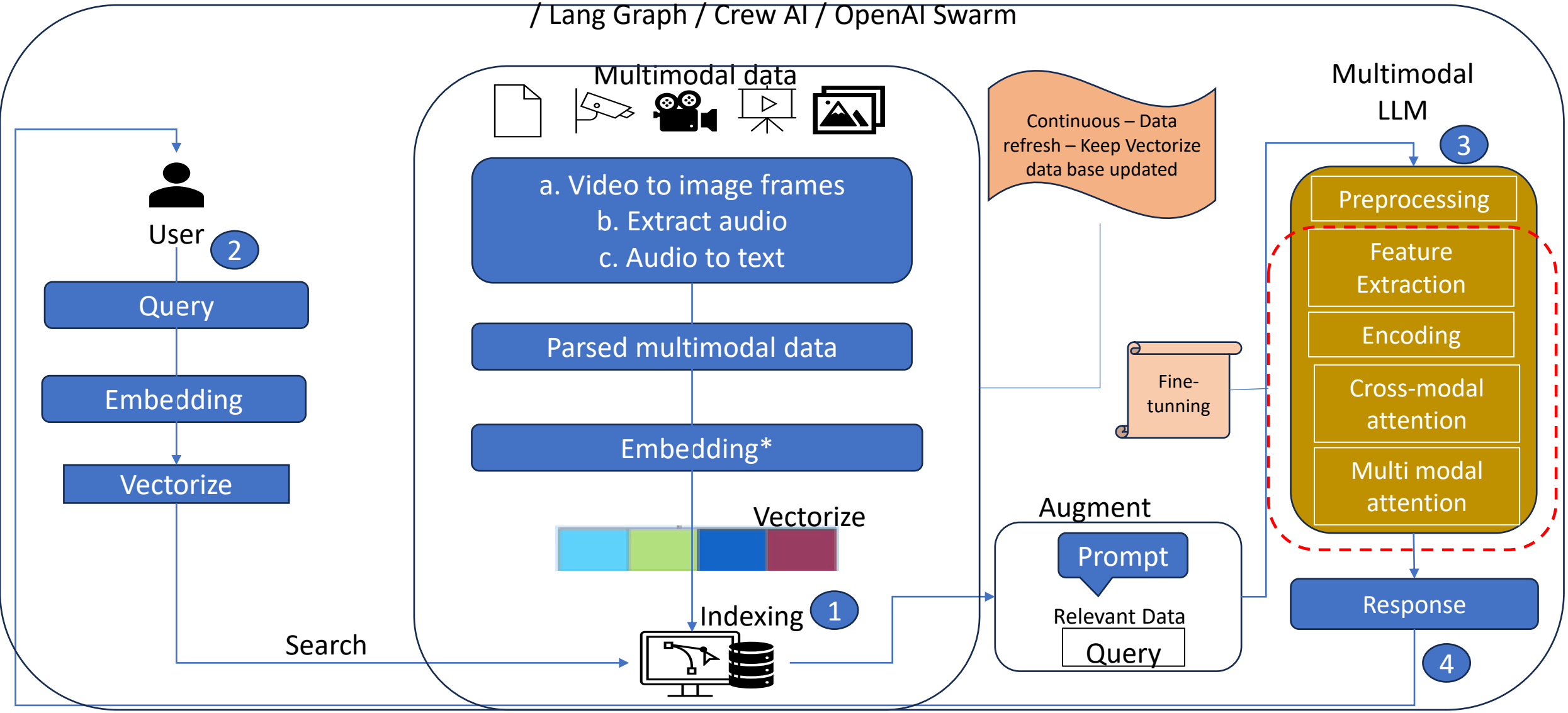
- **A/B Testing**: Conduct A/B tests by deploying multiple versions of the model to see which configuration (e.g., different retriever settings or generation strategies) provides better results and use **Canary Releases** instead of big bang approach
- **Feedback Loop:** Feedback on quality of generated responses
- **Model Updates and Retraining:** Scheduled Retraining & Incremental Learning
- **Security and Compliance:** privacy and compliance standards (e.g., GDPR for user data, HIPAA for healthcare data + Guardrails

