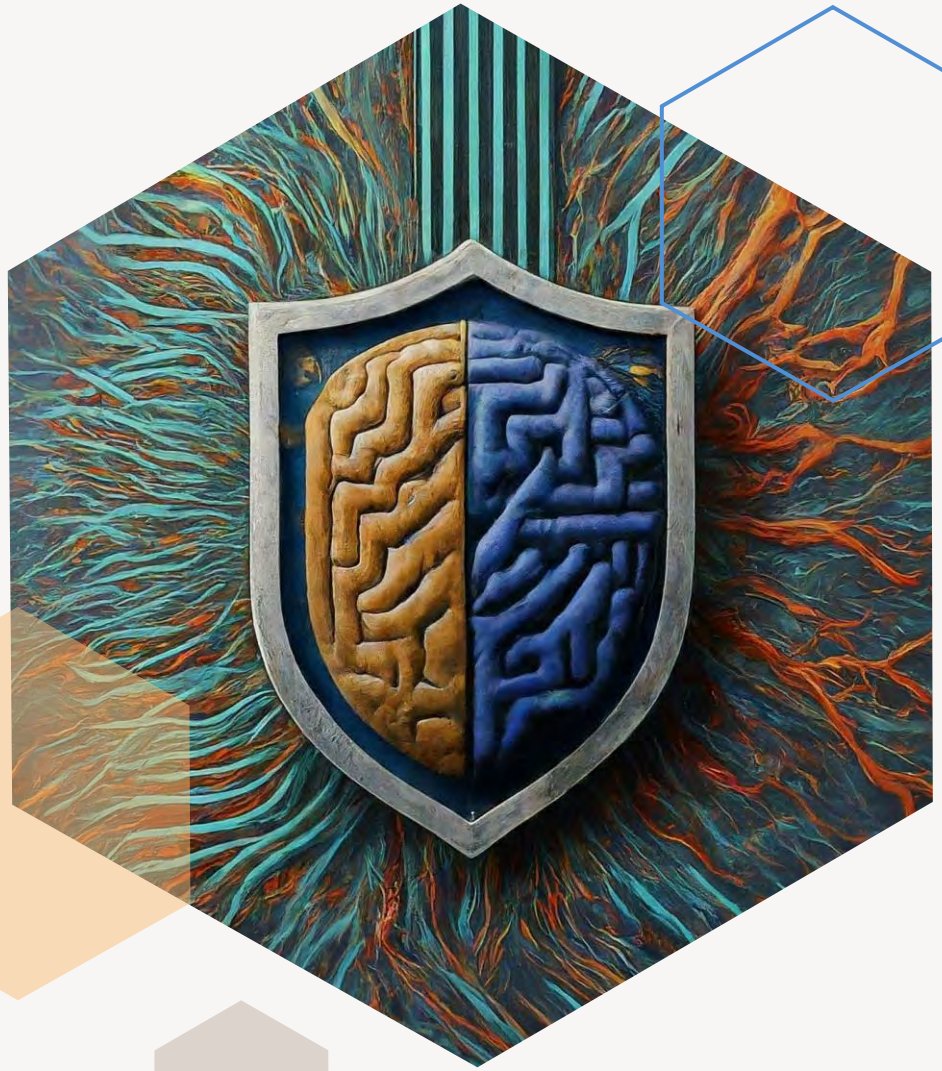
AUXIN

Large Language Model (LLM) SecOps

Secure GenAI Applications



Omer Farooq

August 2024

CTO and Security Engineer

Auxin Security – Auxin.io

