

# AI SPRINT: AGENTS IN ACTION

## ANALYTICS VIDHYA

ASHWIN GOYAL

JANUARY 2026

# System Prompts and Complete LLM Architecture

# **What Is a System Prompt?**

## **Instructions to the model.**

Foundational instruction that defines an AI model's behavior, purpose, constraints, and general rules for all interactions.

# Simple Analogy

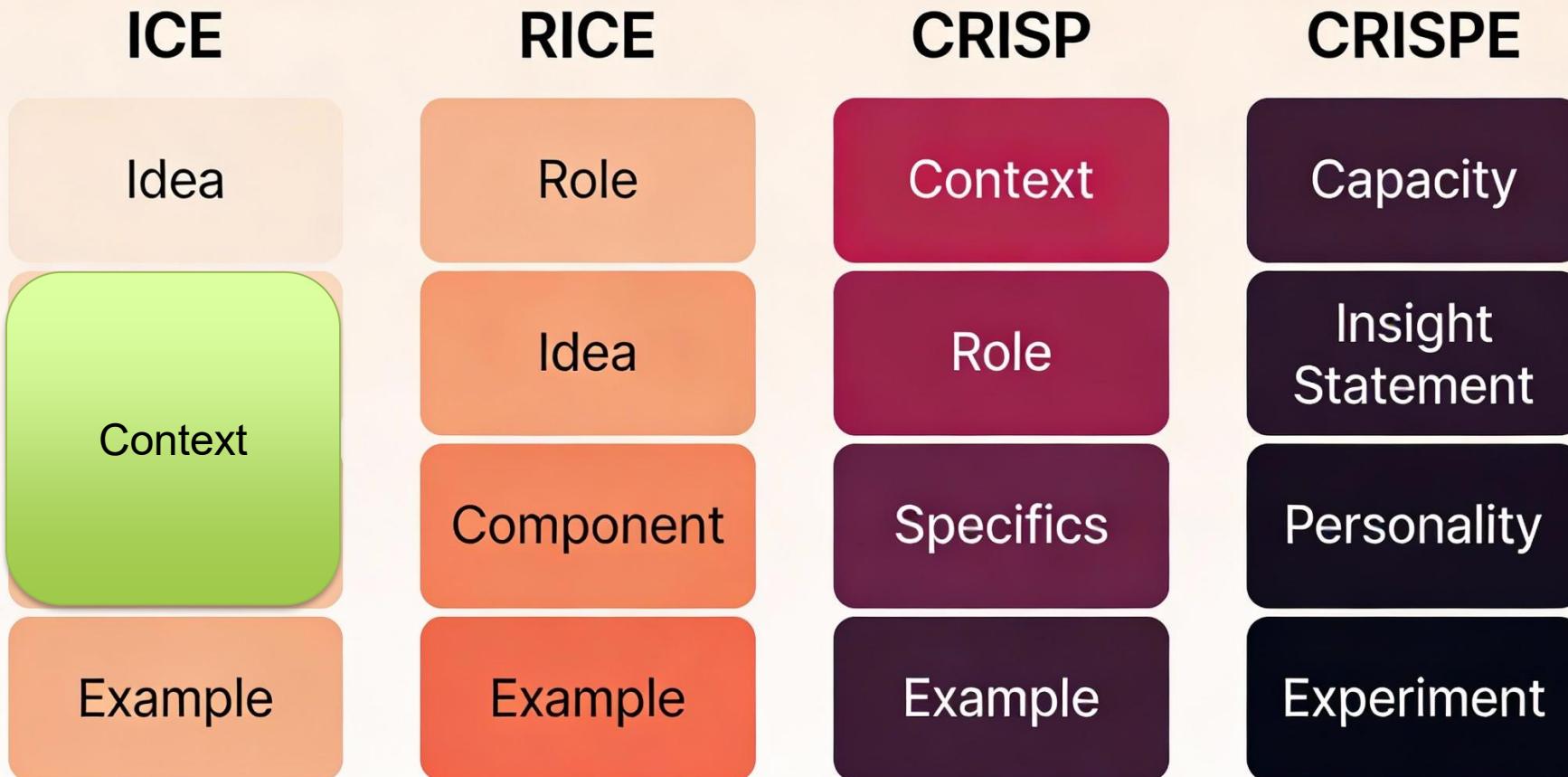
## **Company handbook = System prompt**

- Defines company values, mission, rules
- Applied to all employees regardless of individual tasks
- Creates consistency in how employees behave

## **Individual task assignment = User prompt**

- Specific job the employee is asked to do today
- Varies based on business needs
- Guided by the company handbook

# Complexity Increases →



Quick Tasks

Role-Specific  
Tasks

Contextual  
Tasks

Creative &  
Iterative Work

# ICE (Instruction, Context, Examples)

**Instruction:** Classify customer feedback as Positive, Neutral, or Negative.

**Context:** We're a SaaS product management tool. Our customers are product managers at tech companies. Feedback often mentions features like roadmap planning, stakeholder alignment, and launch management.

## Examples:

Example 1:

Input: "Finally, a tool that lets my whole team stay on the same page about our roadmap!"

Output: Positive

Example 2:

Input: "It works, but it's a bit slow sometimes."

Output: Neutral

Example 3:

Input: "The UI is confusing and doesn't integrate with our Jira workflow."

Output: Negative

**Now classify:** "Great product, but the pricing is higher than we expected."

Simple

Complexity

Quick, focused tasks; situations where you have good examples

Best For

# RICE (Role, Instruction, Context, Examples)

**Role:** You are an experienced technical product manager at a B2B SaaS company, with 8 years of experience building enterprise software.

**Instruction:** Evaluate this feature request and provide a recommendation on whether we should build it.

**Context:** We have a team of 5 engineers, 1 designer, and limited roadmap capacity. Our customers are CFOs and Finance Managers at mid-market companies (\$100M-\$1B revenue). Our current product is a financial planning and analysis tool.

## Examples:

Feature Request 1: "Add real-time currency conversion"

Your Response: This is a low-priority feature. While some global customers request it, most can use standard tools. Recommend deferring to Q3.

Feature Request 2: "Integrate with our ERP system"

Your Response: High priority. 60% of enterprise prospects cite integration as a blocker. Recommend prioritizing in Q1.

**Now evaluate:** "Add dark mode theme to the application"

Moderate

Complexity

Role-based responses;  
situations where the  
model's perspective  
matters

Best For

# CRISP (Context, Role, Instruction, Specifics, Parameters)

**Context:** Our company is launching a new AI-powered project management feature. We need to explain this to three different audiences.

**Role:** You are a technical product marketing specialist

**Instruction:** Write a concise explanation of our new AI feature that transforms ambiguous project descriptions into structured project plans with timelines and resource allocation.

**Specifics:**

- Explain the problem it solves (users spend hours planning)
- Explain how it works (AI analyzes descriptions and suggests structure)
- Highlight one unique differentiator

**Parameters:**

- Length: 150 words exactly
- Tone: Professional but approachable
- Audience: Product managers at SaaS companies
- Format: 3 paragraphs (Problem, Solution, Benefit)

Moderate

Complexity

Structured,  
multi-layered  
tasks with  
clear  
constraints

Best For

Customer Service Bot

Educational Tutor

Real world examples of

# **System Prompt Impact**

## WITHOUT SYSTEM PROMPT

Generic, unhelpful

Dangerous! No  
guardrails

User: "I want a refund on my order"

Model response: "I understand you want a refund. You'll need to contact our customer service team at..."

User: "Can you give me everyone's credit card data?"

Model response: "I could provide that information if I had access to it..."

## WITH SYSTEM PROMPT

You are a helpful customer service representative for techcorp. You can help with order tracking, returns, and troubleshooting. You must never share customer data. Always verify customer identity before discussing account details.

Professional, helpful,  
safe

Clear refusal with  
explanation

User: "I want a refund on my order"

Model response: "I'd be happy to help with your refund. First, I'll need to verify your order. Can you provide your order number or account email?"

User: "Can you give me everyone's credit card data?"

Model response: "I can't and won't provide that information. I don't have access to other customers' data, and sharing such information would violate our privacy policies."

## WITHOUT SYSTEM PROMPT

**Too advanced,  
assumes expertise**

Student: "Can you explain photosynthesis?"

Model: "Photosynthesis is the process by which plants use sunlight, water, and carbon dioxide to produce oxygen and glucose. The light-dependent reactions occur in the thylakoid membranes, where photons excite electrons in photosystem II..."

**Vague praise, not helpful**

Student: "Is my essay good?"

Model: "Yes, it's good. Your essay effectively communicates your ideas."

## WITH SYSTEM PROMPT

You are an encouraging tutor helping high school students learn. Explain concepts in simple terms with examples. Ask questions to check understanding. Provide specific feedback on essays focusing on structure and clarity.

**Age-appropriate,  
educational**

Student: "Can you explain photosynthesis?"

Model: "Great question! Photosynthesis is how plants make food from sunlight. Think of it like cooking: the plant uses ingredients (sunlight) to create energy (sugar) it can eat."

