Typical prod-ready RAG LLM solution challenges

[1]

Building a Production-Ready LLM: key components and structure.

[2]

RAG Storage Options: types and options.

[3]

LLM Testing Strategies: cost and approaches.

[4]

Tackling Hallucinations: Practical approaches in LLM development.

Cookbook.

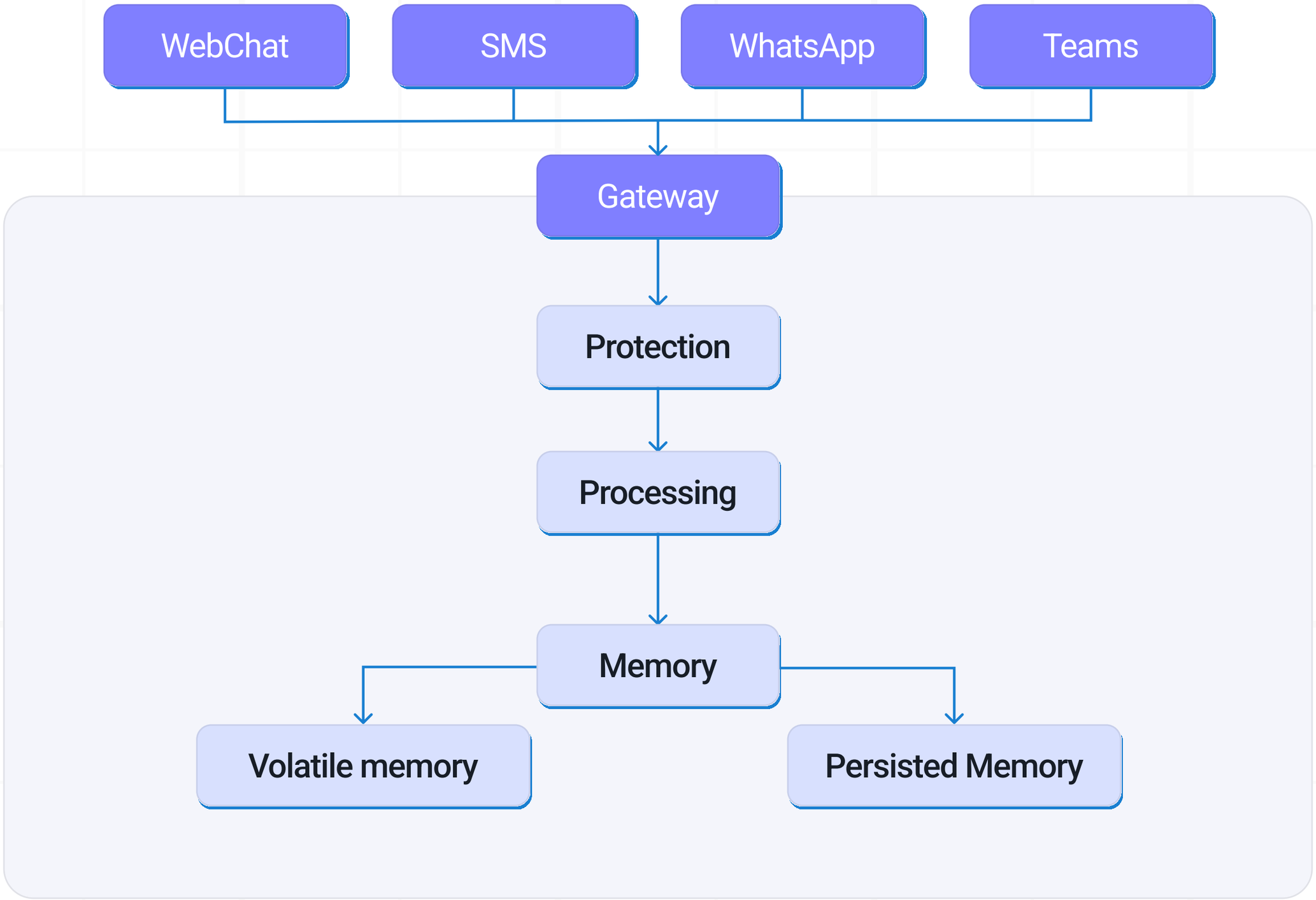
PS Before diving deeper: even though it's a cook book, I intentionally left a place for investigations.