About me

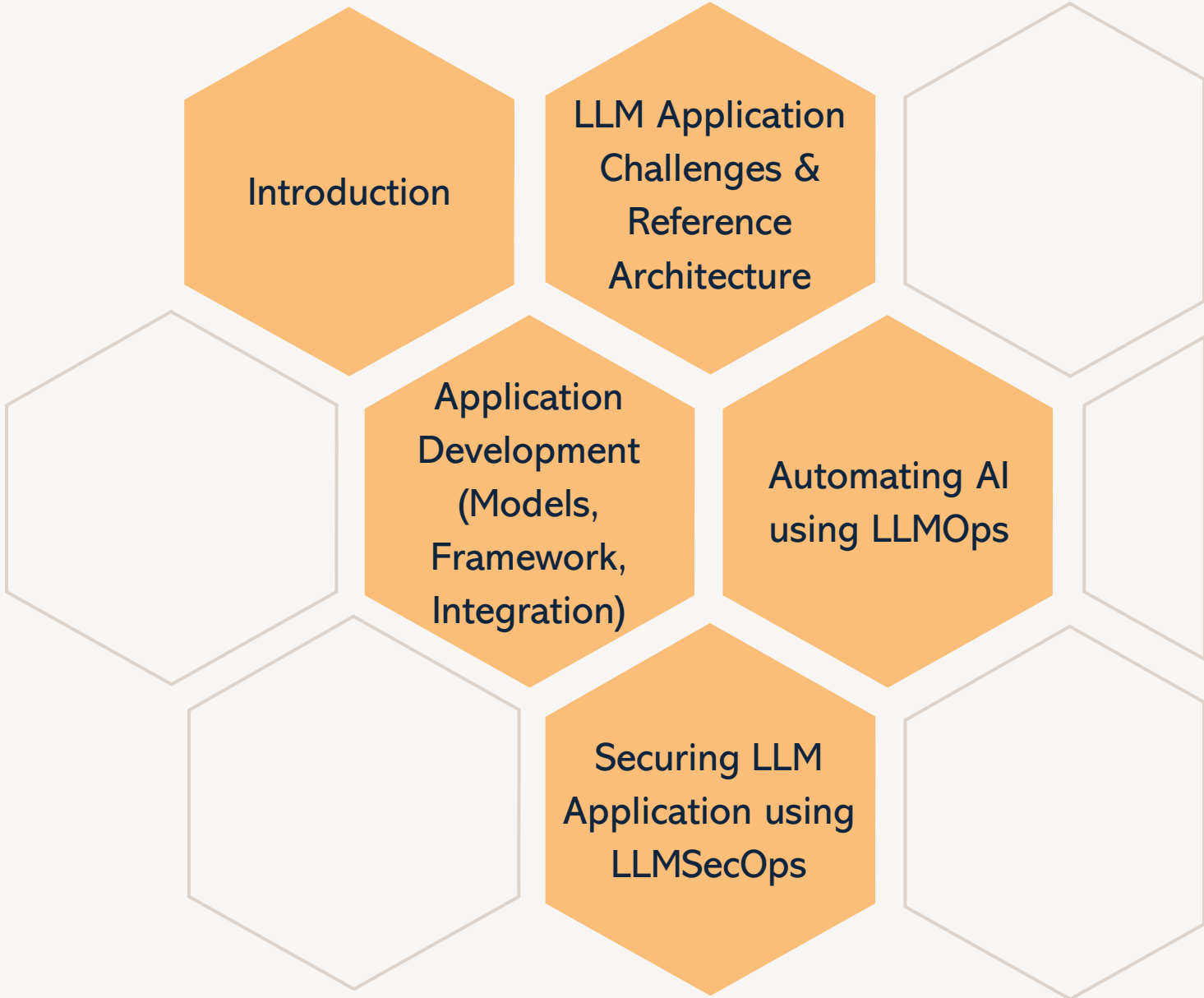
Omer Farooq

Founder and Security Engineer

Auxin Security

ofarooq@auxin.io

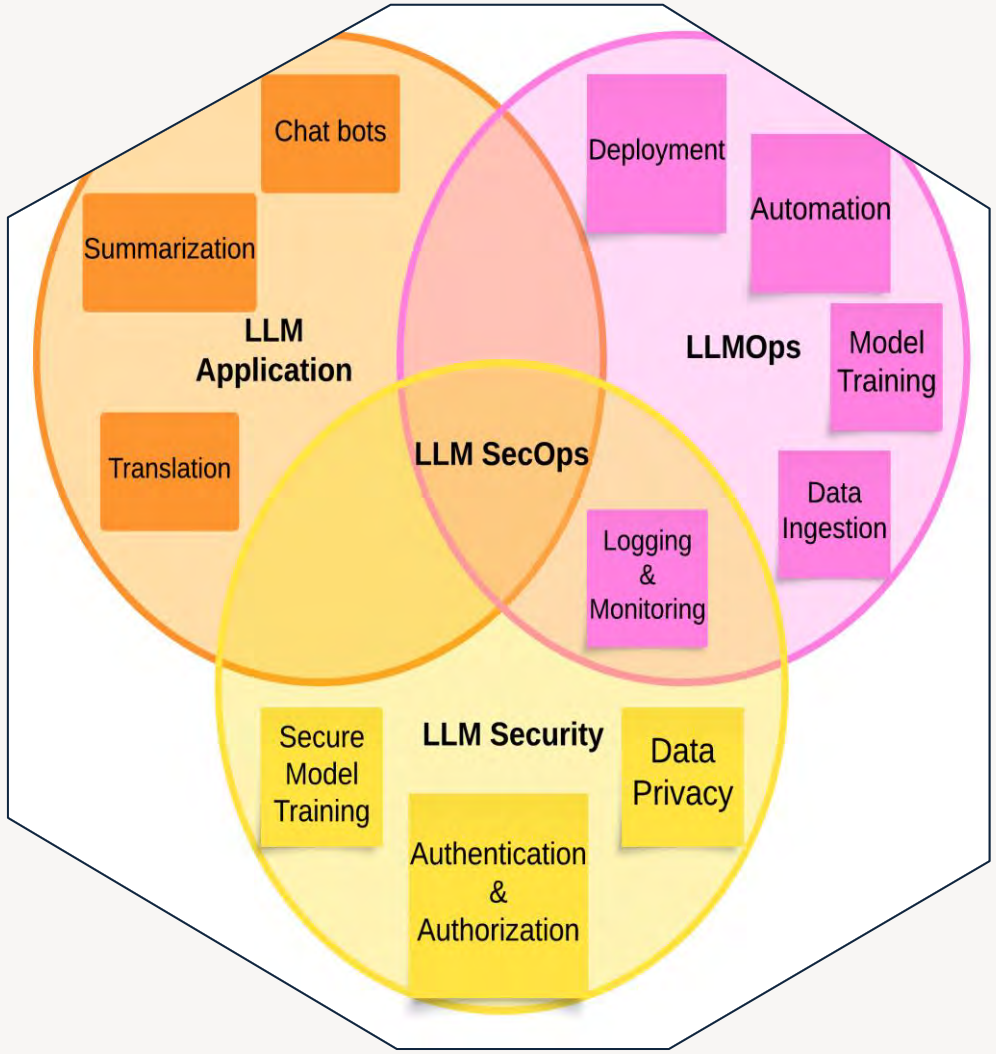
Agenda


Introduction
LLM Application Challenges & Reference Architecture
Application Development (Models, Framework, Integration)
Automating AI using LLMOps
Securing LLM Application using LLMSecOps