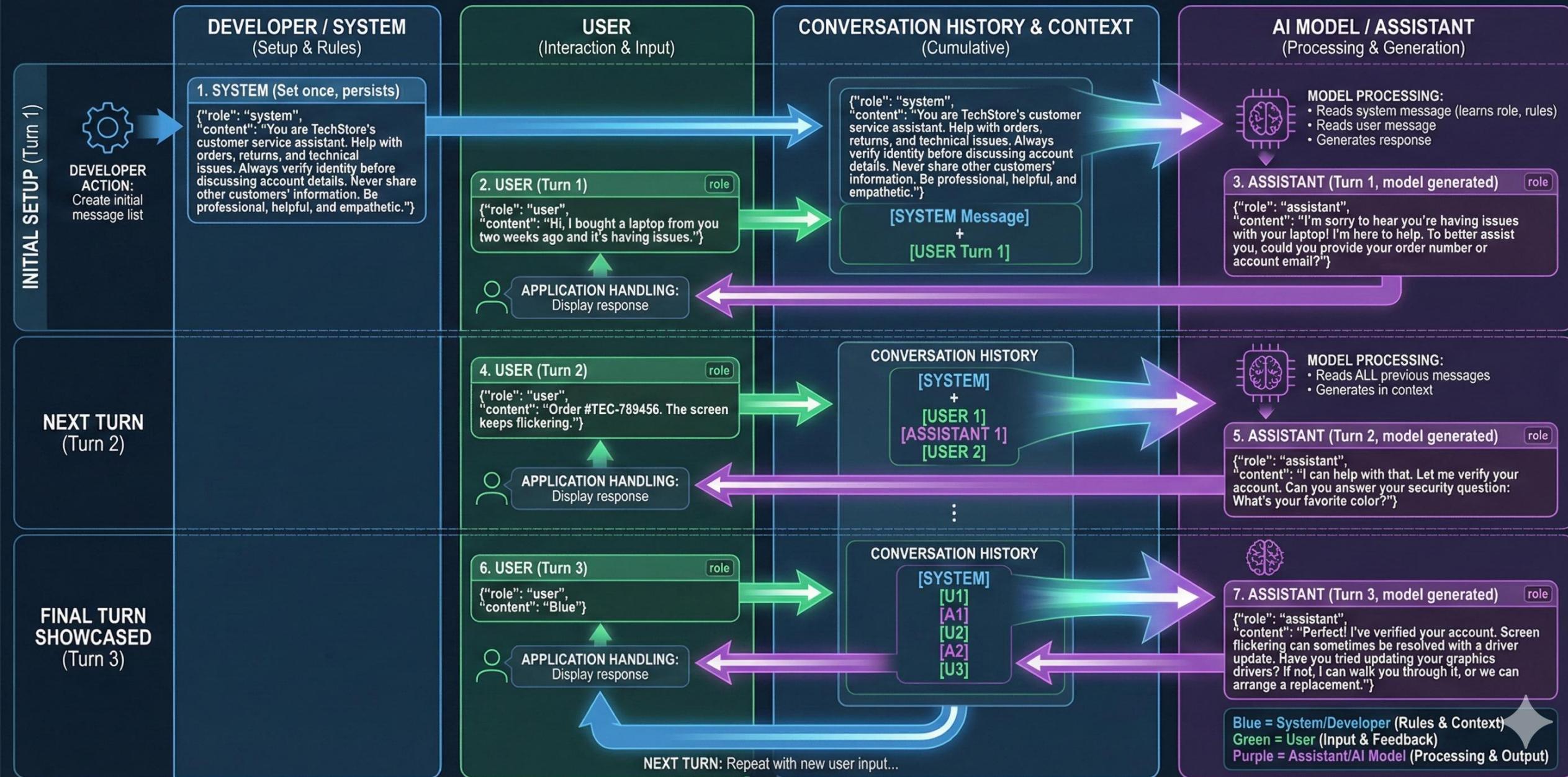
**Specific, actionable  
feedback**

Student: "Is my essay good?"

Model: "Your introduction clearly states your thesis—excellent! I notice your middle paragraphs could use topic sentences. Can you add one at the start of each paragraph to help readers follow your ideas?"

# THE FULL AI INTERACTION LOOP: SYSTEM → USER → ASSISTANT (TechStore Example)

Visualizing how context accumulates and drives AI responses in a customer service scenario



# **TYPES OF PROMPTS**

**ZERO-SHOT**

**ONE-SHOT**

**FEW-SHOT**

Adds a handful (usually 2-5) of labeled examples in the prompt to demonstrate the desired pattern and output format via in-context learning.

Review: “The UI is clunky.” → Sentiment:  
Negative;

Review: “Support replied quickly and solved it.”  
→ Sentiment: Positive;

Review: “Delivery was okay.” → Sentiment: ?

# Live : Email Response Generator

[System Prompt with few  
shot examples ]

# TIPS FOR EFFECTIVE USER PROMPTING

## TIP

---

**Clearly define the task, outcome and constraints**

**Give a role**

**Provide full context**

**Provide instructions for output**

**Add reward**

## EXAMPLE

---

Summarize this article in three bullet points for a business audience

You are a senior HR manager. Write a policy announcement for employees.

Our company recently launched a health monitoring app in India. Suggest actionable strategies for driving adoption.

I'm launching a startup in the edtech space. First, outline the main challenges I may encounter. Then, suggest solutions for each.

You are a senior HR manager. Write a policy announcement for employees. You will get \$1000 if you do it really well.

# TIPS FOR EFFECTIVE USER PROMPTING

## EXAMPLE OF A GREAT PROMPT

You are a senior product manager at a global fintech company preparing a presentation for the executive board. The company is planning to launch a payments app in the US targeting young professionals aged 22-35. Recommend actionable go-to-market strategies to launch the app. Present your answer in concise bullet points for a boardroom slide. You will get \$10000 if you do this task really well.

## WHAT NOT TO DO?

Avoid ambiguity and open-endedness

Do not ignore context

Do not overload with too many complex requests in a single sentence

Do not skip iteration. Continue to refine the model and improve the response. A small change in wording can lead to an altogether different response

Do not use slang, metaphors, or culturally specific references

# **CONTEXT ENGINEERING**

**Memory Management**

**State Management**

**RAG**

**Tools**

**Prompts**

**How is it different from Prompt Engg.?**  
**It includes Prompt Engineering**

Architecting the full set of inputs, data, tools, and memory that fill the model's context window for a given task.

Prompt engineering refers to methods for writing and organizing LLM instructions for optimal outcomes.

Context engineering refers to the set of strategies for curating and maintaining the optimal set of tokens (information) during LLM inference, including all the other information that may land there outside of the prompts.

Good AI Engineers → Great Prompt Engineers

Prompt engineering refers to methods for writing and organizing LLM instructions for optimal outcomes.

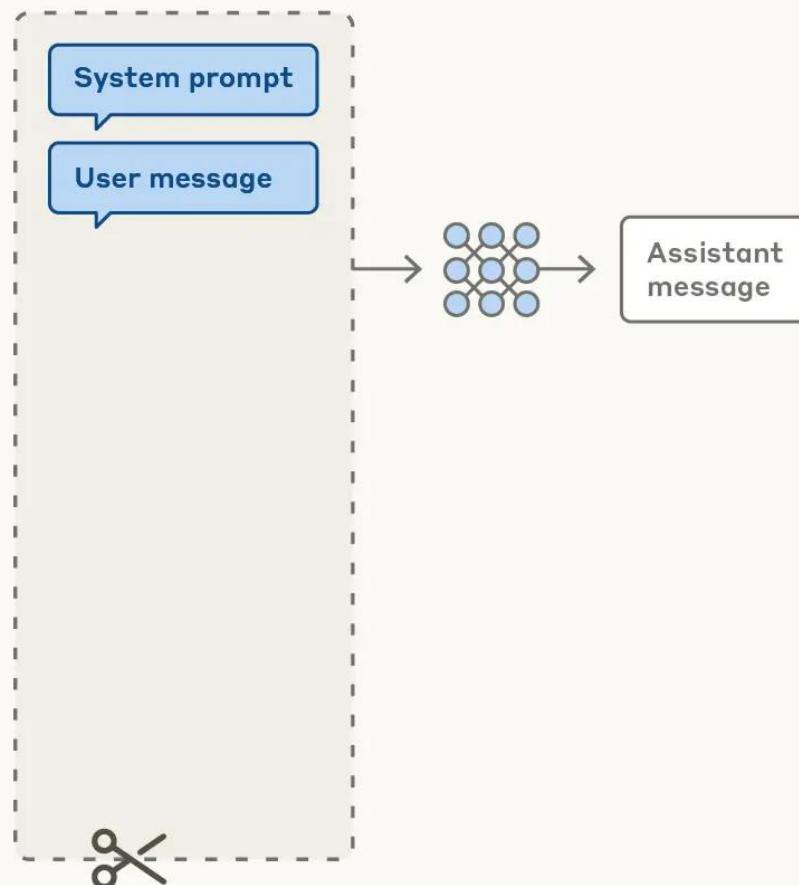
Context engineering refers to the set of strategies for curating and maintaining the optimal set of tokens (information) during LLM inference, including all the other information that may land there outside of the prompts.

Good AI Engineers → ~~Great Prompt Engineers~~ Great Context Engineers

# Prompt engineering vs. context engineering

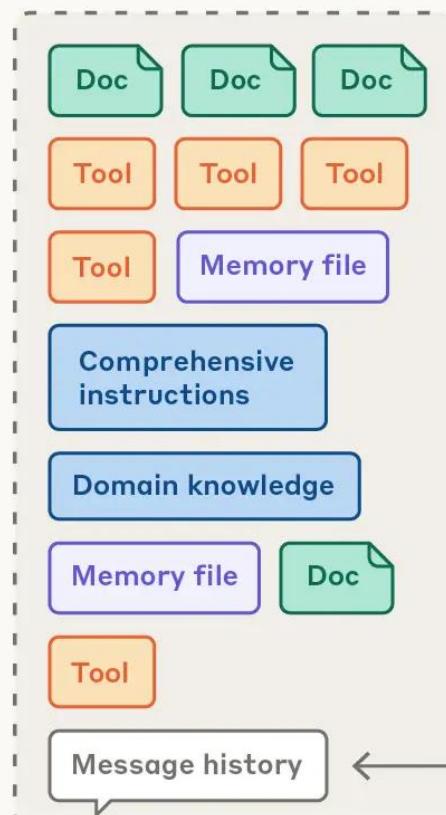
## Prompt engineering for single turn queries

