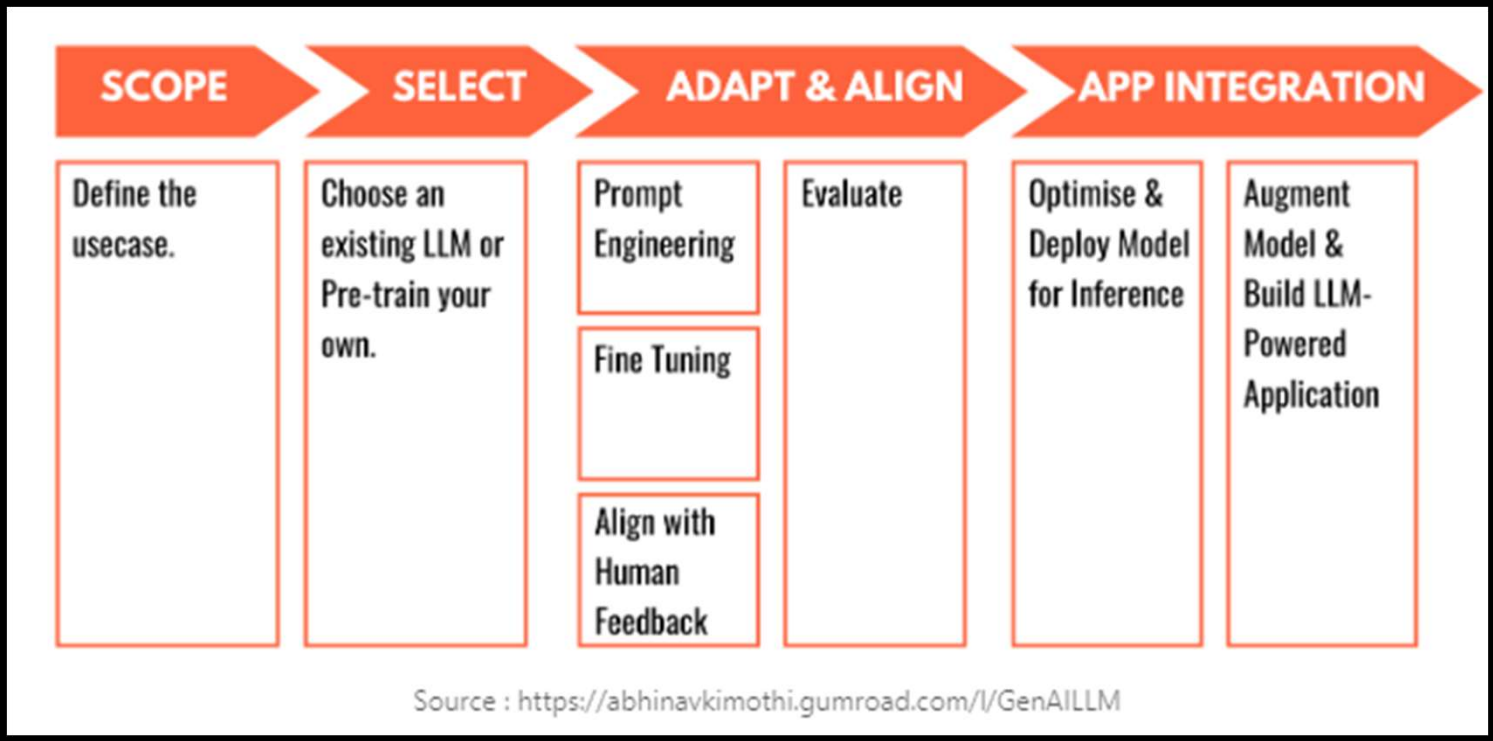
Generative AI Academy

LLM Ops

Goals for production generative AI applications

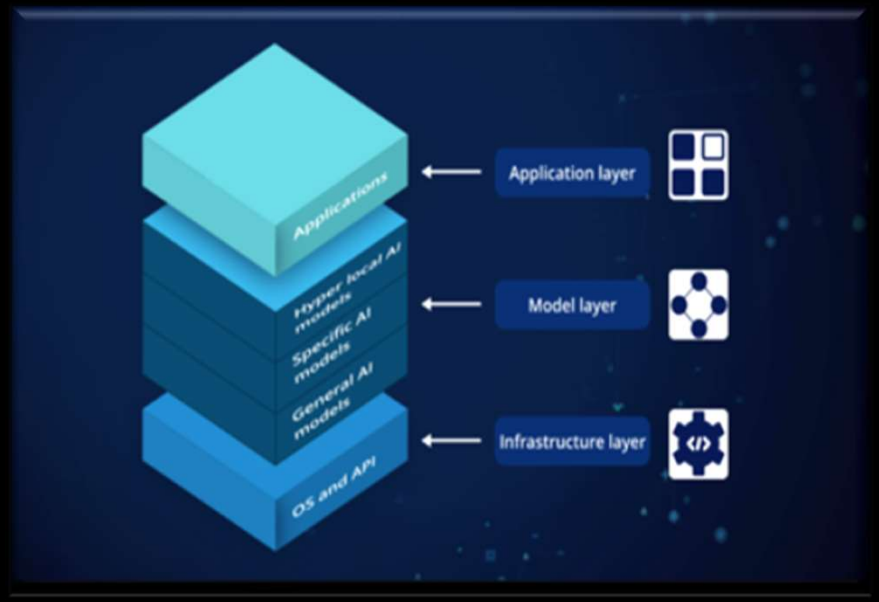
https://www.solulab.com/guide-to-llmops/