Aleksei Kolesnikov
Staff Software Engineer

1. Building a Production-Ready LLM: Pain Points and Industry answers

1. Diverse User Channels - Managing integration across multiple communication channels.
2. Safety and Protection - Ensuring that both input and output are free from harmful content.
3. LLM Architecture - Developing a scalable and testable architecture that can efficiently handle varying loads.
4. Memory Management in LLMs - Organizing volatile and persistent memory.
5. Testing LLM Solutions - Determining key areas for testing and the costs.

Understanding these challenges allows to separate concerns effectively and propose tailored solutions.



Modern LLM applications share many fundamental similarities. It is crucial to leverage existing Industry Standards and Frameworks and avoid unnecessary duplication of efforts wherever it is possible.

In order to build a good LLM RAG Solution you need to separate it onto building blocks: Gateway, Protection, Processing, and Memory

1.1 Gateway

An LLM Gateway is an optional component of your solution that serves to consolidate various user channels.

This integration allows for efficient management using other existing building blocks.

Several effective frameworks and SDKs are available to assist in this regard:

Azure Bot Service

AWS ChatOps

DialogFlow by Google

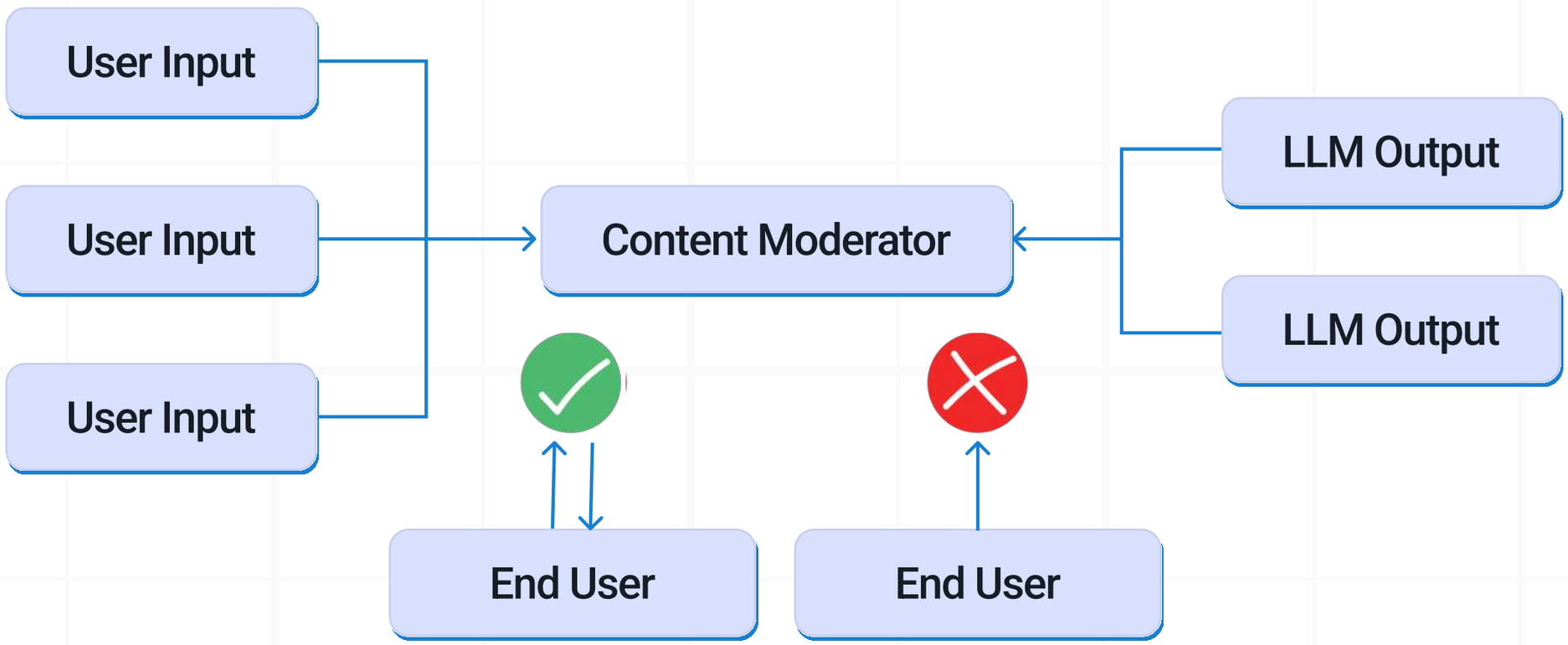
Twilio Autopilot

Each service comes with its own set of advantages and disadvantages:

- Azure Bot Service offers a comprehensive framework and ready-to-use bots.
- Twilio Autopilot provides machine learning capabilities and native support for the SMS channel.
- AWS ChatOps and Dialogflow offer seamless integration within the AWS and GCP ecosystems, respectively.

1.2 Protection

A critical moment of any Prod-Ready LLM solution
is ensuring consistent and safe interactions that protect
both the end-users and your business.



Azure Content Moderator

Sightengine Content Moderation

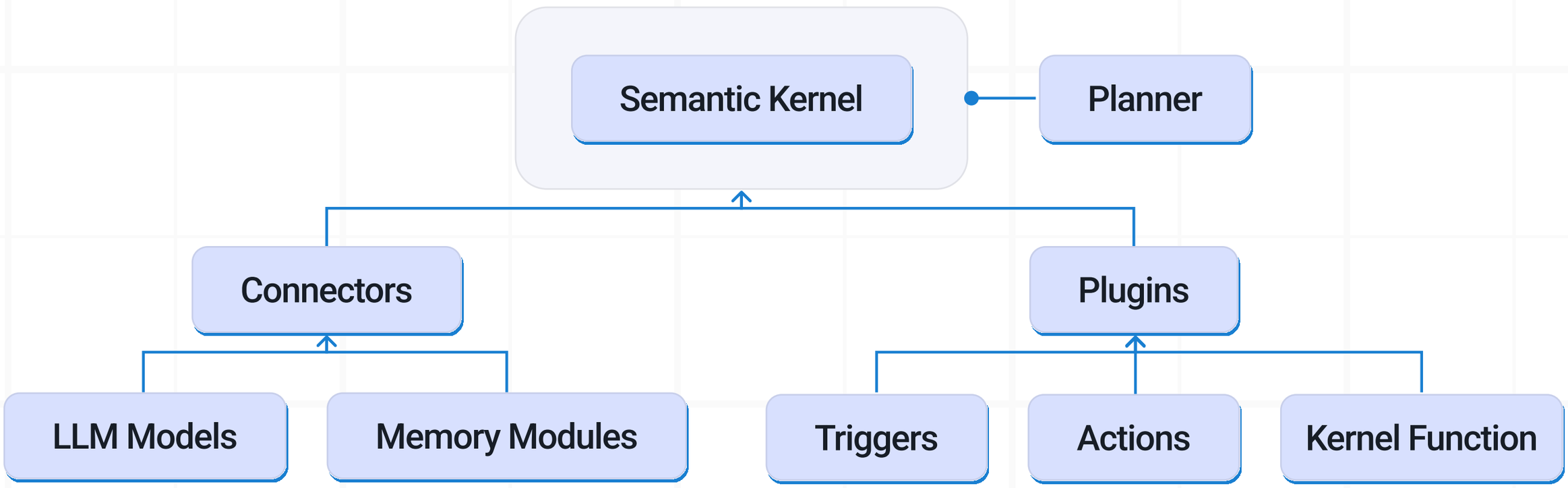
Perspective API by Jigsaw

Disclaimer: regardless of the effectiveness of your RAG solution, generating harmful content can lead to legal consequences and damage your reputation.

DPD and McDonalds cases are very informative.

1.3 Processing.
Semantic Kernel & LangChain

This is your “Cooking Kitchen”.
Its goal - bridges the gap between technical implementation and business functionality which simplifies further solution analysis.