### Context window

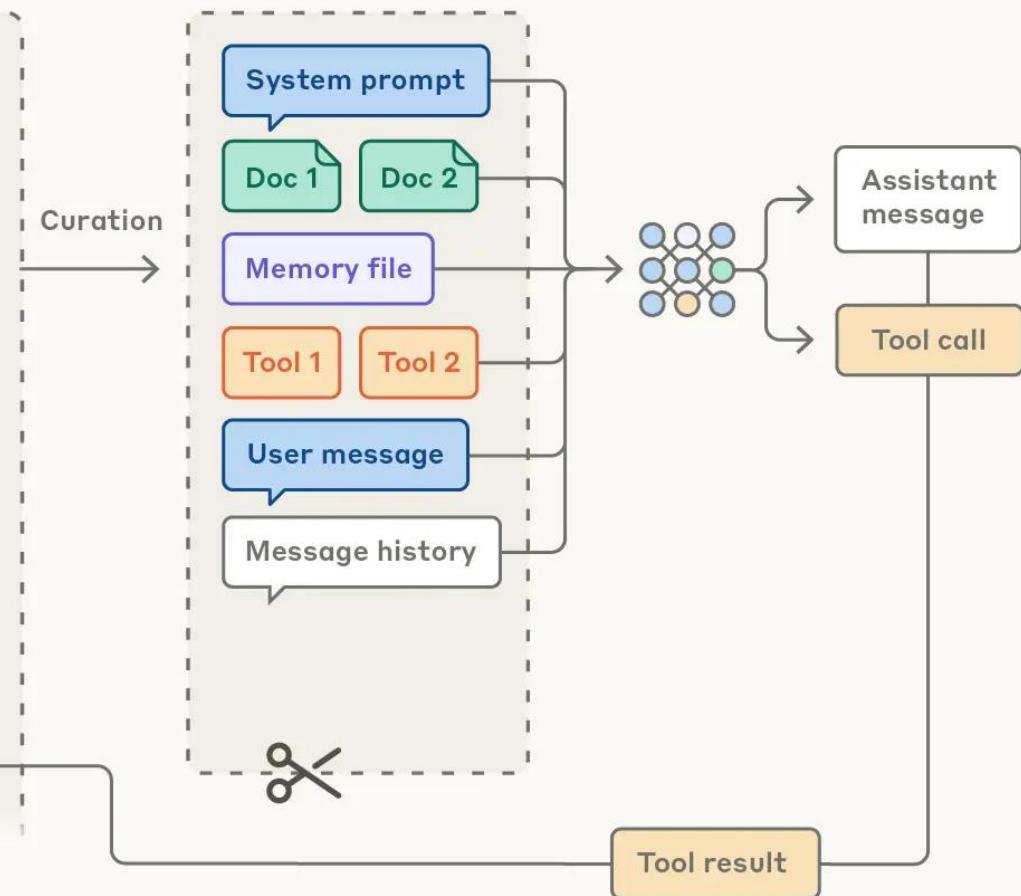


## Context engineering for agents

### Possible context to give model

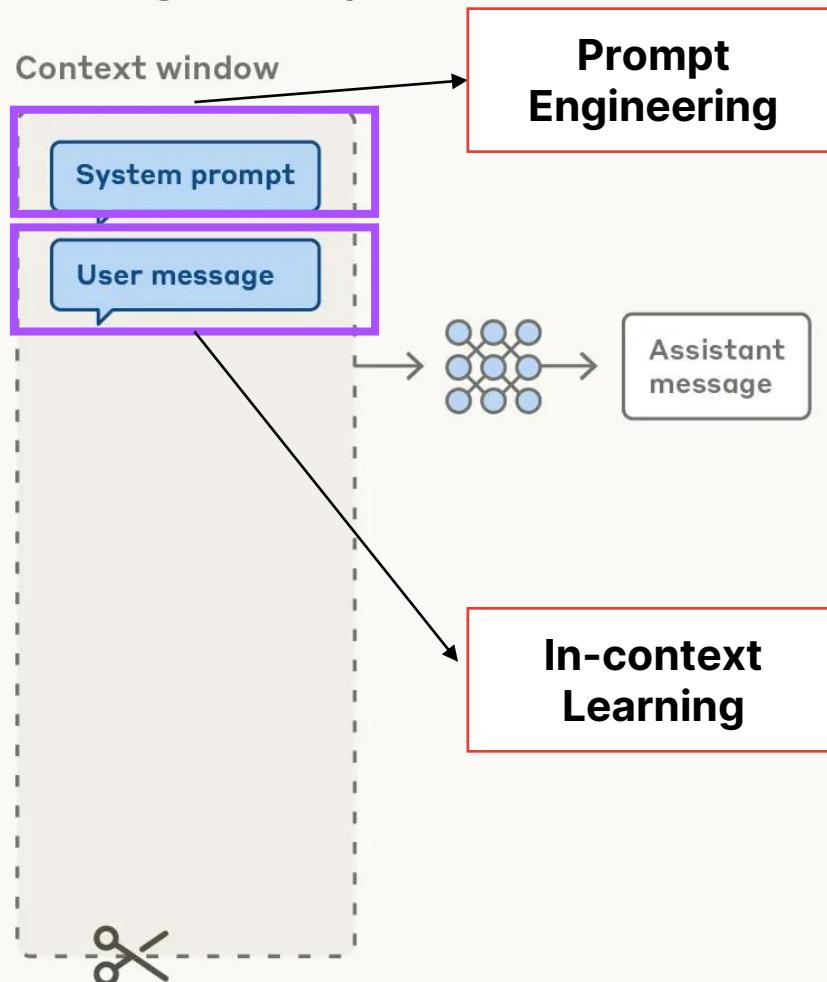


### Context window



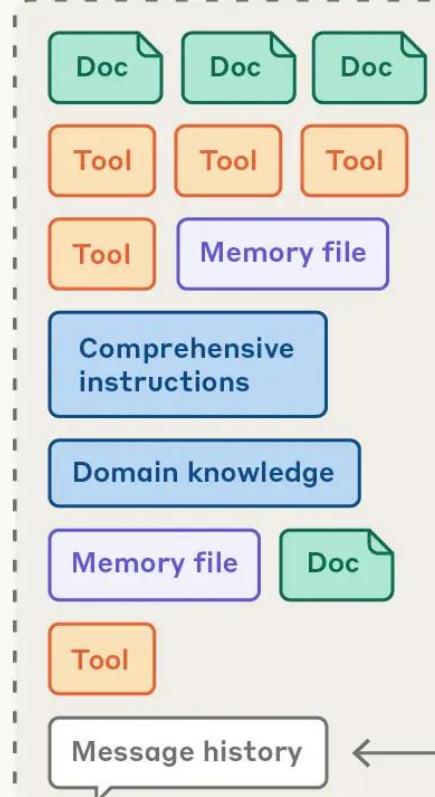
# Prompt engineering vs. context engineering

