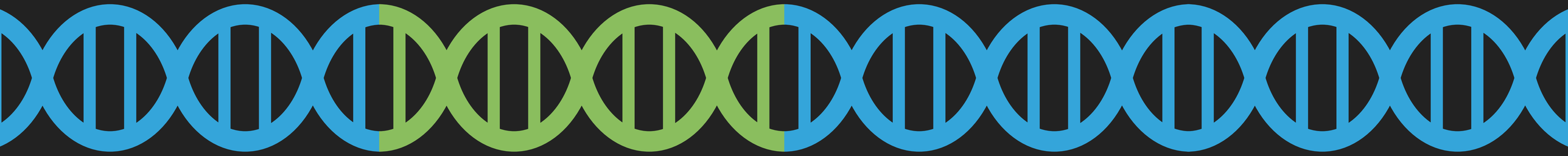


GENAI RAG PROJECT: TOPIC STUDY ASSISTANT [POC]

GENE EXPRESSION ACTIVATION VIA MICRO RNA

USER-FOCUS AND TOPIC CHOICE

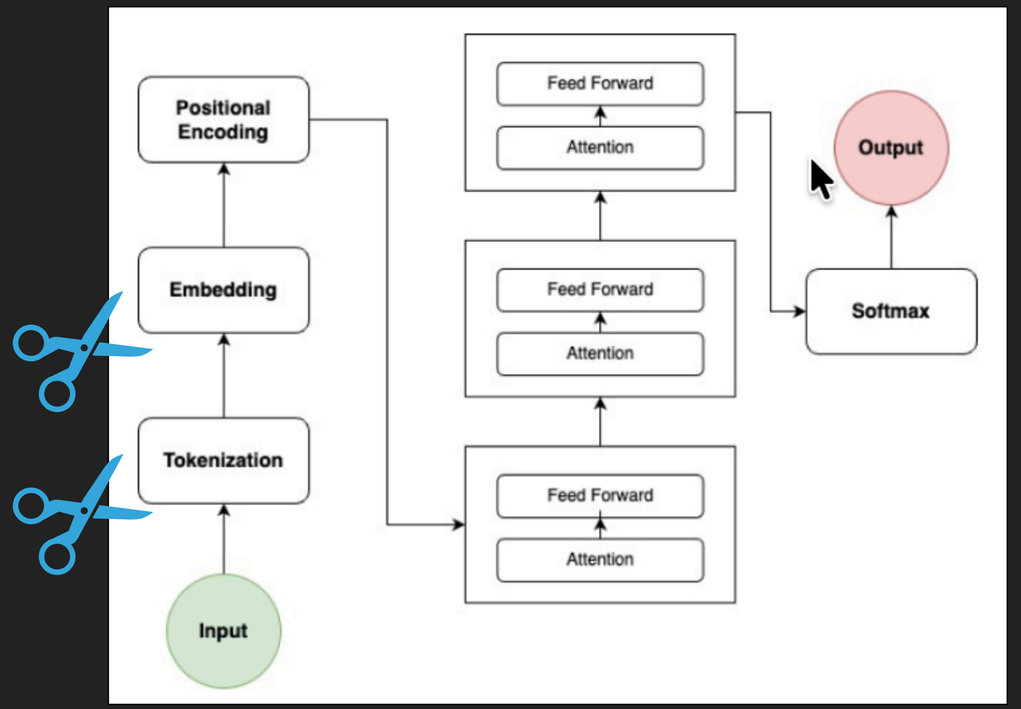
- Developer's goal:
  - Learn and practice implementing cutting-edge technologies.
  - Create a **pet-project** for portfolio and upscale.
  - Practice user-oriented project design.
- User's goal:
  - Get assistance for writing **Master's thesis** in Biology on the topic:

    "GENE EXPRESSION ACTIVATION VIA MICRORNA".
  - Deepen knowledge in the research topic.
  - Start collaborating with data scientists for further steps in Bioinformatics.

TERM DISAMBIGUATION

- **Generative AI** - AI that generates new data by learning patterns from existing data.
- **GPT** - Generative Pre-trained Transformer
- **LLM** - Large Language Model - generates output.
- **RAG** - Retrieval Augmented Generation
- **Embedding model** - converts data chunks into a numeric vector.