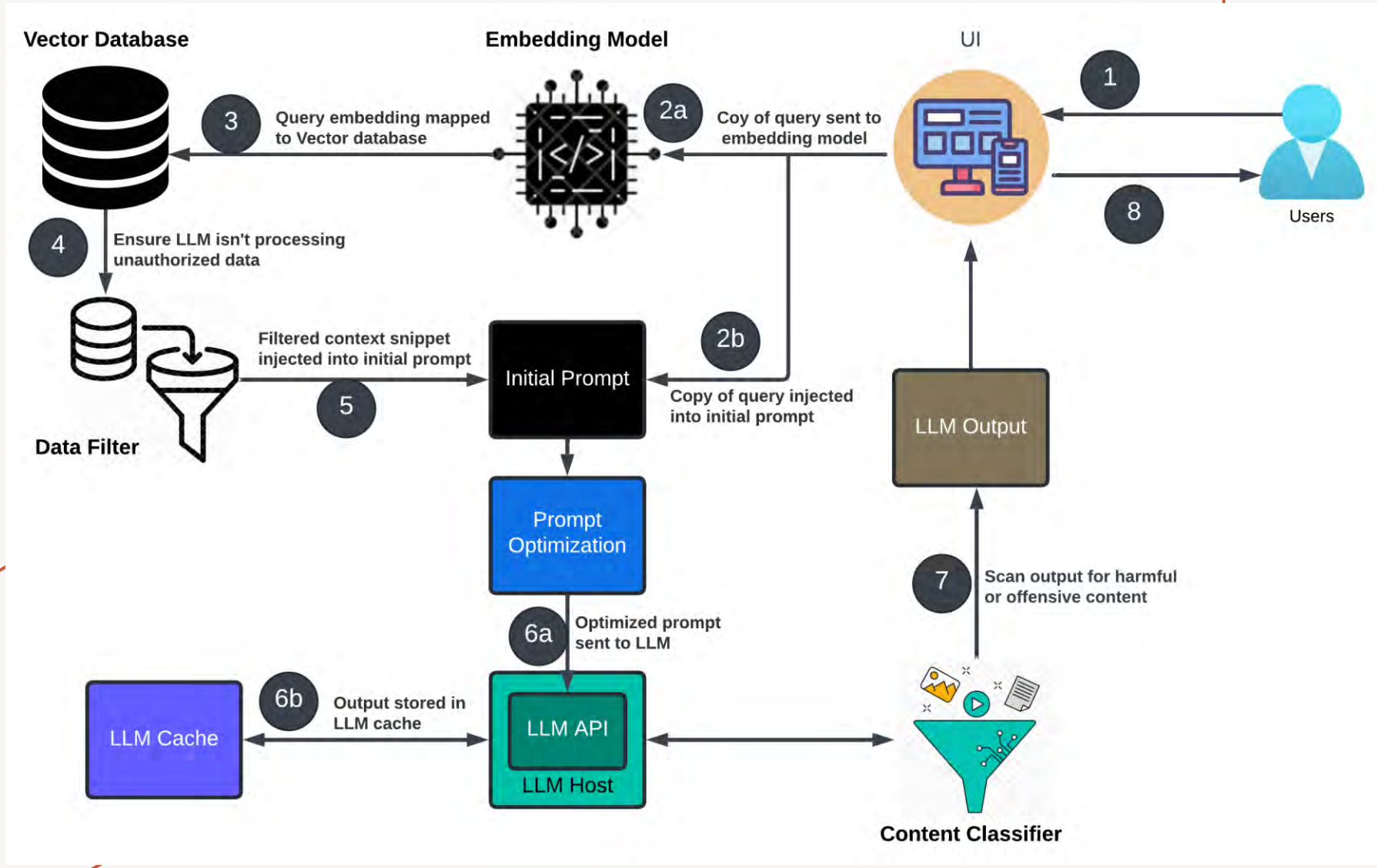
What is LLM SecOps?