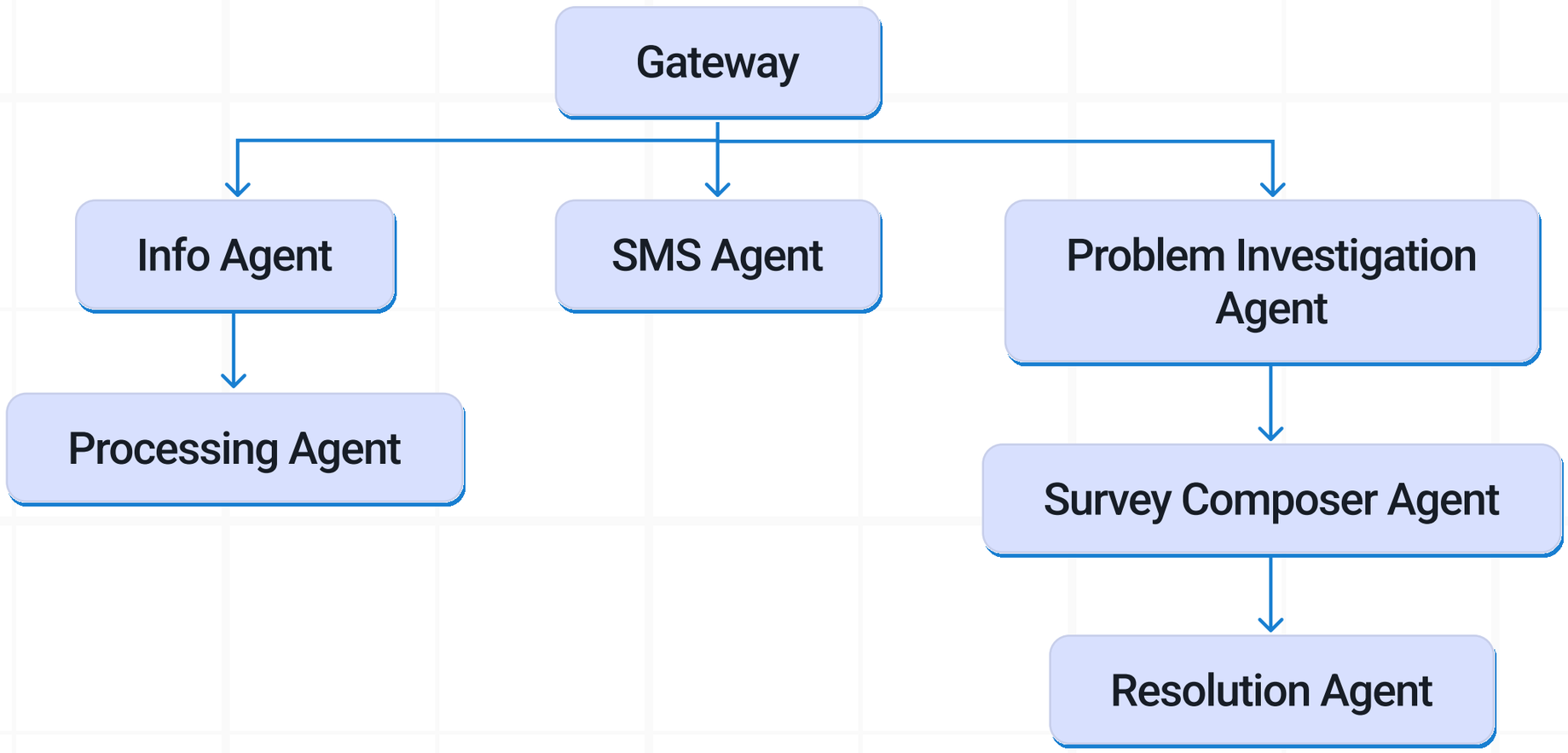


Disclaimer: looks like a typical Layering Architecture features with horizontal and vertical slices.

This design enables the replacement or selective use of modules.

It also facilitate each module testing.

1.3 Processing.
LLM Agents using SK or LangChain



- What happens if you need to manage various channels like SMS, Web Chat, or Audio differently?
- What if addressing a request involves tackling distinct issues?