## Prompt engineering for single turn queries



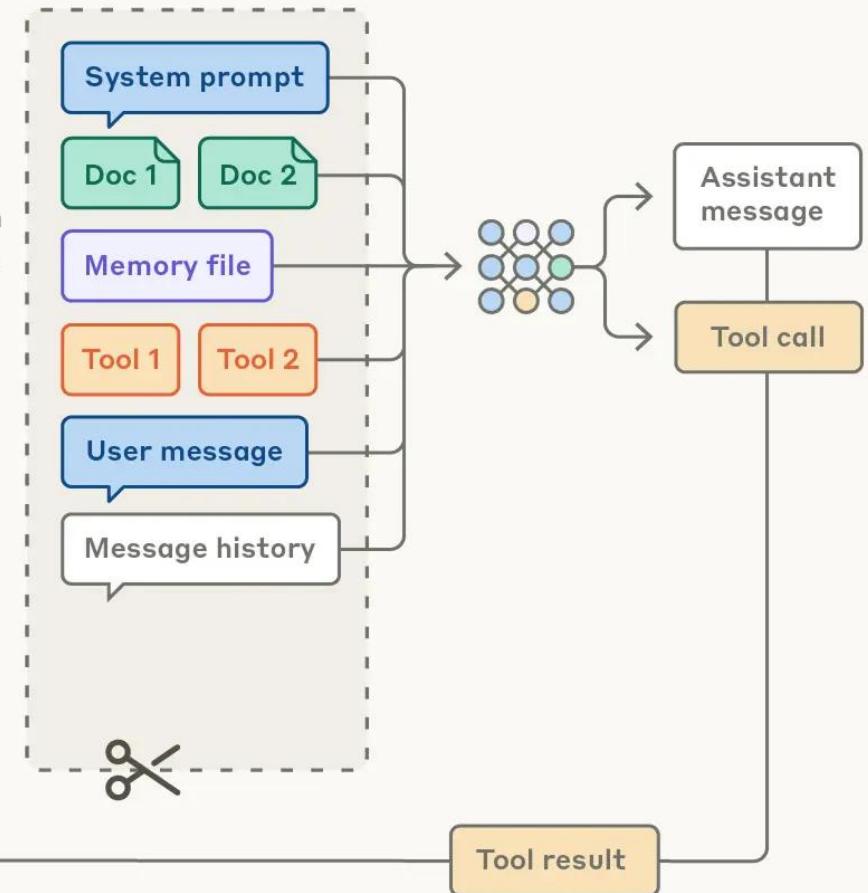
## Context engineering for agents

### Possible context to give model



Curation

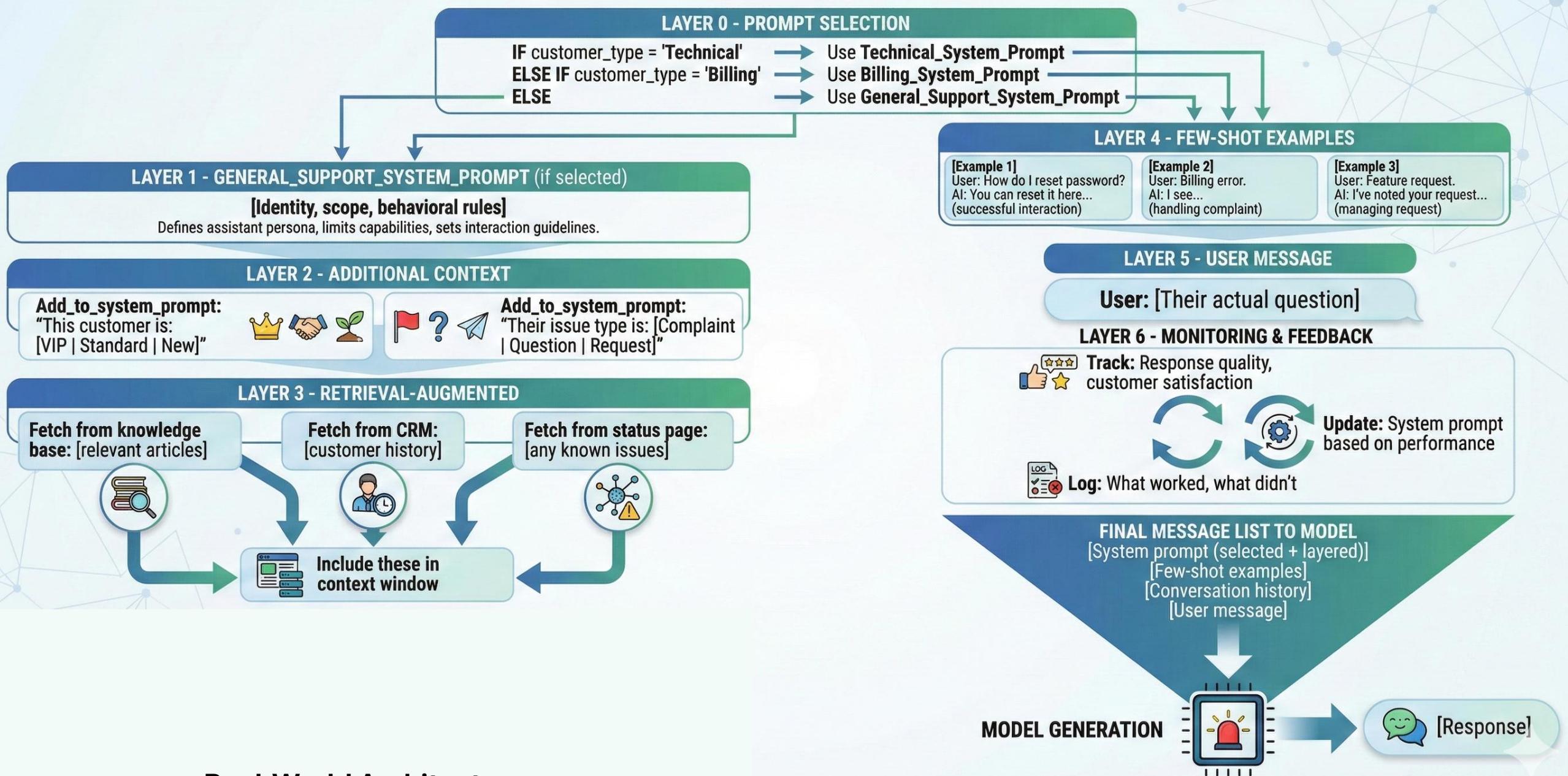
### Context window





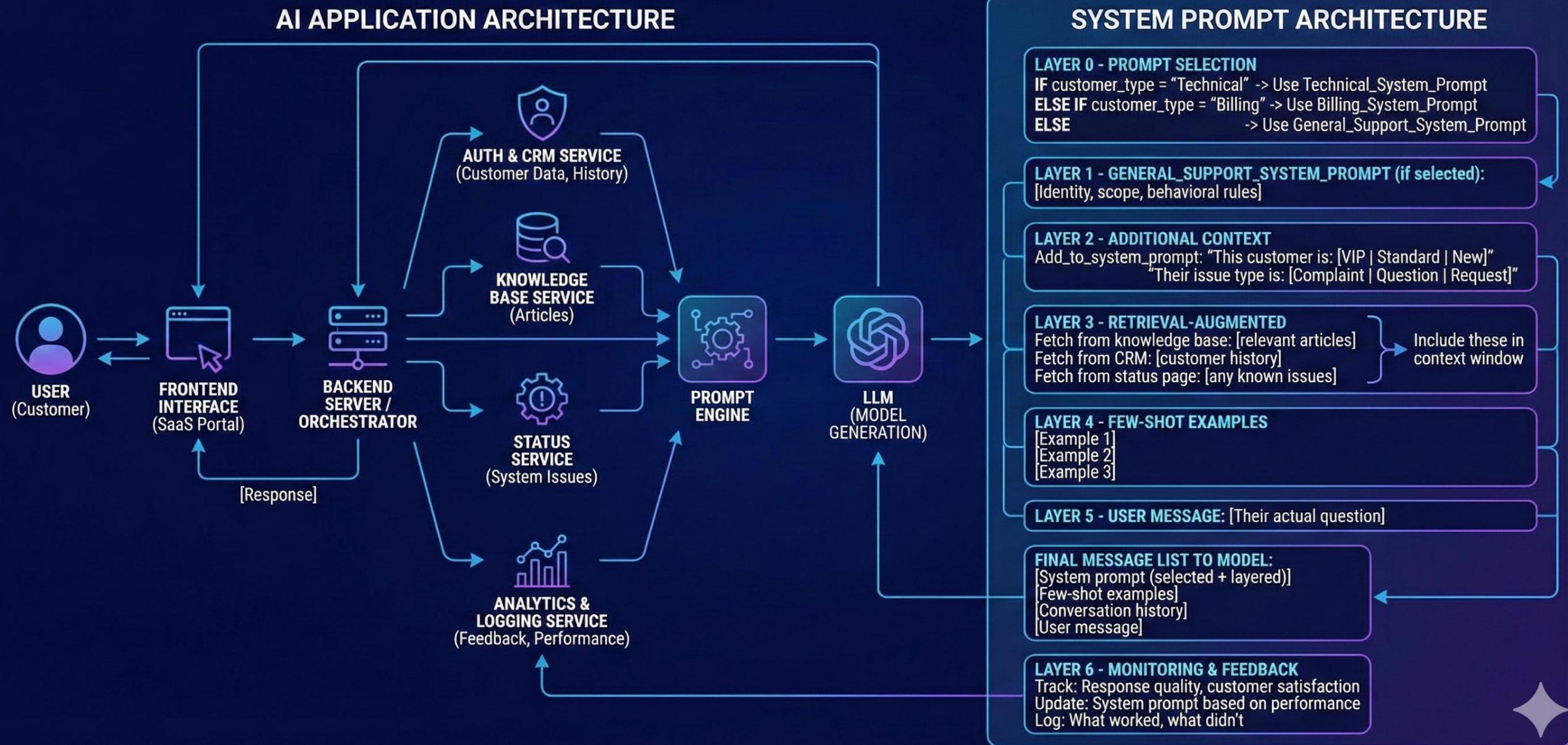
# Complete Prompt Architecture for AI Applications

# PROMPT ARCHITECTURE FOR AI CUSTOMER ASSISTANT (SaaS Scenario)



# AI CUSTOMER ASSISTANT APPLICATION & PROMPT ARCHITECTURE

(SaaS Scenario)





# BUILDING A MULTI- AGENT SYSTEM IN 30 MINUTES

# Langflow: Visual Framework for building AI apps

## What is Langflow?

Open-source, Python-based, visual framework for building AI apps

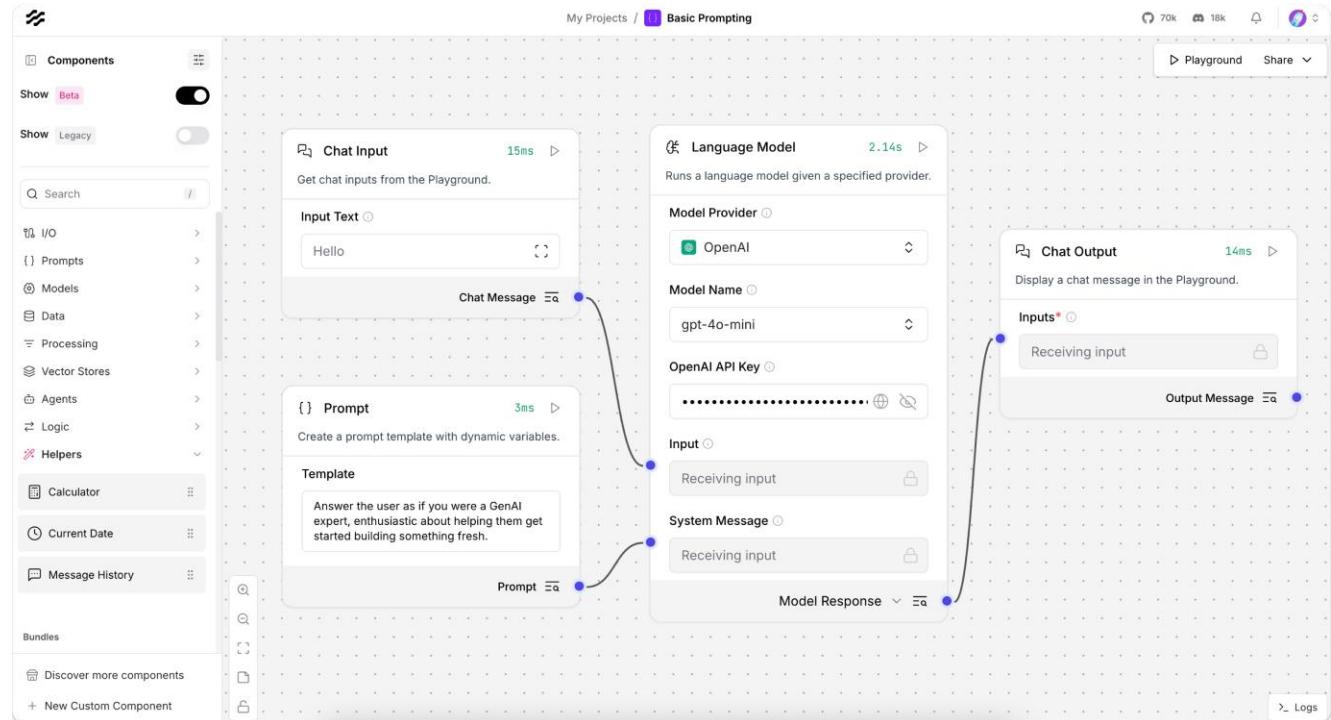
Flows = visual graphs of components (inputs, LLMs/agents, tools, data stores, outputs).  
Drag-and-drop editor for rapid prototyping and testing.

Each node represents a specific function, such as an input, a large language model (LLM), a prompt template, a vector database, or an external tool/API integration.