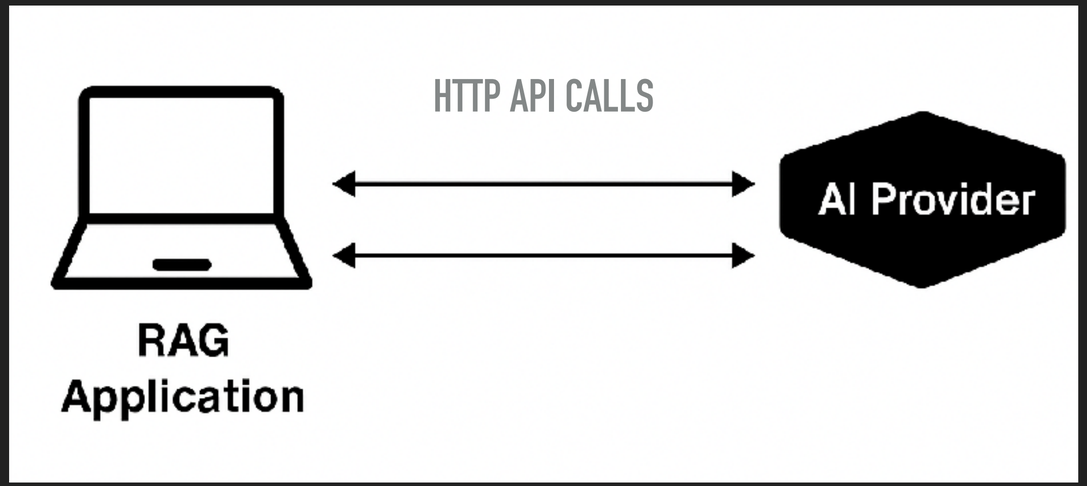
TRANSFORMER / GPT WORKFLOW



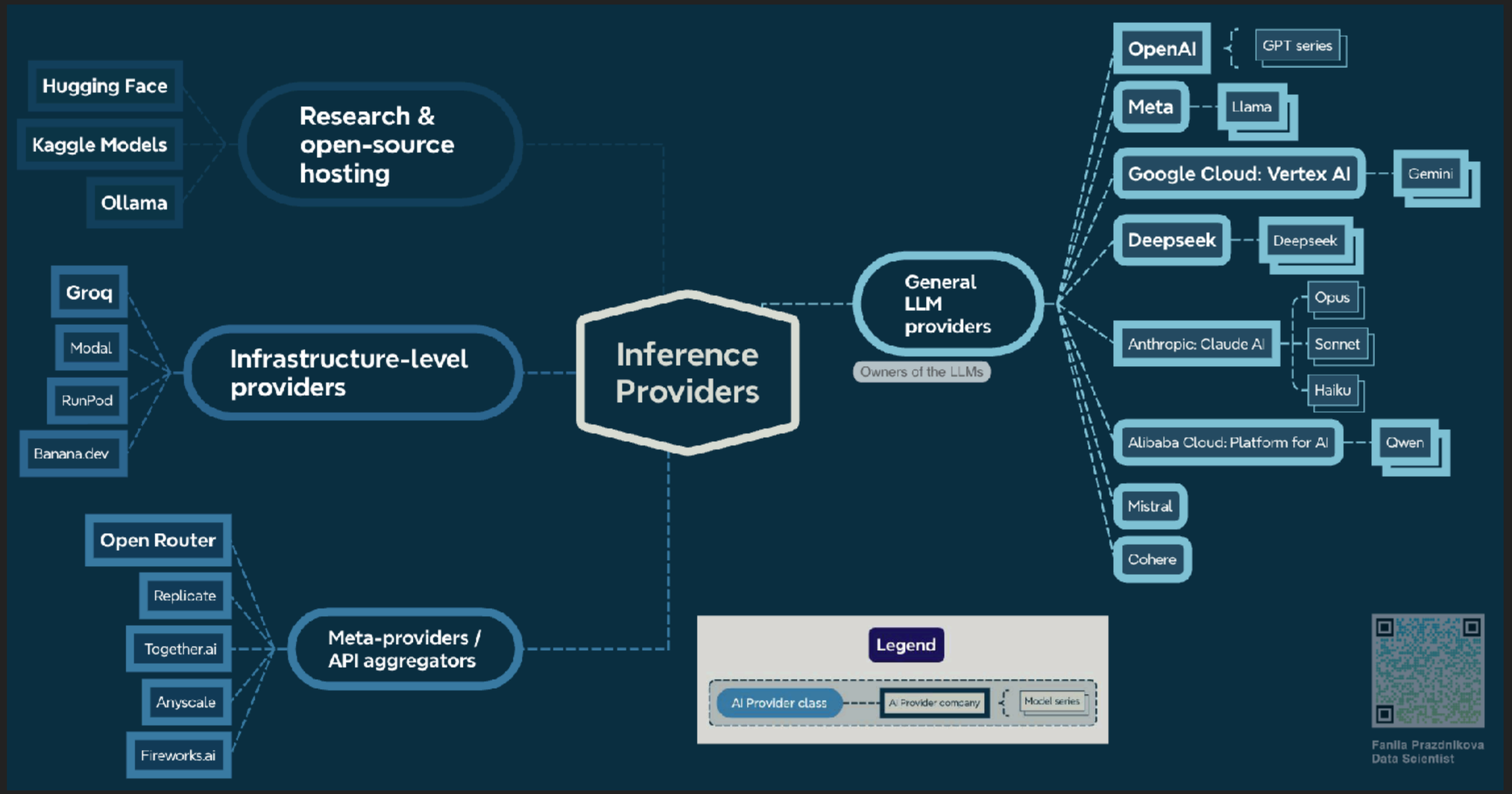
WHAT ARE INFERENCE PROVIDERS

- Platforms that host, run, and expose AI models via APIs.
- Handle the inference - the actual computation that generates answers, embeddings, images, etc.
- Provide:
  - model access (LLMs, embedding models, etc.)
  - APIs / SDKs for querying these models
  - Compute infrastructure (GPU clusters for inference)
  - Model hosting for custom deployments
  - Monitoring, billing, and authentication

SYSTEM ARCHITECTURE



CHOOSING INFERENCE PROVIDER



RAG PIPELINE

1	Data collection	▸ Data types: PDF documents with text, images, formulas ▸ POC: data is collected manually by the user ▸ MVP: web-scraping or GPT-5 for finding relevant scientific articles
2	Data retrieval and tokenization	▸ Extract data from PDF files ▸ Preserve text, images, scientific formulas ▸ Tokenization: split into chunks
3	Embedding and indexing	▸ Embedding model: convert data to vectors ▸ Vector DB: store the vectors in DB and build semantic index
4	Data retrieval for answer	▸ Using GUI for getting query from a user ▸ Query tokenize/encode/embedding using the same model and methods ▸ MVP: top-k similarity search, reranking
5	Generating answer	▸ Prompt LLM with the retrieved context ▸ Return the answer to the user in GUI