GenAI Application lifecycle



Source : https://abhinavkimothi.gumroad.com/l/GenAILLM

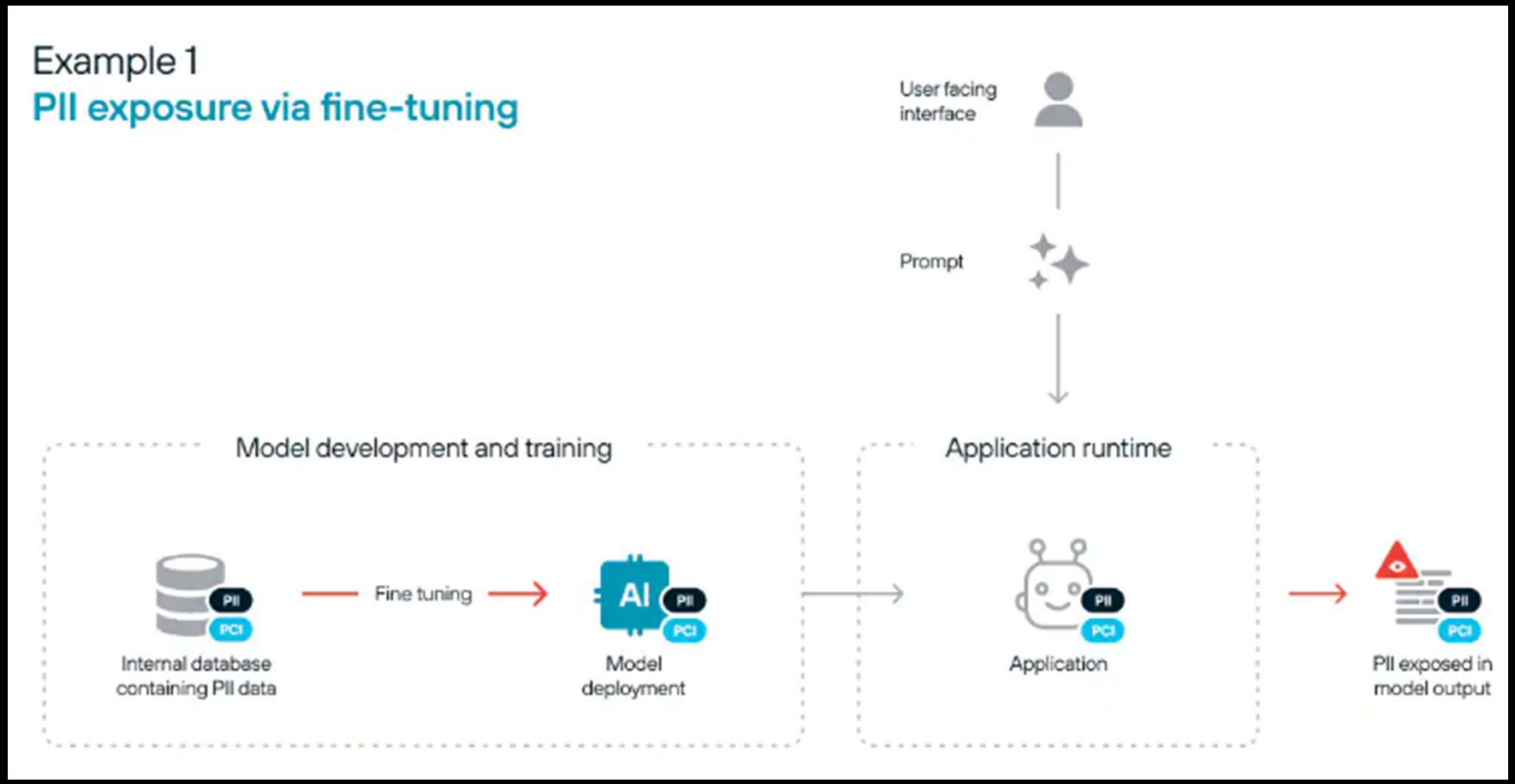
Gen AI Stack



Components used to build custom Generative AI applications

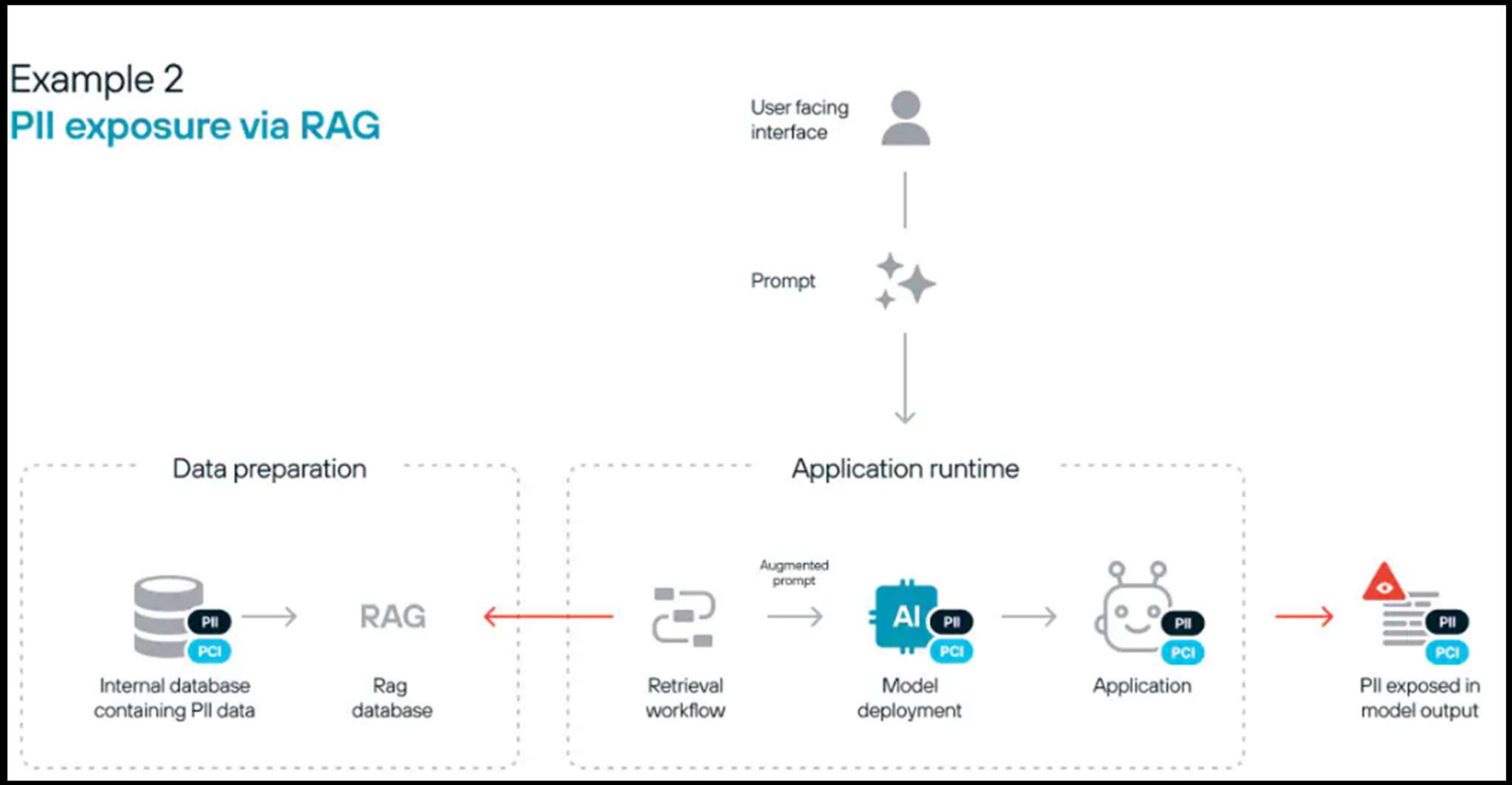
1. Foundation models
2. RAG: Vector Databases, Fine Tuning
3. Tools
   Flowise, Make, LMStudio, AnythingLLM
4. Evaluation frameworks
5. Orchestration frameworks
   Langchain, Llamaindex
6. Monitoring and Logging, Guardrails