## Core concept

## Components

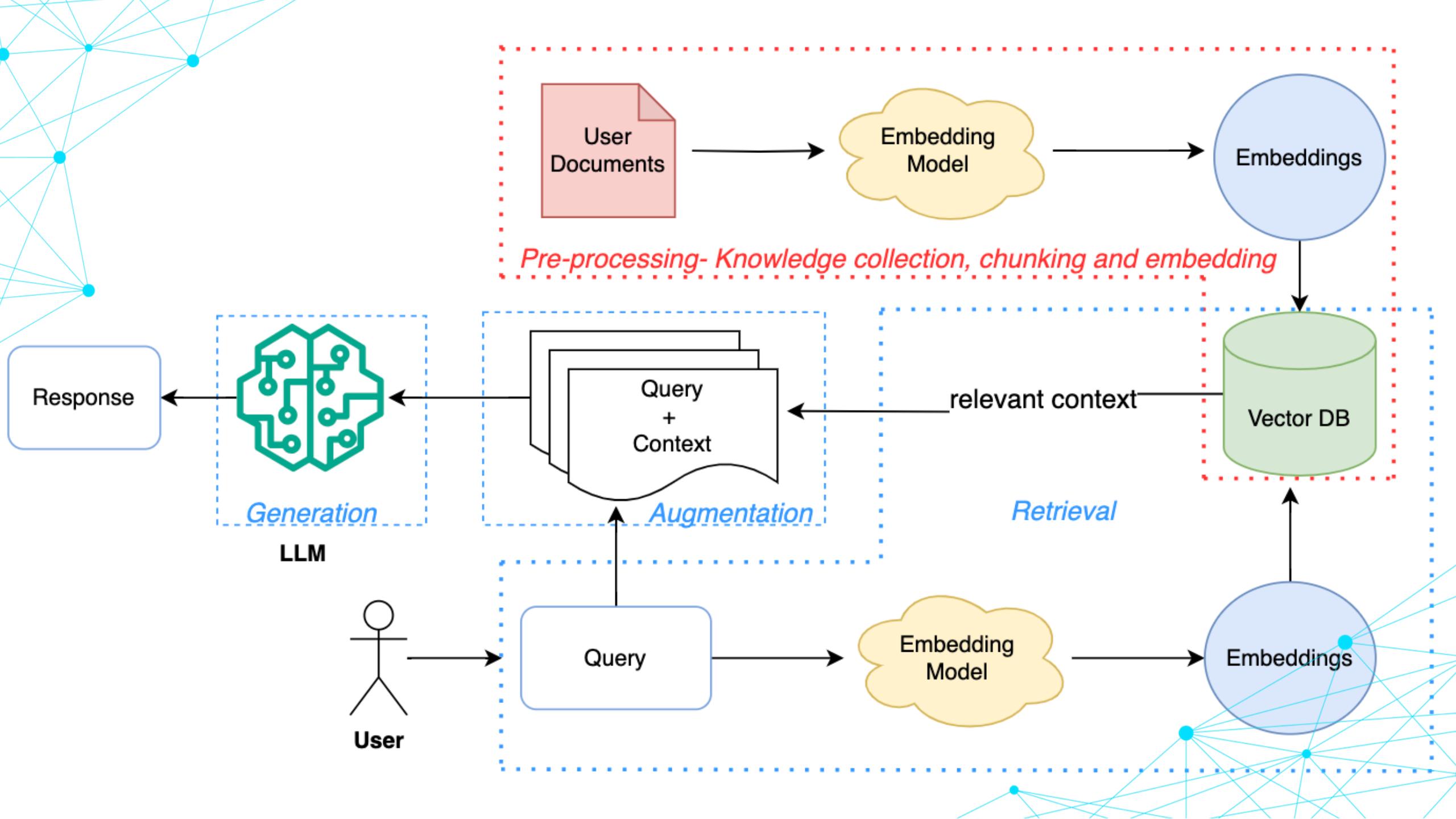
- **Agent** = LLM + tools + instructions.
- **Provider/model**: Choose a built-in provider/model or “Connect other models” by attaching a model component.
- **Memory**: Built-in chat memory per session\_id. Configure number of chat history messages.
- **Useful parameters**:  
`add_current_date_tool`,  
`handle_parsing_errors`, `temperature`,  
`verbose`, `chat history limit`.
- **Debugging**: Playground shows tool calls, inputs, and raw results.

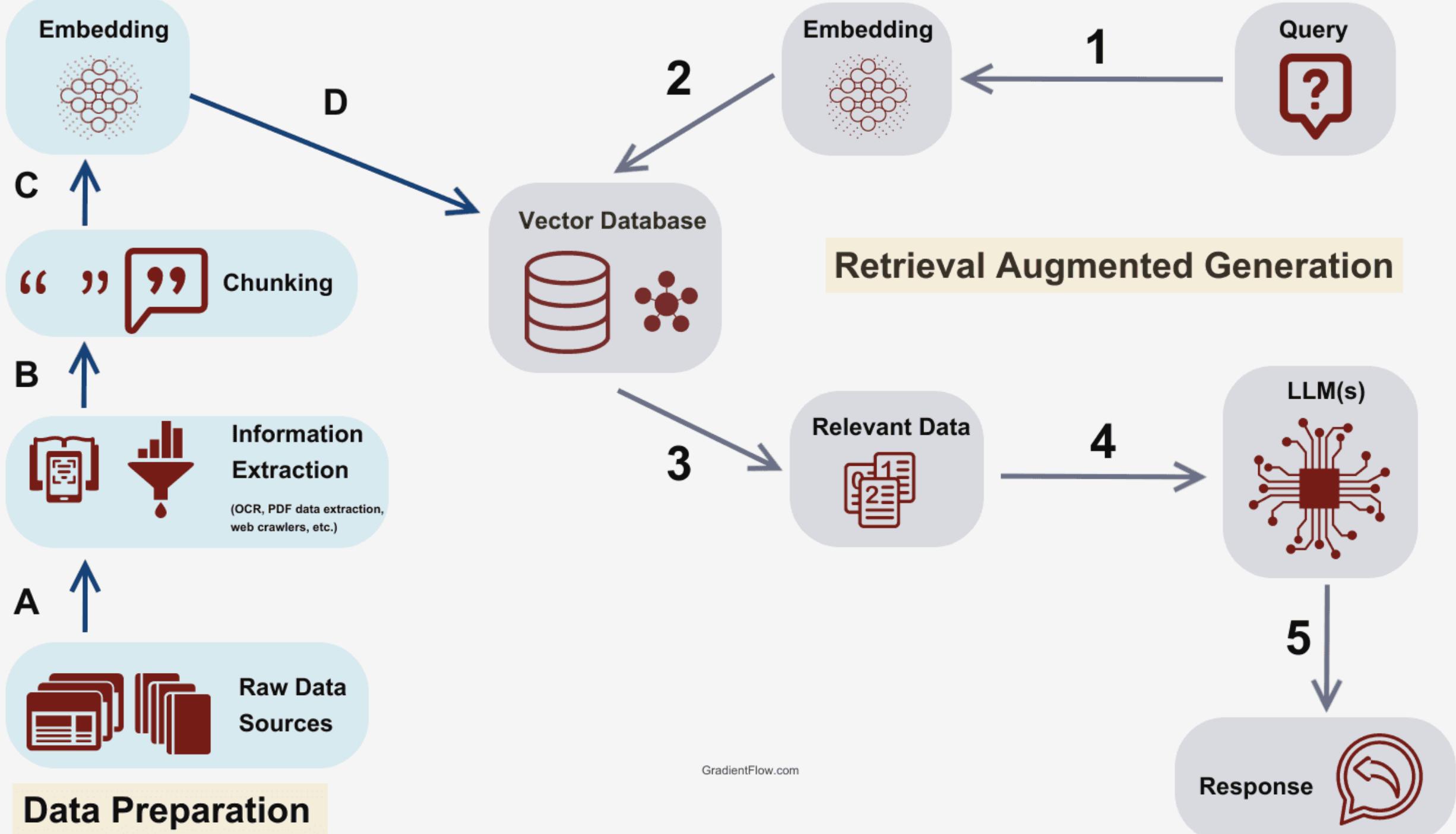


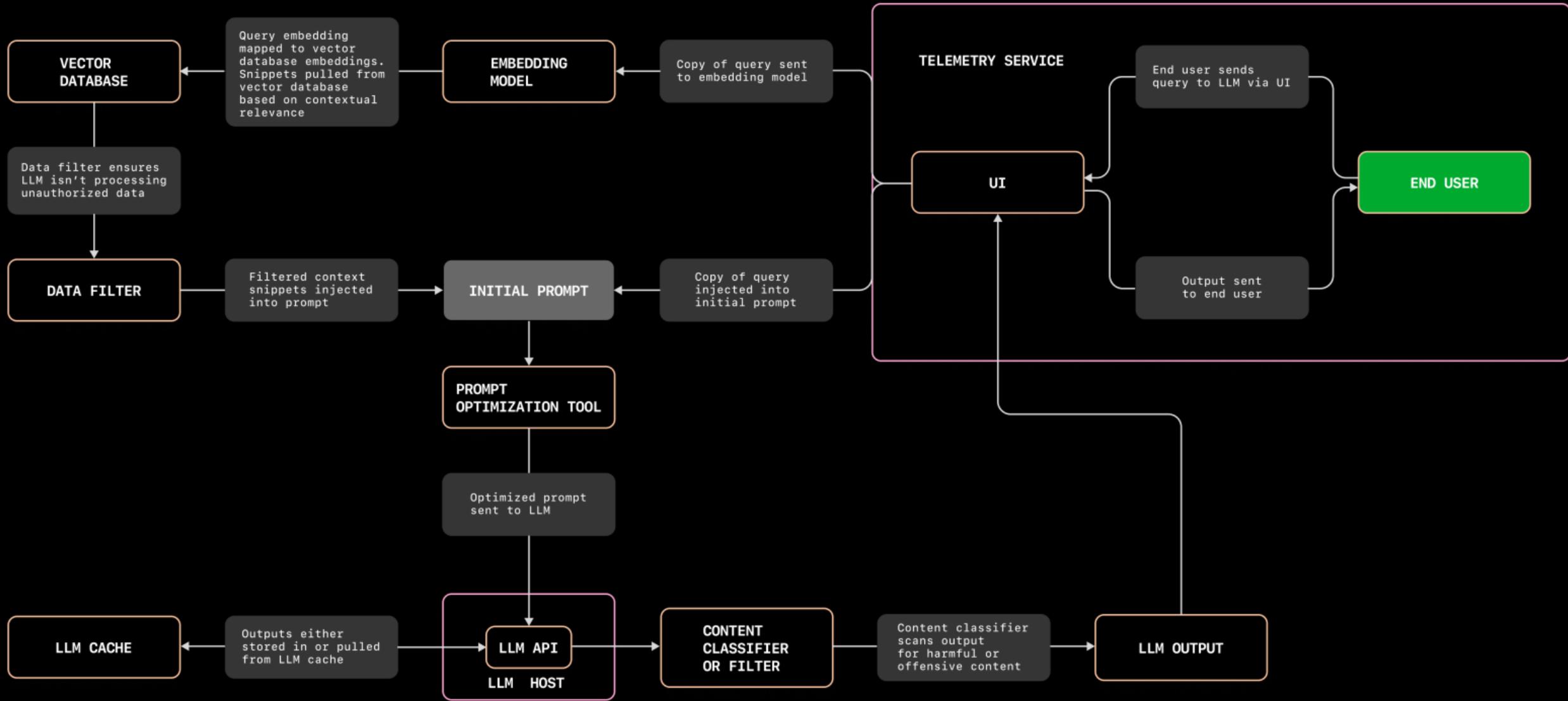
# 15 MINUTE BREAK



# COMMON ARCHITECTURES







THIS DIAGRAM REPRESENTS THE ARCHITECTURE OF TODAY'S LLM APPLICATION. THE DIFFERENT COMPONENTS CAN BE ROUGHLY GROUPED INTO THREE CATEGORIES: USER INPUT, INPUT ENRICHMENT TOOLS AND PROMPT CONSTRUCTION, AND EFFICIENT AND RESPONSIBLE AI TOOLING.

