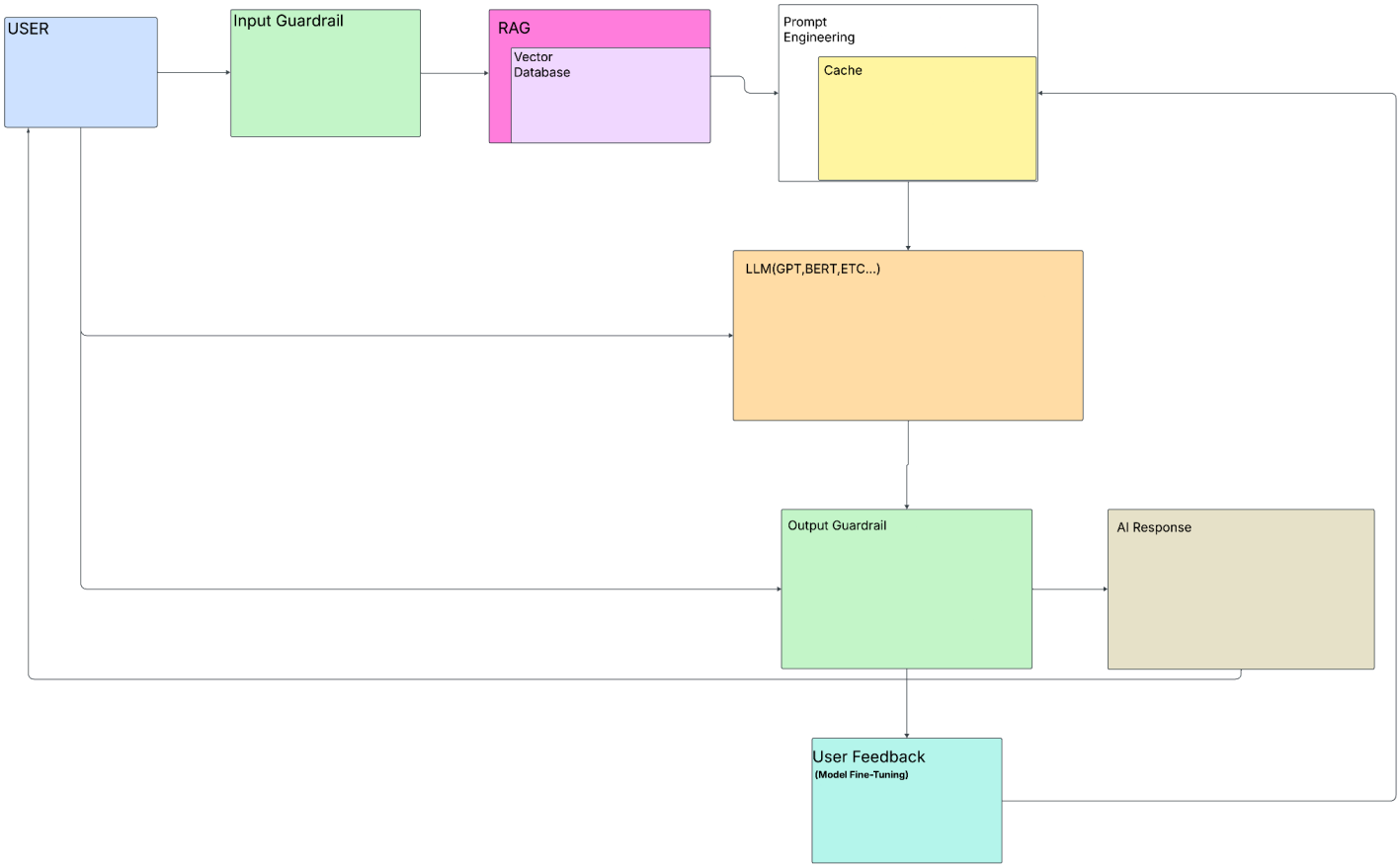
Understanding Generative AI (GenAI) Architecture

Generative AI (GenAI) systems follow a structured architecture that allows them to process inputs, generate meaningful outputs, and refine responses for accuracy and coherence. Your goal is to create a more intuitive and easily understandable representation of GenAI architecture.

Key Components of GenAI Architecture

1. **User Input (Query)**
    - The system starts with an input from the user, such as a prompt, question, or instruction.
2. **Preprocessing & Guardrails**
    - Input Guardrails ensure safe, structured, and meaningful input processing.
    - Guardrails include content filtering, formatting, and rule-based modifications.
3. **Retrieval-Augmented Generation (RAG)**
    - If external knowledge is needed, the system queries a **Vector Database** to fetch relevant documents.
    - It combines retrieved knowledge with the input query for better contextual understanding.
4. **Prompt Engineering & Caching**
    - The input is transformed into an optimized prompt format using **Prompt Engineering**.
    - Caching reduces computational overhead by reusing previous responses when similar queries appear.
5. **Large Language Model (LLM)**
    - The heart of GenAI, where the model processes the input and generates text-based responses.
    - Uses Transformer-based architecture like **GPT, BERT, or LLaMA**.
6. **Postprocessing & Output Guardrails**
    - Generated content is filtered for bias, correctness, and coherence.
    - Guardrails ensure compliance with ethical AI guidelines.
7. **Response Generation**
    - The final output is structured and presented to the user.
8. **Feedback & Continuous Learning**
    - User interactions are logged for fine-tuning the model.
    - Reinforcement learning techniques like RLHF (Reinforcement Learning with Human Feedback) improve responses.

[Lucid Link]


USER
Input Guardrail
RAG
Vector Database
Prompt Engineering
Cache
LLM(GPT,BERT,ETC...)
Output Guardrail
AI Response
User Feedback
(Model Fine-Tuning)