Data Privacy considerations when using LLM



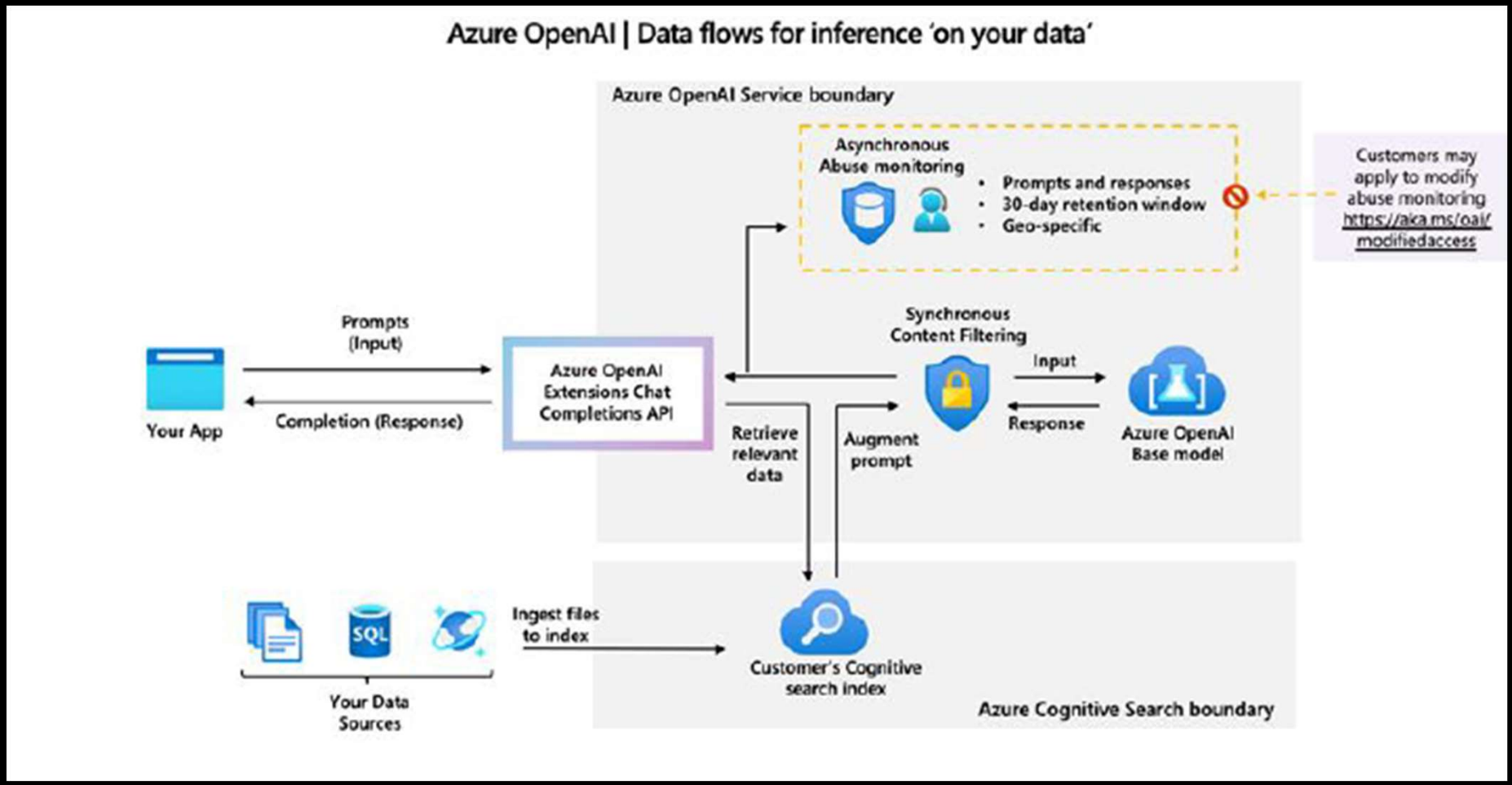
https://www.paloaltonetworks.com/blog/cloud-security/deploy-secure-llm-rag-applications/

Data Privacy considerations for RAG



https://www.paloaltonetworks.com/blog/cloud-security/deploy-secure-llm-rag-applications/

Data Architecture with Azure



Deployment Options

- On cloud: Azure, AWS etc:
  - As Managed Service:
    - Shared
    - Provisioned Manged (more secure)

- On cloud: Deploy an Open source LLM on a VM
  - More control, harder to deploy and maintain, not all models possible

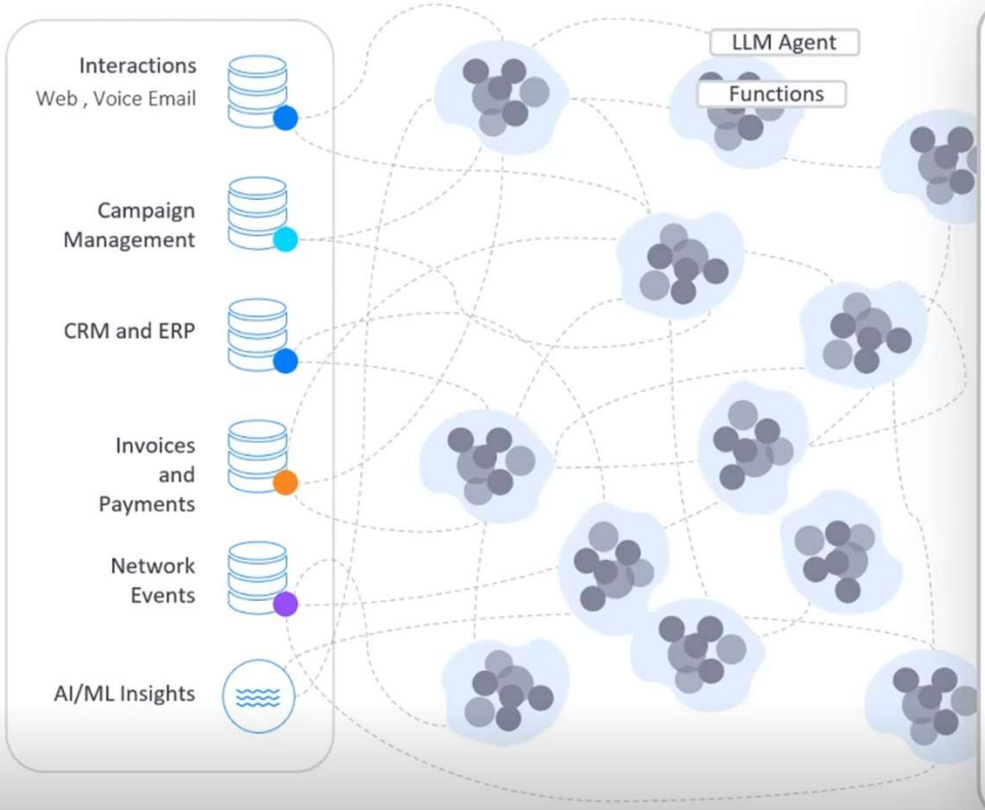
- On-prem
  - Most secure, can be deployed without internet access (intranet only)
  - Limited Model choices
  - Harder to deploy, maintain, upgrade