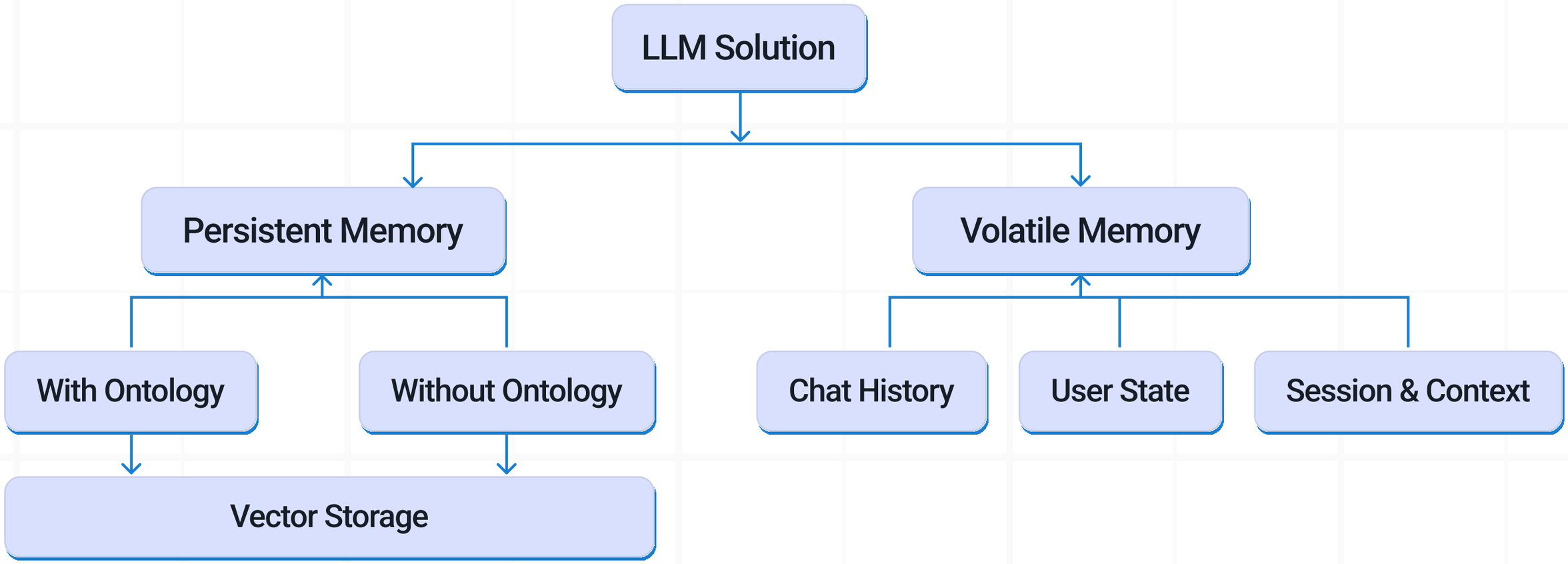
Divide and conquer your problems.

LLMs perform more effectively when dealing with highly specific, isolated issues.

From a technical standpoint, this might involve configuring multiple **SemanticKernels** differently and organizing them within a pipeline if needed.

2. RAG Storage. Memory Options

A typical RAG solution may have one or set of various components.



SK KernelMemory

Solutions like KernelMemory aim to simplify the different memory types management by abstracting the underlying complexities.

Azure Cognitive Search · Milvus · PostgreSQL

There are plenty of other solutions which support vectors.
I compared them in another post.