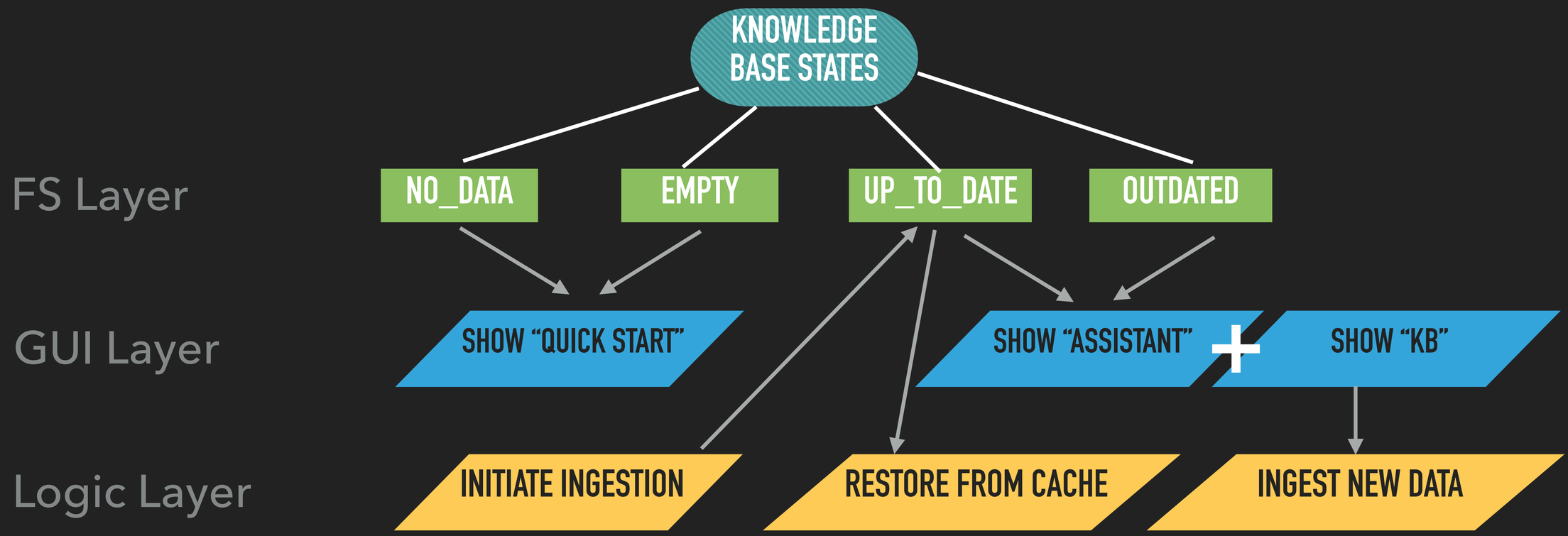
CHOOSING AI FRAMEWORK

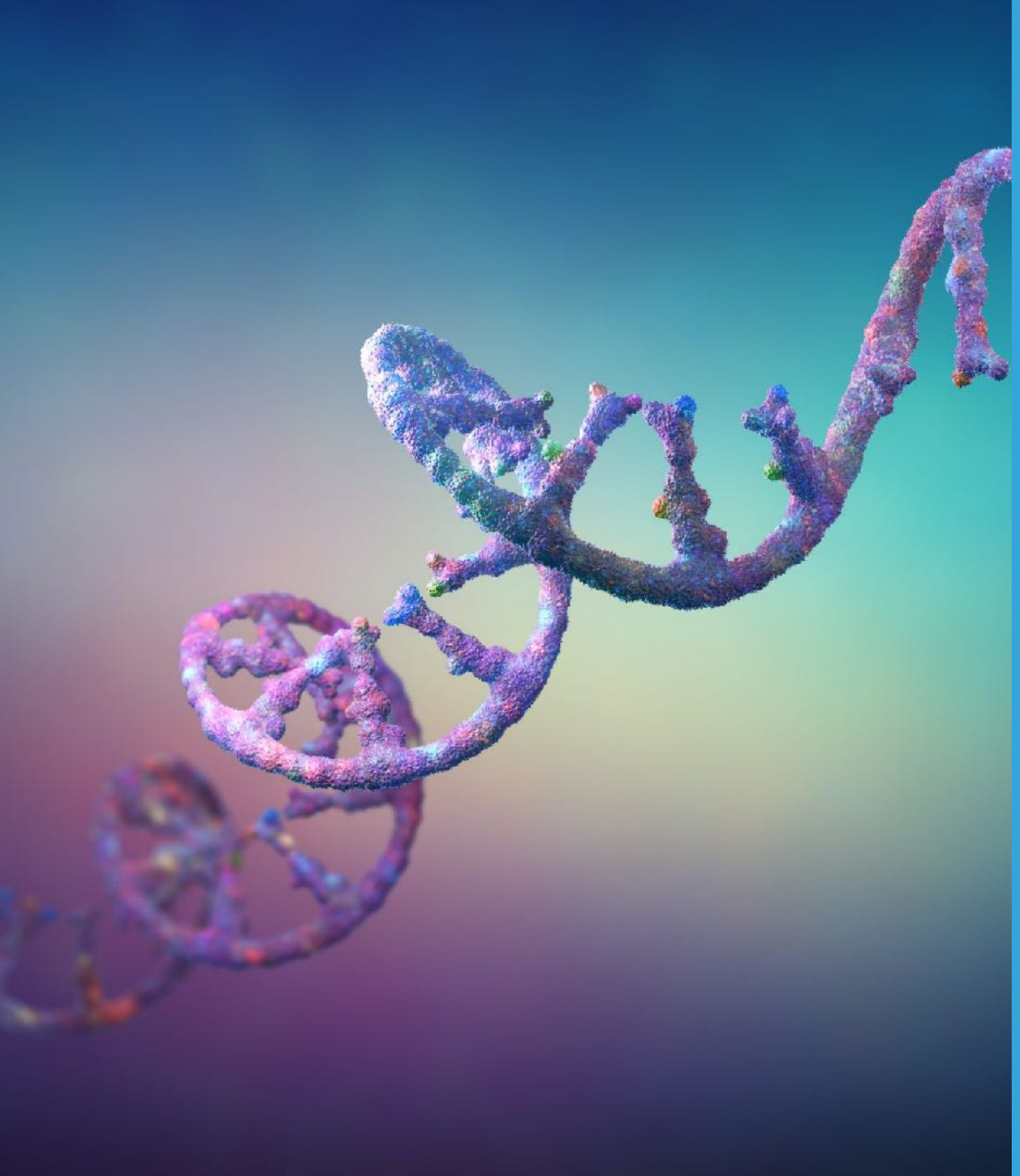
LangChain	LlamaIndex
tool- and workflow-centric	data-centric
for various LLM projects	specific for RAG projects
more divers functional modules for architecture flexibility and scalability	more efficient in document indexing, data retrieval, semantic search
open-source and free	**commercial**, price depends on usage
	migration to it can be considered after MVP and efficiency evaluation + donations / monetization prospects

RAG TOOLCHAIN FOR POC PHASE

User Interaction:	Type: Web site ‣ Back-End: none for POC ‣ Front-End: StreamLit with caching on localhost
AI framework:	LangChain
PDF extractor:	PyPDFLoader
Vector DB:	Chroma DB on localhost
Chunker:	RecursiveCharacterTextSplitter
Retriever:	Chroma.as_retriever
Embedding model:	text-embedding-3-small
LLM:	gpt-5-mini (primary), gpt-5-nano (for drafts/tests)
Inference Providers:	‣ Embedding model: OpenAI ‣ LLM: OpenAI

STATE MACHINE





THANK YOU FOR ATTENTION

Faniia Prazdnikova