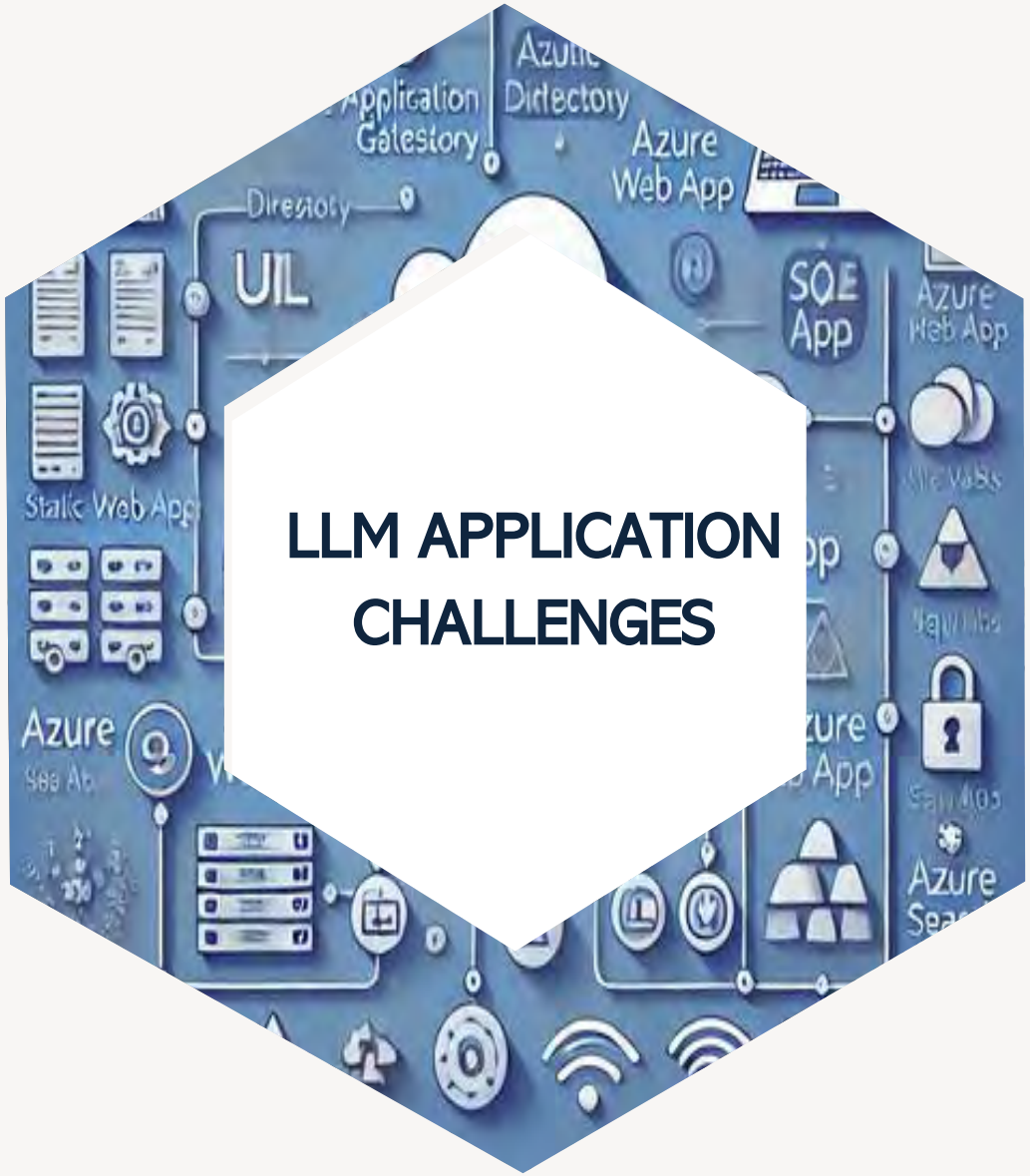
LLMSecOps = LLM Application + LLMOps + LLM Security

- Large Language Model (LLM) Applications are **Data Routers**
- LLMOps includes
  - Automation
  - Deployment
  - Model Training
  - Data Ingestion
  - Monitoring
- LLM security focuses on protecting LLMs from vulnerabilities and threats



LLM Application Reference Architecture





- **Validate all Integration points for Security**
- **Data Integration**
  - Authentication – Control data ingestion and employ OAuth flows to create connections
  - Authorization – Need to control data ingestion and exposure points tightly
  - Service Account and Secret Management
- **Data Lifecycles and DLP**
  - Based on the Governance policies needed to ensure Applications manage Data protection
  - Use data cataloging and DLP solution with an automatic classifier to detect data exposure
- **AI Service API Management**
  - Enable API Management and Security on all LLM-based products for security and monitoring
  - Model, Token usage, and Metered usage should be performed outside of the Service level

LLM Application Development – Where to Start?