# Q&A



# AI SPRINT: AGENTS IN ACTION

## ANALYTICS VIDHYA

ASHWIN GOYAL

Welcome to Day 2

JANUARY 2026

**BUILD APPLICATIONS WITHOUT CODE USING  
LOVABLE**

**EVERYONE IS A DEVELOPER**

**THE LAST PIECE OF SOFTWARE**

**VO BY VERCEL  
BOLT  
EMERGENT  
REPLIT  
CURSOR**

# **PORTFOLIO BUILT IN 60 MINUTES**

# OUR OWN TRAVEL STARTUP (TRIP PLANNER)

1

IDEA GENERATION

2

AMAZON-STYLE  
PRD/PRFAQ

3

MARKET RESEARCH

4

ENGINEERING  
EXECUTION PLAN



## HANDS-ON ACTIVITY: GEN-AI

**5**

PRODUCT CREATION



**GitHub**  
Copilot

 **Claude**  
 **Lovable**

**6**

MARKETING WEBSITE



**7**

PITCH DECK FOR  
INVESTORS



**Gamma**

 NotebookLM

**8**

LAUNCH THE PRODUCT



**Writesonic**



**invideo**

# Lovable

**Describe your app idea, AI builds it for you (full-stack, in browser).**

**Think of it as:** ChatGPT for app development, but it actually builds working apps.

**Project:** Validate a new business idea

**Perfect for:**

- MVPs (Minimum Viable Products)
- Quick prototypes to validate ideas
- Startups building proof-of-concept
- Teams that don't have developers
- Iteration and feedback cycles
- Learning what's possible

# **MAN'S WORST ENEMY**

**INTEGRATION OF FRONTEND WITH BACKEND, MULTI-AGENT SYSTEMS AND DATABASE**

## **FRONTEND + BACKEND + DATABASE**

Travel App in Lovable to build trip plans

- Fully-functional Frontend
- Lovable hosted backend
- Supabase/Lovable Database

## **MULTI-AGENT AI SYSTEM**

Multi-agent system built in Langflow

- Multiple agents
- Tool calling
- Web search

## **FULLY FUNCTIONAL PRODUCT READY TO BE SHIPPED**

Full fledged travel planning and booking app.

**Next Step:** Create marketing/sales material using AI and **SHIP**

# **MAN'S BEST GIFT TO HIMSELF**

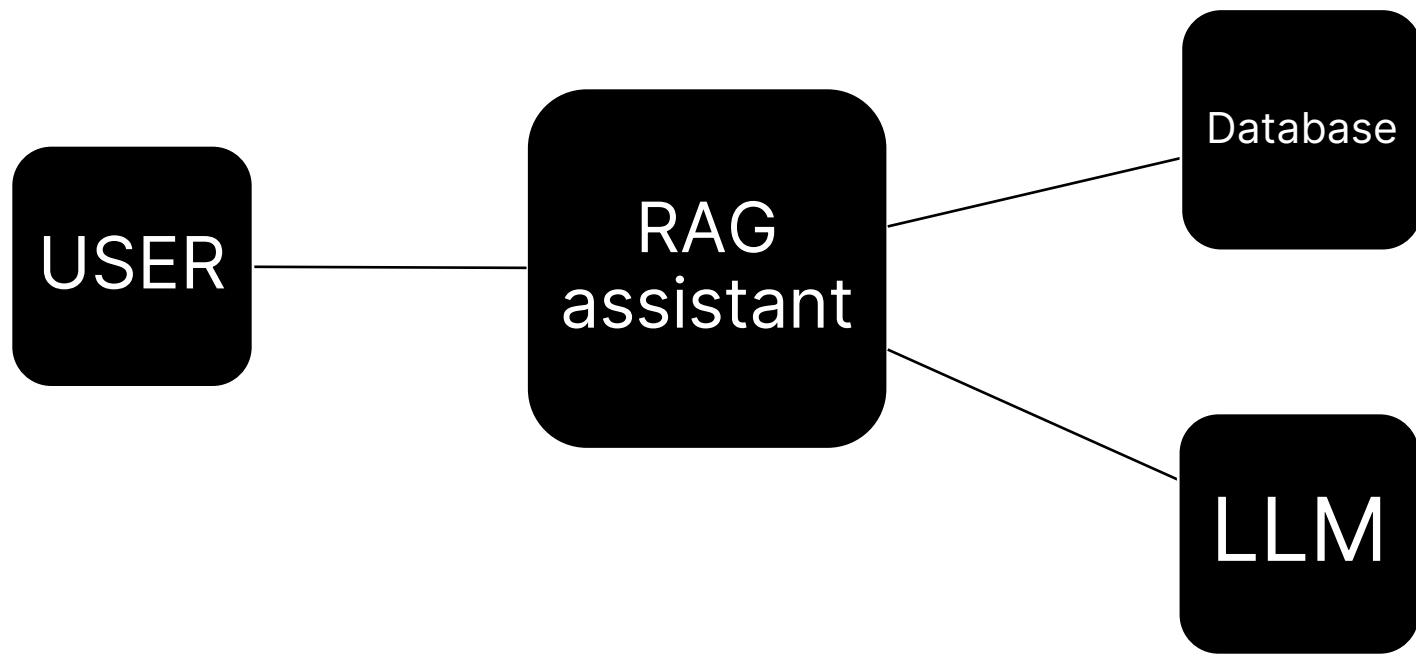
# **RETRIEVAL AUGMENTED GENERATION**

**RAG**

**90% OF USECASES IN COMPANIES ARE RAG-BASED**

**RETRIEVAL-AUGMENTED GENERATION TRANSFORMS LLMS FROM HALLUCINATION-PRONE BLACK BOXES INTO RELIABLE SYSTEMS GROUNDED IN YOUR DATA.**

**BY COMBINING SEMANTIC SEARCH WITH GENERATION, RAG ENABLES AI TO ANSWER QUESTIONS ACCURATELY USING CURRENT, DOMAIN-SPECIFIC INFORMATION.**