Data Architecture to enable Gen AI

k2view

Enterprise Data - What are your options today?

Option 1: Direct access to operational systems



Direct access to operational systems

1. Build an LLM agent focused on single domain
2. Create multiple functions in the agent
3. Each function accessing multiple enterprise data sources
4. Add thousands of agents and functions for new domains and questions

Result: The agents spaghetti

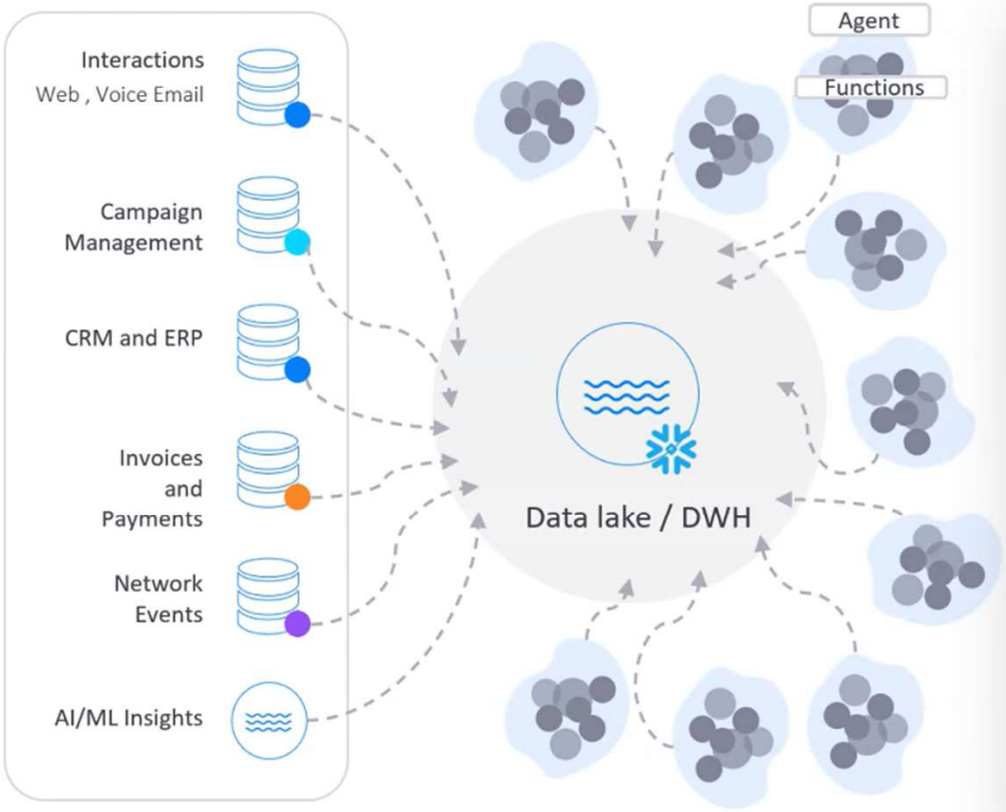
- ⚠ Fragile due to applications changes
- ⚠ Risk to operational systems due to unpredictable parallel load
- ⚠ Build & maintain 1000s of agents and functions
- ⚠ Privacy and security risks

Data Architecture to enable Gen AI with Data Lake architecture

2view

Enterprise Data - What are your options today?