Choosing the right LLM Model is vital

- Understand the quality and limitations of the model
- Model results can vary significantly
- Multi-Model – Text/Image
- Model cost varies

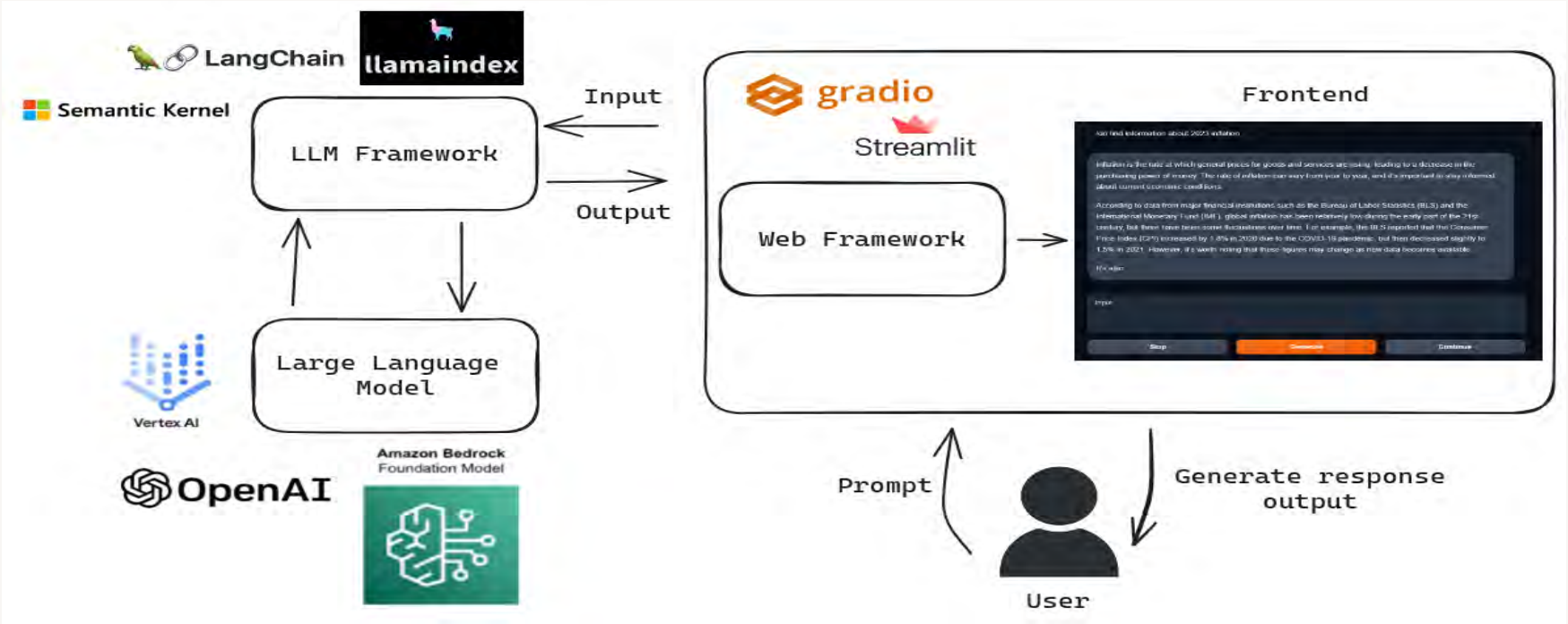
Develop Application and LLM Separately

- Integrate LLM as a Service in your Application
- Keep all Data in separate Storage