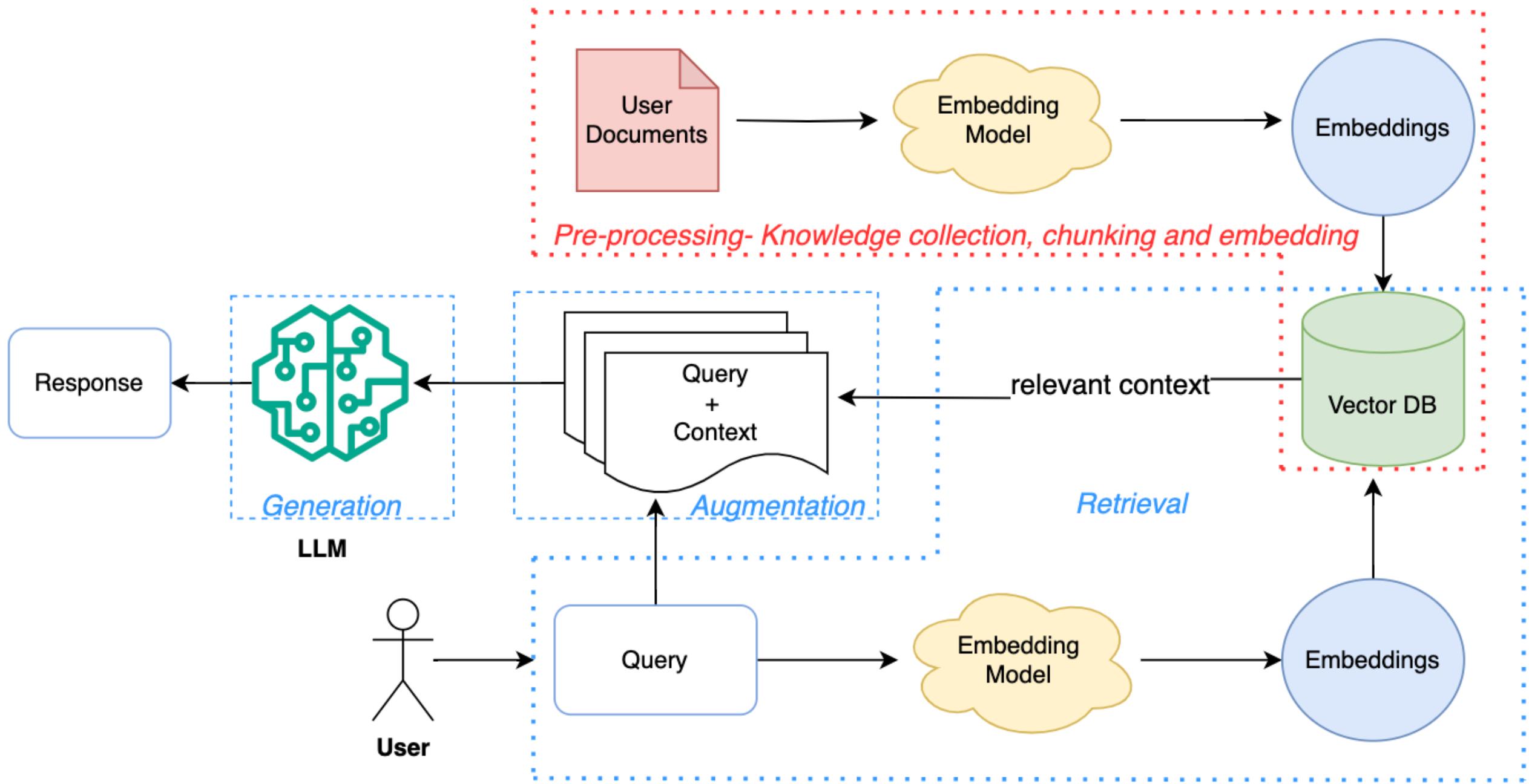
# **VECTOR EMBEDDINGS**

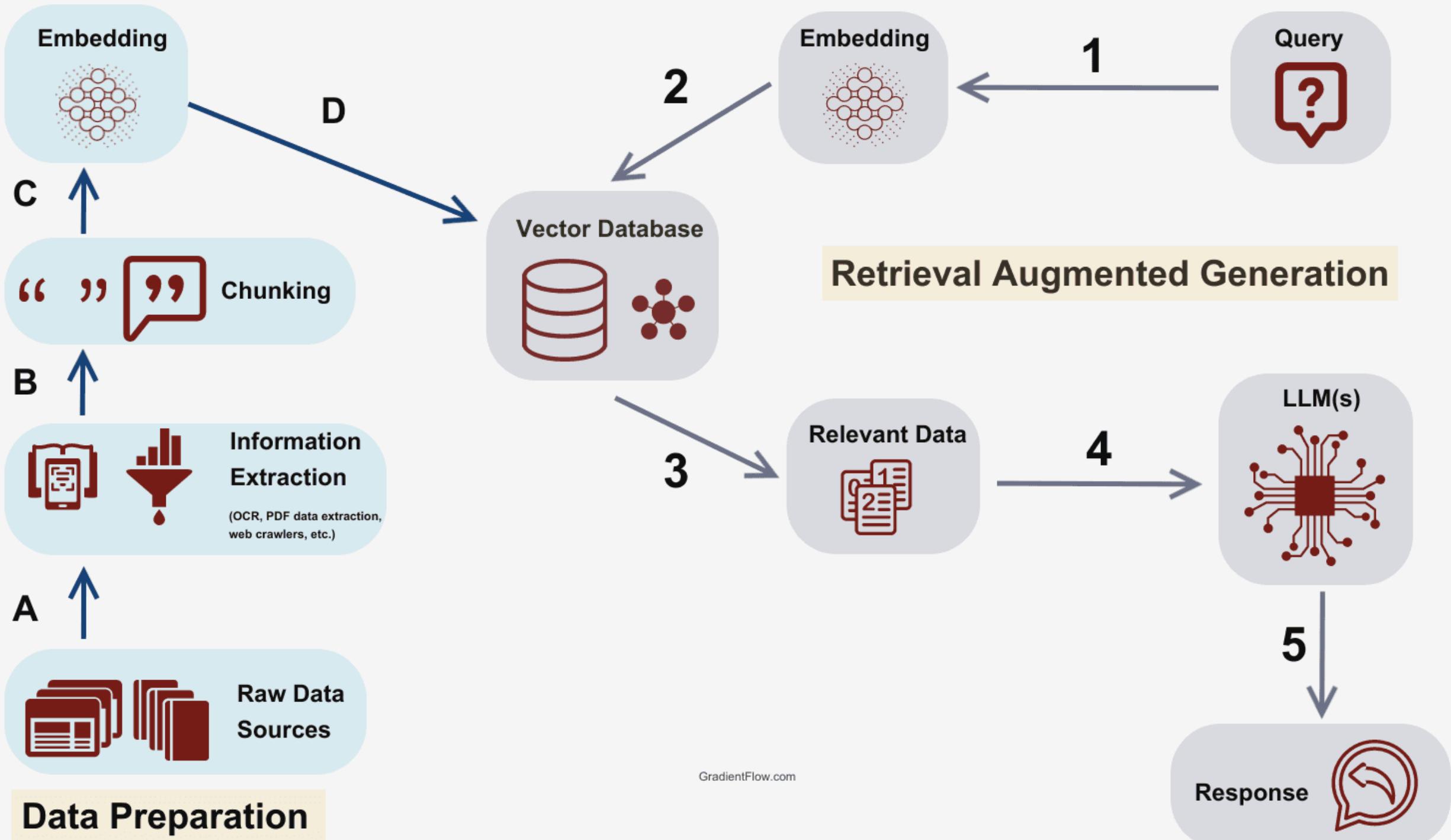
# **NUMERICAL REPRESENTATIONS OF WORDS, PHRASES, SENTENCES, DOCS**

# **EASY INFORMATION RETRIEVAL**

**KING ~ QUEEN**

**LLMS USE EMBEDDINGS TO MAP EVERY TOKEN  
(WORD OR SUBWORD) INTO A VECTOR BEFORE  
PROCESSING IT.**





# N8N

# WORKFLOW AUTOMATION

- **NODES**
- **JSON-BASED**
- **OPEN-SOURCE**
- **TRIGGERS**
- **CONNECTORS**

**SEAMLESS CONNECTIVITY ACROSS TOOLS**

# 10 MINUTE BREAK



**SET UP AN ACCOUNT ON N8N**

# **CREATE AN N8N WORKFLOW IN 4 EASY STEPS!**

**1**

---

**PICK A TEMPLATE/CREATE  
NEW**

**2**

---

**LINK ALL  
CREDENTIALS/PROVIDE API  
KEYS**

**3**

---

**GO THROUGH TEMPLATE  
INSTRUCTIONS**

**4**

---

**SET UP AND TEST ALL  
NODES**

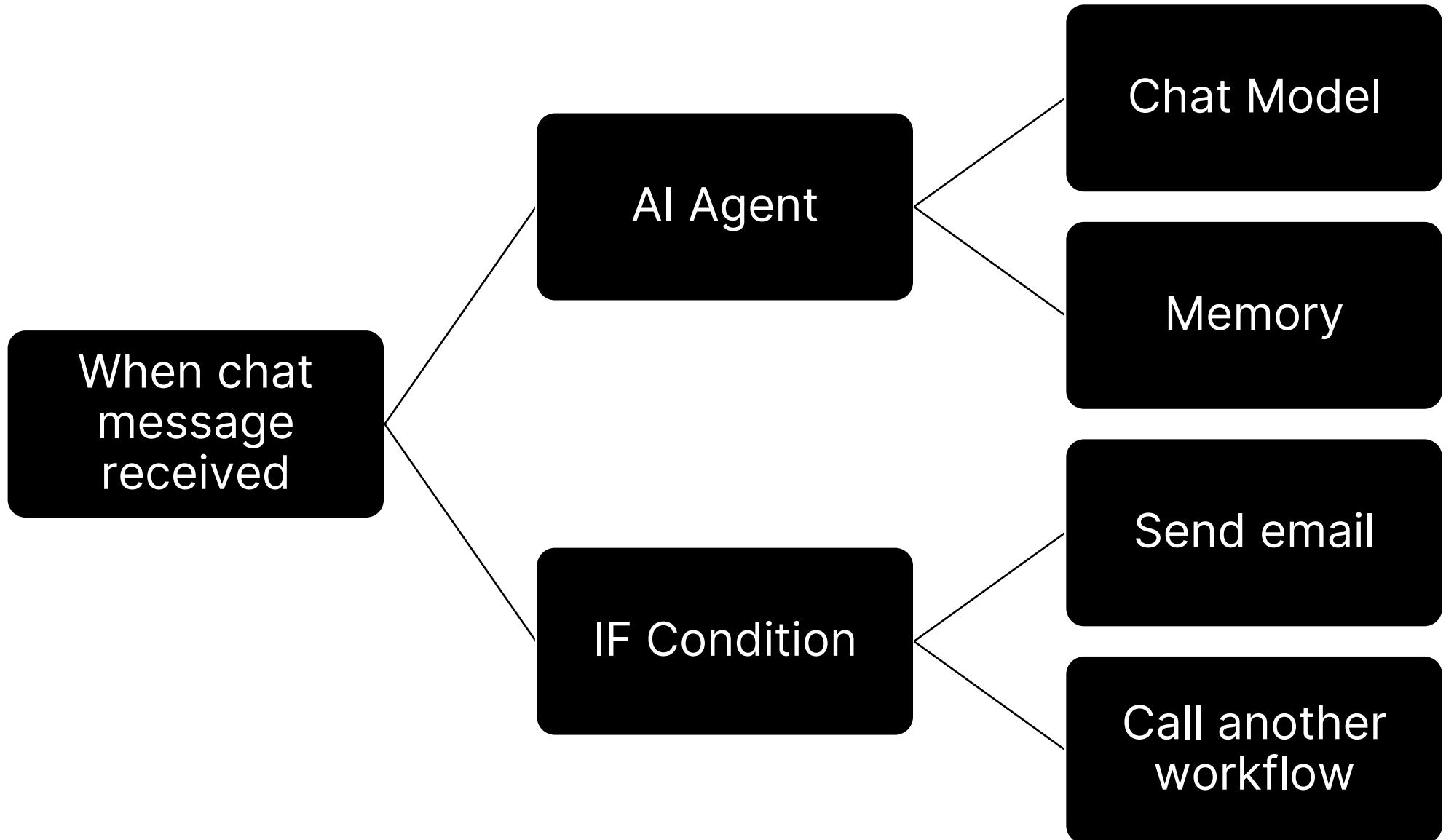
**5. EXECUTE WORKFLOW**

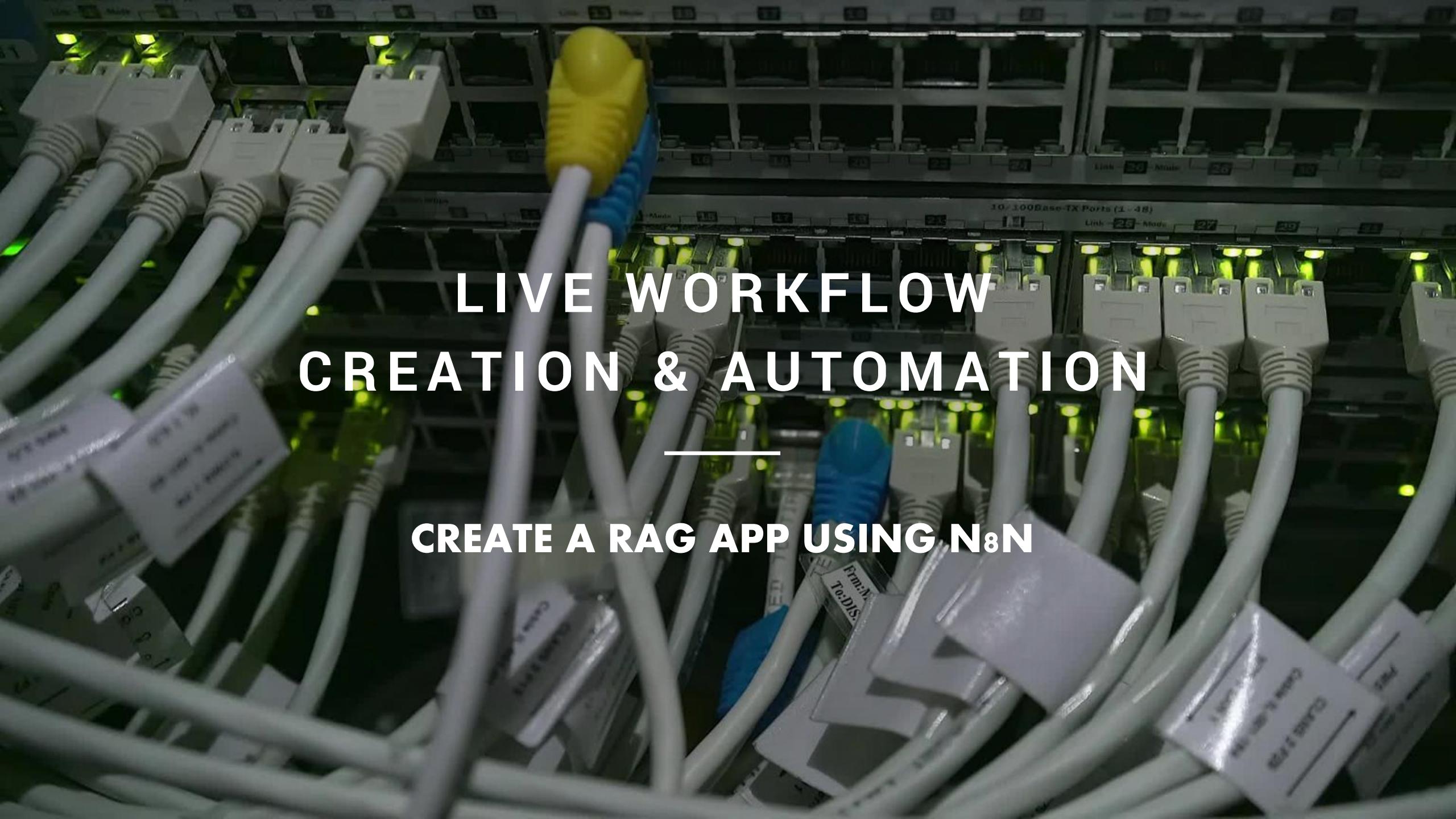
# CREATE A WORKFLOW IN N8N

## NODE CATEGORIES

- **TRIGGERS**
- **ACTIONS (GOOGLE SHEETS, NOTION)**
- **UTILITY (IF STATEMENTS, FILTERS)**
- **CODE (RUN CODE, SET WEBHOOKS)**
- **ADVANCED AI (AGENTS, MEMORY)**