Option 2: Access data in data lake / DWH



Access data in data lake

1. Build an LLM agent focused on single domain
2. Create multiple functions in the agent each function accessing the EDW/data lake with queries requiring optimization
3. Add agents and functions for new domains and questions

Result: Slow and expensive EDW/Lake

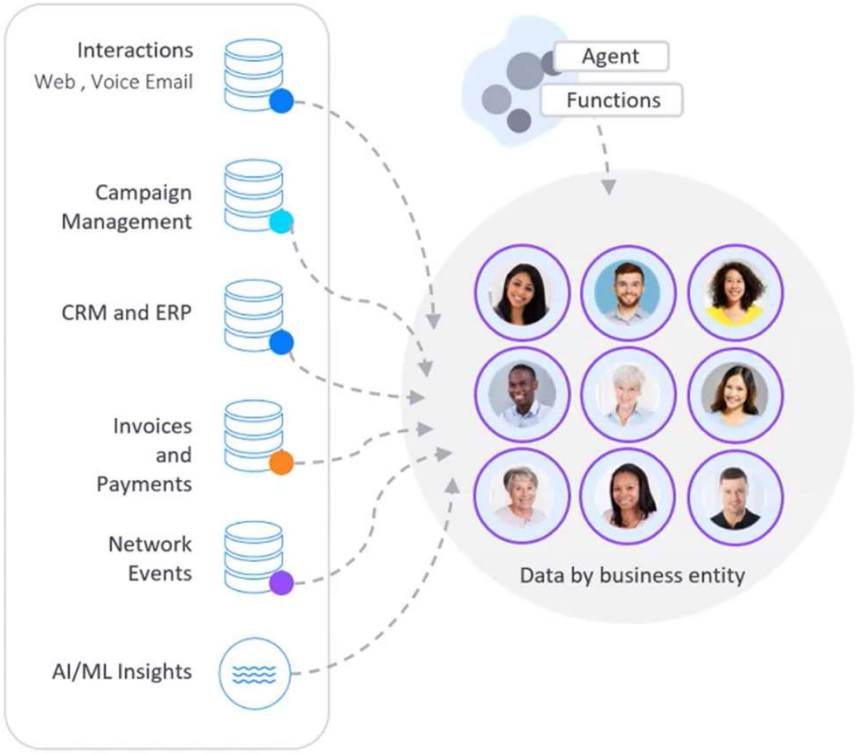
- ⚠ Privacy and security risks
- ⚠ High query costs
- ⚠ Query latency issues
- ⚠ Hard to get fresh data
- ⚠ Build and maintain 1000s of agents and functions

Data Architecture to enable Gen AI by functional modeling

k2view

Enterprise Data - What are your options today?