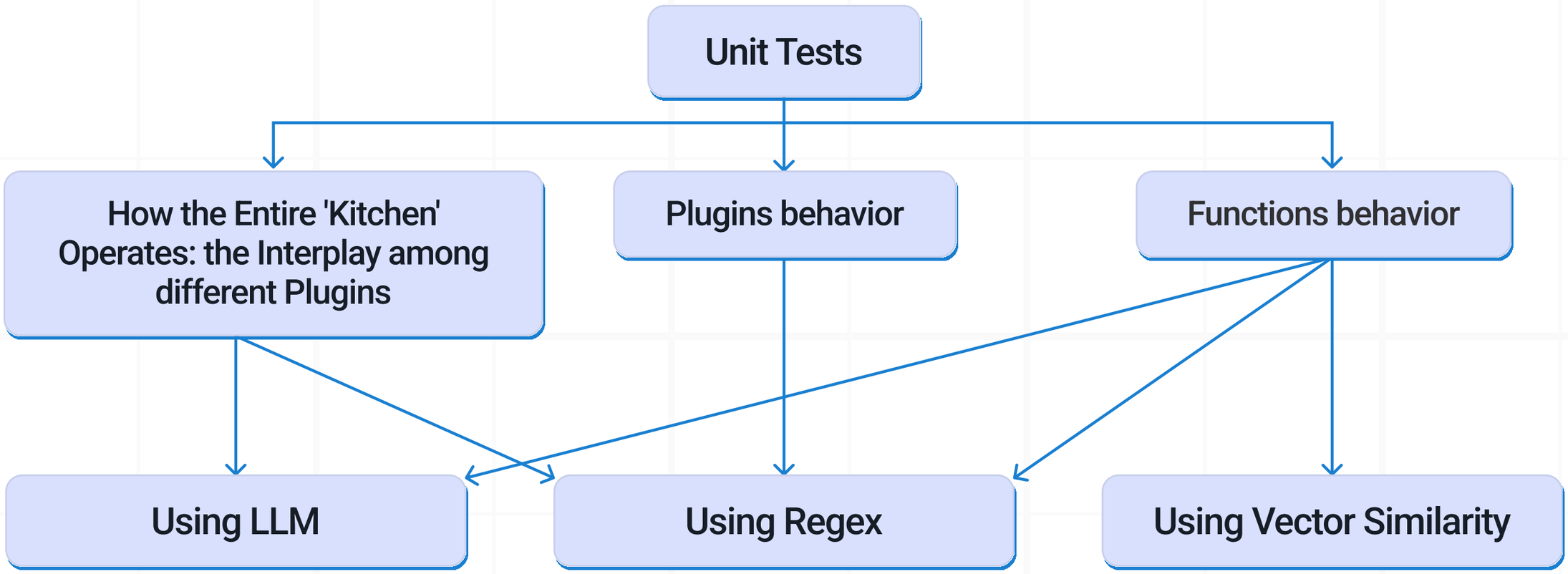
A RAG solution typically involves vector storage such as **Qdrant**, **Milvus**, **PostgreSql with pgvector**, or **AI Search,** and chat history.

However, it's crucial to recognize that along with data **without ontology** there are more sophisticated data structures available.

In sectors like Life Sciences, complex data analysis and mapping are often necessary, leading to the use of data with **ontology**.

3. Testing Strategies

Taking into account all previously discussed we can formulate what is necessary to test and how.



1. After introducing new SemanticKernel Plugin;
2. After changes in connectors and models;
3. After changes in memory;
4. New Language and Market specifics.

1. Answer Format;
2. Input and output validation;
3. After Context changes.

An interesting point to consider is the volume of tokens these tests could consume. With 20 plugins and 100 tests, you could easily use up to 300,000 tokens per run.

It might be surprising to find out that just the unit tests for local builds and PR requests could cost for relatively small team up to $100 per day!

4. Tackling Hallucinations. Practical approaches

All pieces of the puzzle take place.

And how to handle the hallucinations of the system?

Employ modern technique(s) such as: [1]

- Chain of Thoughts (CoT);
- Single-shot or Few-shot;
- System to Attention (S2A);
- FLARE;
- Tree-of-thoughts (ToT);
- Everything-of-thoughts (XoT).

Ensure to attach only significant volatile and persistent memory segments: [2]

- The size of persistent memory chunk should be ~1000 tokens as recommended by Meta;
- Include only necessary messages from Chat History;
- Add Session & User State when needed;
- Apply summarization technique to the chat history and document chunks.

Divide the complex task into steps and assign each step to specific agents.

Structure these agents work into a pipeline. [3]

Use Content Manager in PROD. [4]

Conduct automatic tests to verify that your application performs reliably within expected business scenarios. [5]

Hallucination isn't unique to LLMs.

It reflects the workings of the human brain. Therefore, strategies that are effective in real-life scenarios can also be applicable for LLMs.