Use Application and Role Based authentication, authorization to protect Data

Rapid Prototype & Fast Result

- **Use Existing Data Router Frameworks**
  - Langchain, Semantic Kernel, LlamaIndex
- **Low Code Frontend**
  - Streamlit, Gradio
- **Leverage Cloud Services**
  - Serverless and Managed Service
- **Use purpose build Cloud Native Storage**
  - Vector, caching, SQL, NoSQL



LLM Development Principles

- **Decoupled Architecture** - Keep the application and LLM service separate
  - LLM requires high computation and will cost more

- **Applications should be Cloud Native -** Ensure Scalability, Agility, & Resiliency
  - Use a framework, save time, and increase productivity

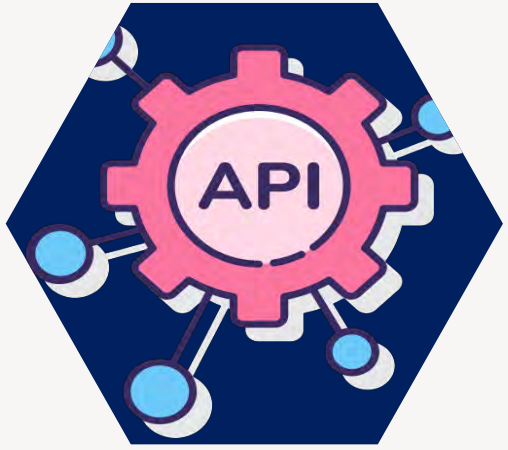
- **Vector database –** Embedded Storage and Performance is critical
  - Test often Models do not provide consistent results

LLMOps

LLMOPs is an extension of existing DevOps but fine grain control designed for LLM Application

Automation & Deployment

- Automate deployment CI/CD
- Automate LLM pipeline
- All software containers or serverless



Integration with Application

- Inject secrets at the time of deployment
- Ensure all integration authentication tokens are rotated

Monitoring & Observability

- Index for LLM Apps
- LLM Applications are Data Routers thus, logging and monitoring is essential

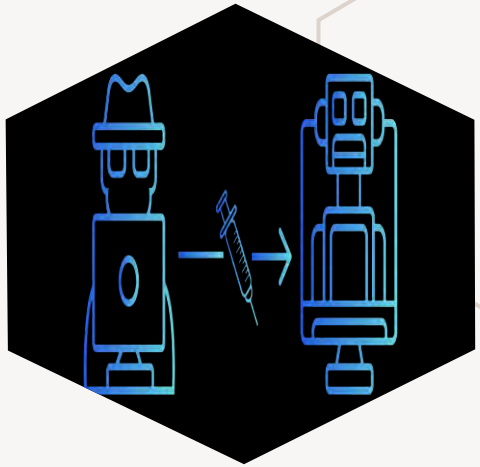
Data Management

- LLM Application requires blob storage, SLQ, NO-SQL, Vector, Cache, and memory storage
- Ensure data quality and consistency

Strategic LLM SecOps

LLMSecOps is the automation of Security practices for LLM based applications and deployment process

LLM Application Security Concerns



Prompt Injection

Model Poisoning

Adversarial Attacks

Data Leakage

Ethical Concerns

Tactical LLM SecOps - Ensuring AI Security

Reuse DevSecOp Automation SAST, DAST, Cloud Scans with a focus on Chains

Adhere to secure coding practices including LLM data routing

Implement content filtering to prevent generating offensive or inappropriate content

Log AI Service logs in correlation with Cloud Compute, storage, and network logs

Review code using SAST and SCA tools for vulnerabilities and prioritize their remediations

AUXIN

Thank you

Auxin Security

www.auxin.io