Option 3: Access data by business entity in a data product platform



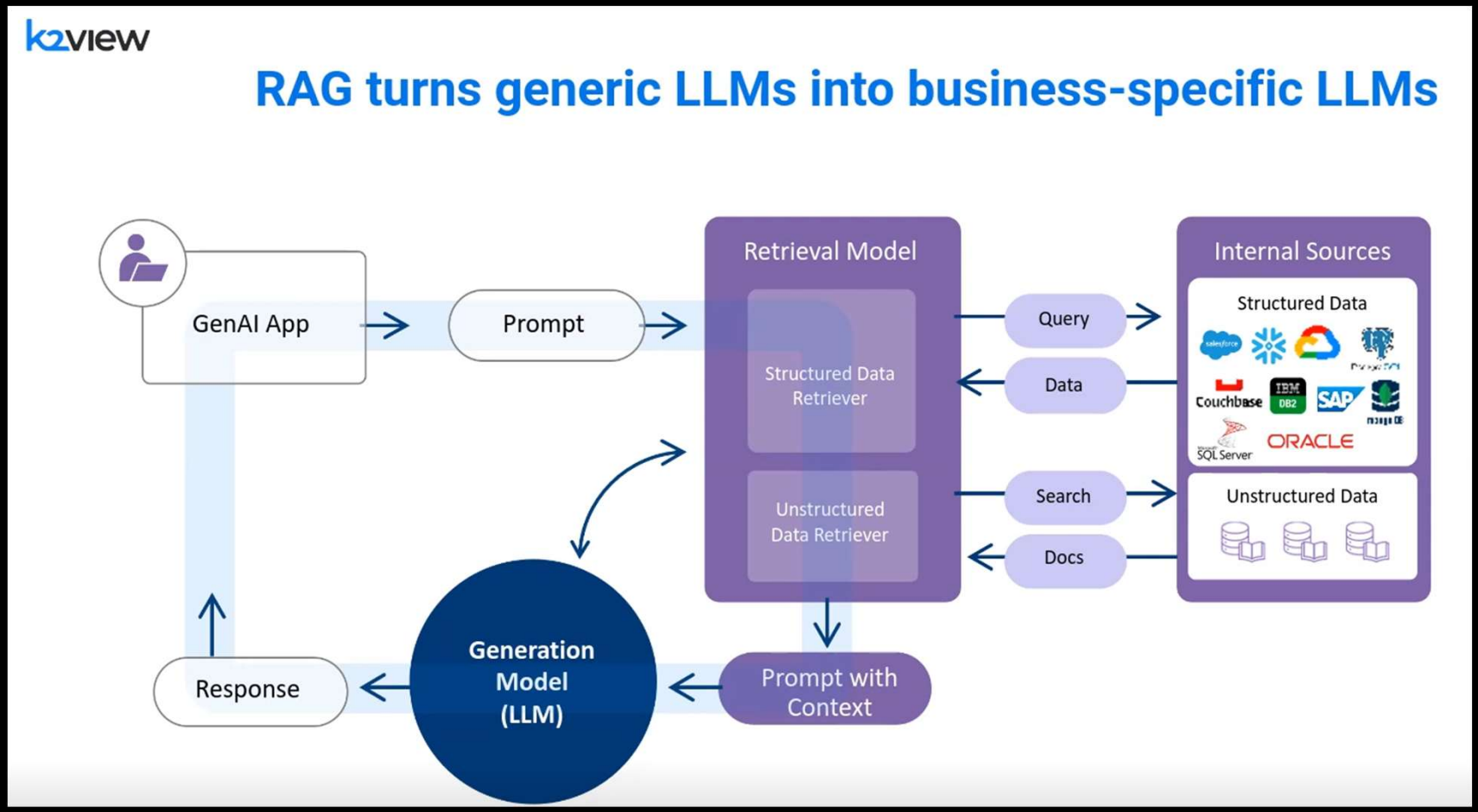
Access data in a data product platform

1. Map operational systems to a business entity
2. Configure GenAI data agent to dynamically query the entity and augment the LLM

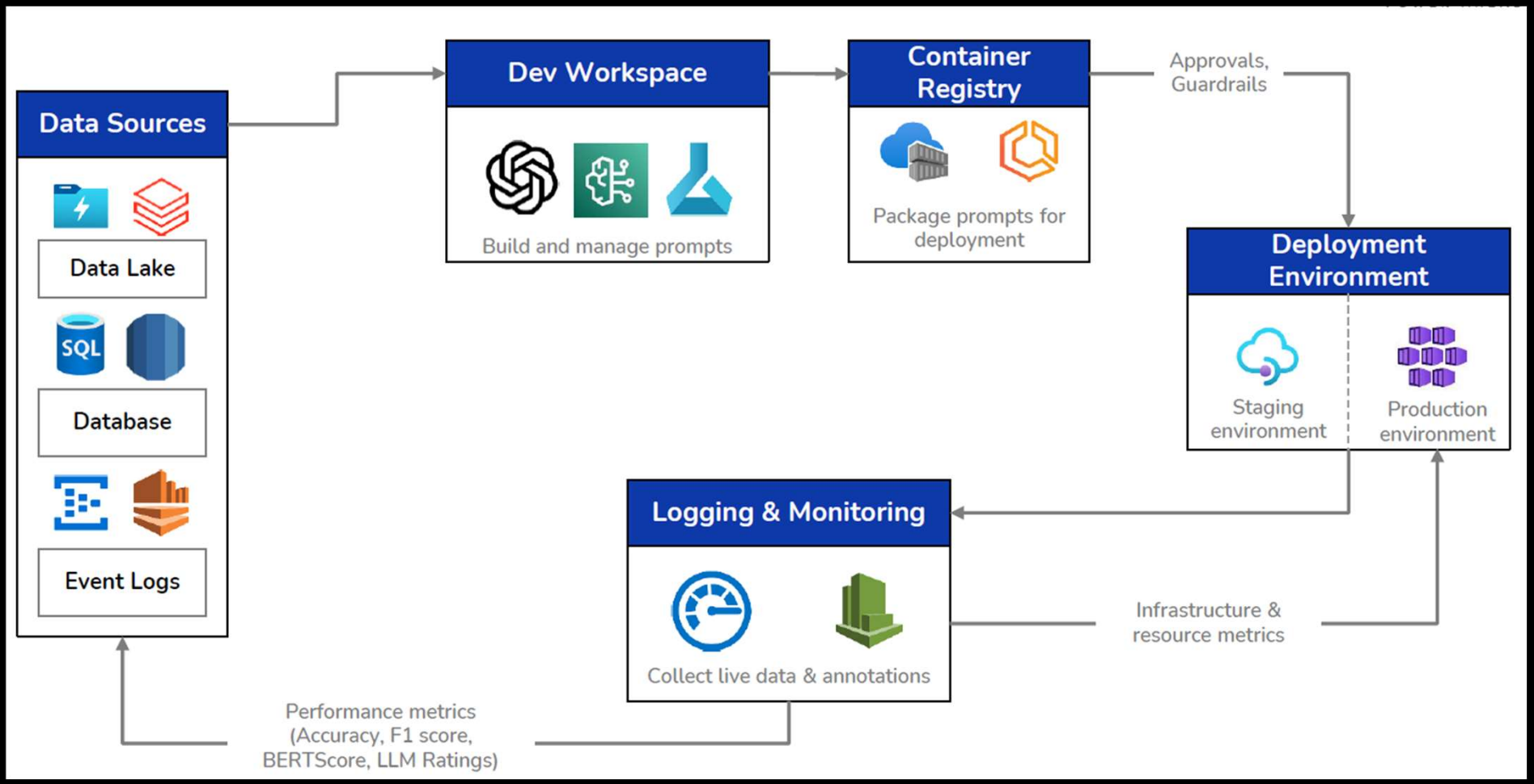
Result: GenAI-ready data

- **Security and privacy**
  Isolated, masked, and encrypted, per customer
- **Interactive**
  Conversational latency of queries at <100ms
- **Right-now-data**
  Complete, contextual data, always fresh and relevant
- **Controlled cost**
  Storage, query, LLM tokens, building agents functions
- **Scale and resilience**
  High query concurrency, huge data volumes, HA/DR
- **Reliability**
  Grounded, trusted, and personalized answers