- **Caching: F**or frequently retrieved information
- **Load Balancing:** Load balancing techniques to distribute traffic across multiple servers on for the multimodal model and retrieval service
- **Fine-Tuning Resource Usage: Model quantization** or **distillation** to reduce the size of the model and improve inference times

Data Flow for Multimodal RAG - Continuous pipeline

Frameworks such as Langchain / Llama index / Lang Graph / Crew AI / OpenAI Swarm

MLOps or **RAGOps**



* Use MM-LLM specific library

Internal to MM-LLM processing

Multimodal RAG – Industry use cases (1/2)

Use cases	Industry
Customer Onboarding - Faster and more secure onboarding process • **Multimodal LLM** – To extract the images and text • **RAG** – KYC check, check against fraud database • **Output** – Generate onboarding report **Credit Risk Assessment** - Streamlined and consistent underwriting decisions • **Multimodal LLM** - Analyzes text data and property images • **RAG -** Retrieves external credit reports, market trends, and internal lending policies • **Output -** Provides a credit risk score and tailored lending terms based on analyzed data **Fraud Detection and Prevention -** Enhanced fraud prevention with real-time insights • **Multimodal LLM -** Transcribes the call and combines it with the customer's financial data (e.g., income, risk tolerance) • **RAG** - Retrieves relevant product recommendations, market trends, and portfolio performance • **Output** - Enhanced fraud detection using dynamic retrieval and multimodal analysis	Banking
Clinical Decision Support – A doctor inputs a patient's symptoms, lab results, and a chest X-ray. • **Multimodal LLM:** Analyses text data (symptoms, lab reports) and medical images (X-ray) to identify potential conditions. • **RAG:** Retrieves relevant clinical studies, diagnostic guidelines, and similar case histories. • **Output:** Suggests a differential diagnosis, treatment options, and required tests **Radiology and Imaging Analysis -** A radiologist uploads an MRI scan for tumor analysis. • **Multimodal LLM:** Analyses the scan to detect abnormalities (e.g., tumor size, type). • **RAG:** Retrieves previous imaging records, clinical studies, and treatment guidelines. • **Output:** Generates a report summarizing findings, compares them to historical data, and suggests next steps. **Personalized Treatment Planning** - A cancer patient's genetic test results and treatment history are uploaded • **Multimodal LLM:** Processes structured data (genetic mutations) and text (patient history) • **RAG:** Retrieves research on targeted therapies and clinical trial opportunities • **Output:** Suggests a personalized treatment plan aligned with the latest medical evidence	Healthcare

Multimodal RAG – Industry use cases (2/2)

Use cases	Industry
Analyses image data: Identifies product attributes (e.g., style, colour, material) from the product image. Processes **text data**: Summarizes sales trends, customer reviews, and feedback to assess performance. Cross-**links data**: Combines visual and textual inputs to correlate product features with sales performance and customer sentiment. **RAG:** • **Retrieves relevant market insights**: Fetches similar product performance data, competitive pricing, and trends in fashion or retail. • Pulls **customer behavior studies**: Finds case studies or reports detailing consumer preferences for similar items. • **Fetches complementary product recommendations**: Retrieves data on products frequently bought with or alongside the input product. **Output:** • **Product Performance Insights**: This product (e.g., red floral dress) has high engagement on social media but slightly below-average in-store sales.“ • **Recommendations**: **Stock Optimization**: "Increase inventory for size medium and introduce size-specific promotions for faster turnover." • **Complementary Products**: "Bundle with matching handbags and shoes, as similar products in this category see 15% higher sales when sold as a set.“ • **Market Trends**: "Similar floral patterns are trending in summer collections. Consider adding a brighter palette to this product line.“ • **Actionable Suggestions**: "Revise the product description on the website to include sizing tips to reduce returns.	Retail
A car accident claim - **Image Analysis:** The LLM detects damage in the car images. Text **Processing:** It summarizes the client’s description and transcribes witness recordings. **RAG Retrieval:** The system pulls: Historical claims for similar accidents. Repair cost estimates. Policy details for coverage. With the increasing adoption of multimodal LLMs and RAG, insurance agents can expect more advanced tools that integrate: • **Geospatial data** for natural disaster claims. • **IoT sensor data** from insured assets (e.g., telematics for cars, smart home devices) • **Augmented reality (AR)** for virtual damage inspections	Insurance

Thank you