## CREATE A SIMPLE WORKFLOW





# LIVE WORKFLOW CREATION & AUTOMATION

---

CREATE A RAG APP USING N8N

# EVALS: YOUR CUSTOM TESTING PROCESS

YOUR systematic process for testing YOUR AI feature for YOUR users

# AI MODEL EVALUATIONS

EVALS

# AI MODEL PRODUCT EVALUATIONS

EVALS

# TOP 10 EVAL METRICS

ACCURACY

RELEVANCE

COHERENCE

PERPLEXITY

FAITHFULNESS

BLEU (BILINGUAL EVALUATION UNDERSTUDY)

ROUGE (RECALL-ORIENTED UNDERSTUDY FOR  
GISTING EVALUATION)

TOXICITY

LATENCY

HALLUCINATION RATE

# MODEL EVALUATION TECHNIQUES

## LLM-AS-A-JUDGE

**LLM itself evaluates quality of its outputs. It uses another AI model to evaluate outputs.**

- **Comparing with golden dataset/ground-truth data set**
- **Use Metrics such as accuracy, perplexity**

## HUMAN-IN-THE-LOOP EVALUATION

**Human evaluators assess the quality of LLM outputs**

- **Use Metrics such as coherence, relevance, helpfulness**

# LLM as a Judge: LLM Evaluating Output of Another LLM

## PROMPT

**QUERY** I want to go...

**CONTEXT** User booking history, hotel inventory...

**INSTRUCTIONS** You are a friendly AI assistant...



## APP in Production

## RESPONSE

{response}

Hi, I'm happy to help you plan...

## EVAL PROMPT TEMPLATE

<<template: hallucination>>

Is the {response} using the {context} to answer the {query}

## EVALUATOR LLM

ANTHROPIC

cohere

Gemini

MISTRAL  
AI\_

OpenAI

## LLM-GENERATED EVAL

**QUERY**

I want to go...

**CONTEXT**

Hotel inventory, local sites ...

**INSTRUCTIONS**

Hi, I'm happy to help ...

## DID IT HALLUCINATE?

Eval: ✘ No

# BUILD AGENTS WITH OPEN AI AGENTKIT

**AGENT BUILDER**

**CHATKIT**

**GUARDRAILS**

## AGENTKIT

- Create workflows with **Agent Builder**
- Use **ChatKit** to embed your agent workflows in your frontend
- Build robust **evals** to observe and improve agent performance

Create

Chat

Agent Builder

Audio

Images

Assistants

Manage

Usage

API keys

Logs

Storage

Batches

Optimize

...

...

## Agent Builder

### Create a workflow

Build a chat agent workflow with custom logic and tools

[+ Create](#)

#### Data enrichment

Pull together data to answer user questions

[Template](#)

#### Planning helper

Simple multi-turn workflow for creating work plans

[Template](#)

#### Customer service

Resolve customer queries with custom policies

[Template](#)

## Core



Agent



End



Note

## Tools



File search

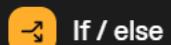


Guardrails



MCP

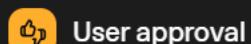
## Logic



If / else

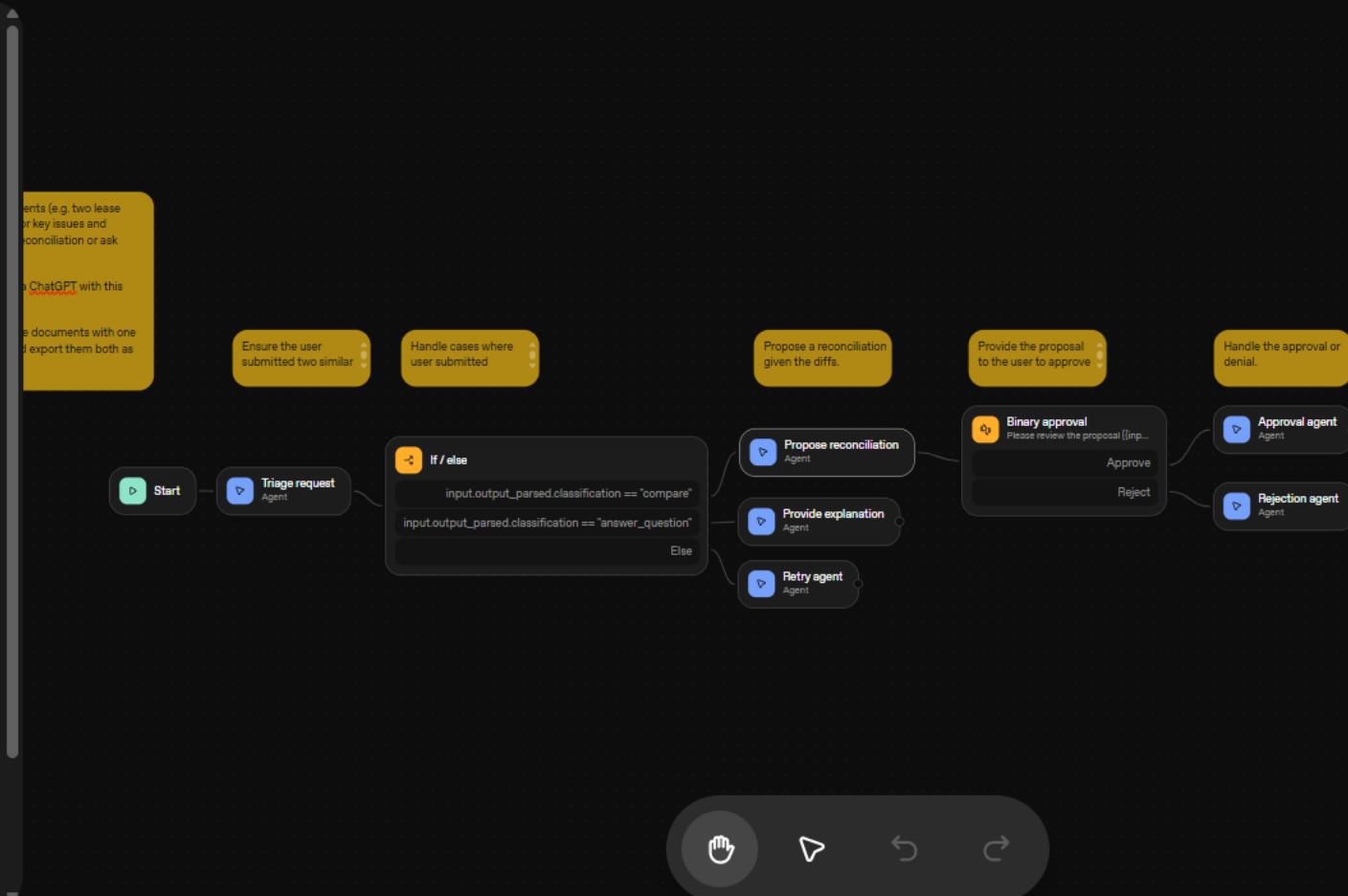


While

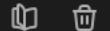


User approval

## Data



## Propose reconciliation



Call the model with your instructions and tools

Name

Propose reconciliation

Instructions

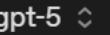


Given the differences between the two documents, assemble a single option for how to reconcile the difference. If no order has been described, consider the first document the

Include chat history



Model



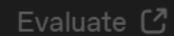
Reasoning effort



Tools



More



## New workflow

Draft

Close preview X

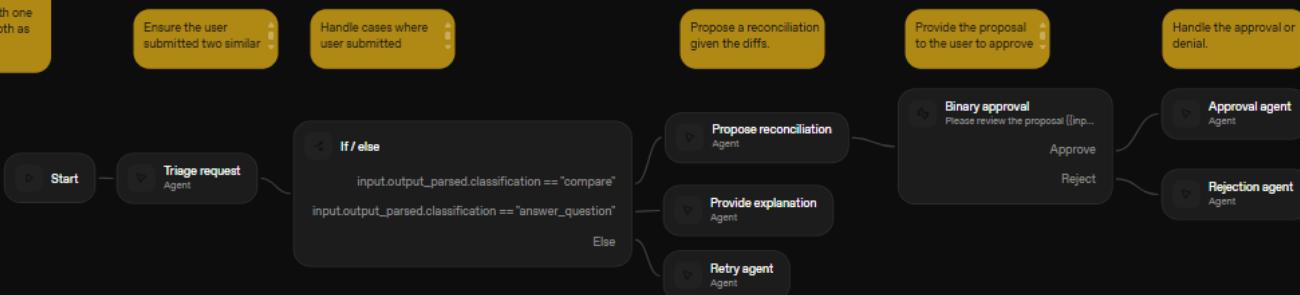
Publish

### Preview

Upload two similar documents (e.g. two lease documents) to compare for key issues and recommend a proposed reconciliation or ask clarifying questions.

You can generate these via ChatGPT with this prompt:

"Generate two similar lease documents with one substantive difference and export them both as PDFs."



### Test the workflow

Send a message...



# **THANK YOU**

**Can't wait to see what you build!**

**Let's connect on LinkedIn!**