Data Architecture to enable for RAG+Structured



Enterprise Gen AI solution workflow



Cost factors

Enterprise Gen AI Bill Of Materials

LLM

- Open source or closed source?
- Pre-trained LLM or customized?
- As-an API service from cloud?
- As managed service?
- deploy and manage on our own
  - on cloud?
  - on-prem?

Customizations

- RAG
- Model Fine Tuning (PEFT/LoRA/QLoRA, DPO, PPO, RLHF)
- Agents

Model selection and eval

- Use case : Text generation, Code generation, image generation(Media, marketing, design), voice synthesis, embeddings, etc.
- Evaluate data- What type of datasets your use case requires? (General purpose or Domain specific)?
- Performance - Quality of the response and supported latency
- Context window size
- Fine tuning & customization support
- Required Modality support- Single, multiple
- Type of model- General purpose model (Pre trained model), instruction tuned for your domain specific tasks & RL tuned models
- Hosting type - Self hosted or fully managed with model as a service
- Training Data – Type of data used to train the model- internet data, code
- License type- Open source, Open model or Proprietary
- Licensing conditions
- Data Privacy
- Ethical & Responsible AI considerations
- Language support – Most models are trained on English
- Cost- Infrastructure, software requirement to host the model
- Pricing- Hosted models are typically priced based on input tokens and completions.

https://medium.com/@gopikwork/comprehensive-guide-for-model-selection-and-evaluation-fcd7fe299a50

Considerations for using LLMs as API service Vs Hosted Vs on-prem

- Data privacy and security
- Time-to-market
- Usage (Application characteristics)
- Cost
- Skills
- Performance (Speed and accuracy/precision)
- Intellectual property ownership
- Available data / volume of proprietary or RAG data

Costing with SAAS such as Azure OpenAI/OpenAI

Let’s say we were to build a model that needs to be trained on all financial reports of all publicly traded companies in US.

mean annual reports are 55,000 words (~ 75K tokens).
Approx $0.12 to summarize each annual report
There are 58200 annual reports of publicly traded companies as per AAA
So approx. $6730 total cost
If you want to add quarterly reports, let’s say $5k for them
Add earnings call transcript summarization – roughly 10k words - $1250 for sentiment analysis

So, for $14K we are able to summarize all financial reports without building our own model

**Numbers are for reference only*

Costing with SAAS such as Azure OpenAI / Open AI

Task	GPT 3.5-turbo	LLaMA 2
55,000 words summary of a public company annual report	$ 0.12	$ 0.03
58,200 public companies annual reports summary	$ 6,729.38	$ 1,872.59
3 quarterly report summary	$ 5,047.03	$ 1,404.44
10,000 words transcription summary for all 58,200 companies	$ 1,236.75	$ 343.31
10,000 words call transcription sentiment analysis for all 58,200 companies	$ 1,236.75	$ 343.31
Total approximate cost	**$ 14,250.02**	**$ 3,963.67**

Numbers are only for reference and approximate/dated

RAG Solution Costing

1) the cost of creating the embeddings via the embedding model,
2) the storage cost for the vector database, and
3) the cost of an LLM for inference

Assuming that we are still using the SaaS/API approach and purchasing that service):

$0.50 to generate embeddings (using OpenAI's ada-2 model)
$120/month to store them in a vector DB (like Pinecone).

Based on the number of queries to the LLM, we still need to account for LLM usage costs. Let's stick to our earlier case where we were spending $7,000 on prompt workloads, and even with the RAG approach, we will generate an equal number of prompts.

So, summing it up, for approximately $8,500/year, we are able to apply our own dataset and ask questions with higher accuracy.

Even if we update our data nightly, requiring the recreation of embeddings, it will add another ~$180 to the bill.

Fine Tuning Costing

It takes 48 GPU (A100 - 80G) hours to fine-tune Mixtral-8x7B model with approximately 5M tokens.

A100 from Lambda Labs (one of the cheapest currently) is at $1.79/hr.

For 48 GPU hours, it will cost us ~$86.

So it will cost us **~$86** to fine-tune a model with 5M tokens (our yearly report example in earlier scenario also has 5M tokens).

We will need to host it separately – so add hosting costs

While MoE models can do inference very fast, it requires large amount of VRAM. In case of Mixtral-8x7B requires 90GB of VRAM in half-precision - so we will need two A100-80G GPUs to support this for inference.

If we do simple math, at the same GPU price, it will cost us $31,360/year to do the inference.

**Numbers are approximate only*

https://huggingface.co/blog/mixtral