

AI Assistant for IT Architecture Design

Presented by Neural Ninjas

- Brijesh Rai
- Divesh Yadav
- Yogesh Mishra
- Vaibhav Singh



Problem Statement

Main Challenge

Designing IT systems is hard work. It usually takes many tries and a lot of talking between the people who create the designs and everyone else involved.

Current Limitations

Today's methods use old, unmoving diagrams and complicated tools that require a lot of manual effort. They also don't have smart AI features to create or analyze designs automatically.

The Need

We need a smart AI tool that can create interactive design diagrams and suggest the best parts to use. This would make the design process much smoother and lead to better results.



Solution



AI Creates Diagrams

Our AI uses what you tell it to automatically draw architecture diagrams for you.



Easy Design Changes

You can chat with the AI to easily make changes and improve your designs.



Ways to Export

Get your diagrams as images or in editable formats like Graphviz.



Ready-Made Templates

Use our collection of built-in templates and best practice design examples.

- IMPROVEMENT
- DEVELOPMENT
- SOLUTION



Implementation



How We Build It

01

Getting Started

First, we'll connect our AI brain (the LLM) and set up the tools that draw the diagrams. We'll also create a collection of pre-made templates for quick use.

02

Building the Main Features

Next, we'll build the chat interface so you can easily talk to the system. We'll also teach it how to design different architectures and recognize common design patterns.

03

Finishing Up and Launching

Finally, we'll add ways for you to save and share your diagrams, make the app super easy to use, test everything carefully, and write simple guides to help you get started.

Tech Stack Used

Our solution is built with a set of modern and powerful tools. This helps us make sure it works fast, can grow easily, and is simple to keep running smoothly.

AI Brainpower

- OpenAI GPT-4.0
- LangChain

For Generating Diagrams

- Graphviz

Behind the Scenes (Backend)

- Python
- Flask

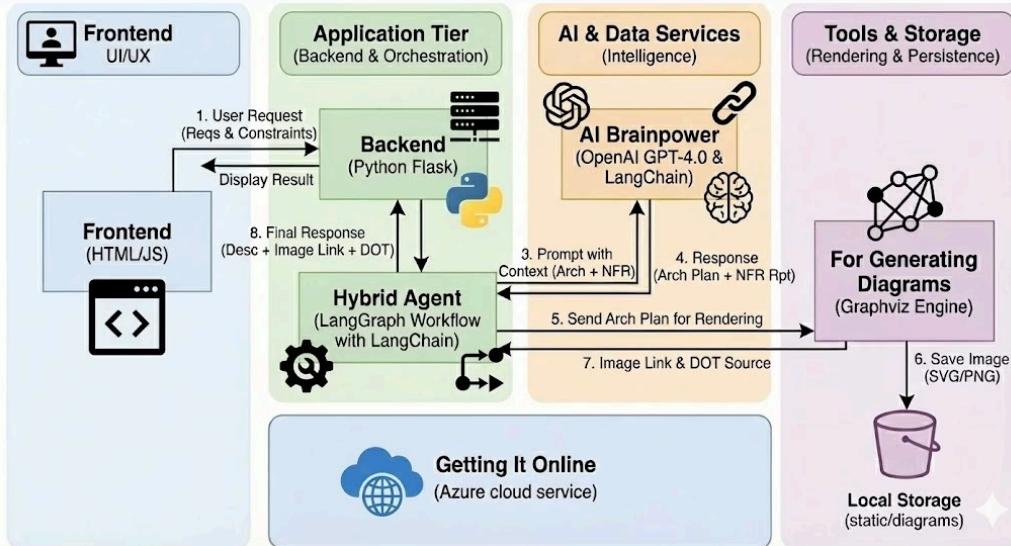
What You See (Frontend)

- HTML
- Javascript

Getting It Online

- Azure (cloud service to host it)

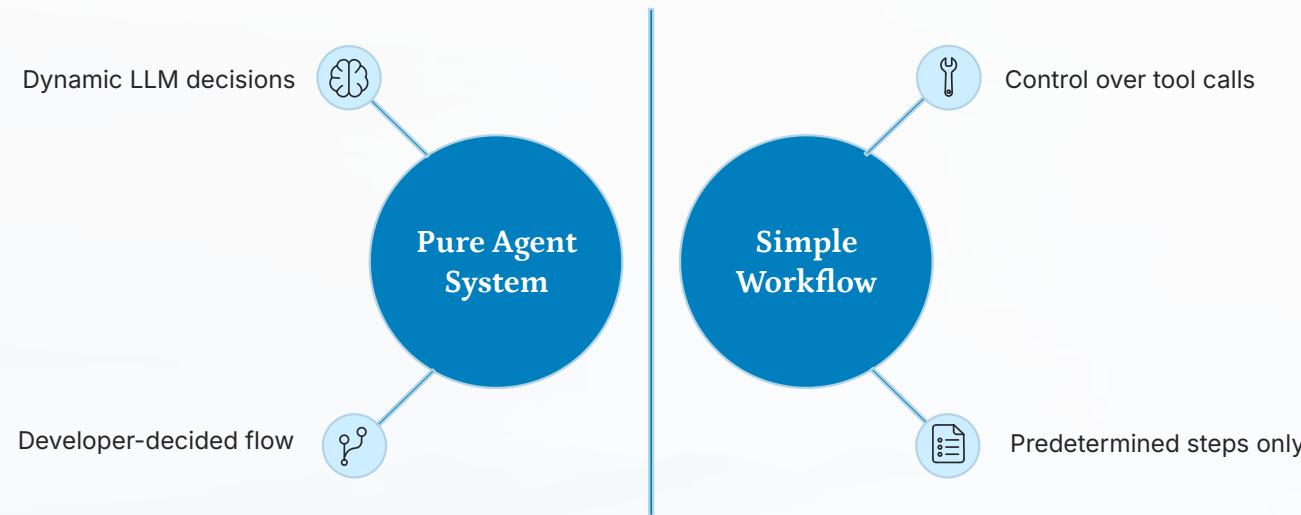
Solution Architecture



- User Interface (Input): The user enters their system requirements into the Web UI(HTML/JavaScript Frontend).
- API Gateway (Routing): The Python Flask server receives the user's message and passes it to the backend logic.
- Hybrid Agent (The Brain): LangGraph acts as the orchestrator, managing the workflow and deciding when to call the AI or the drawing tools.
- Intelligence (AI & Data): Azure OpenAI (GPT-4o) generates the architecture plan, using design patterns retrieved from the Vector Database.
- Visual Generation (Output): The agent sends the plan to Graphviz, which renders the final architecture diagram image for the user to view.

Our system has a strong, layered design that's easy to understand, works efficiently, and can grow easily. This means each part of the system handles its own specific job and talks to other parts in a clear, organized way.

Pure Agent & Simple Workflow



Pure Agent System (Agentic)

- LLM has full autonomy and control
- Decides when to call tools, which tools to use, and in what order
- Can call tools multiple times as needed
- Determines when reasoning is complete
- Follows ReAct pattern (Reason + Act)
- Process flow: LLM → Tool → LLM → Tool → LLM → ... → Final Answer

Simple Workflow

- Developer controls the entire flow
- Fixed, predetermined sequence: Step 1 → Step 2 → Step 3 → END
- No decision-making by the LLM
- No tool selection by the LLM
- Completely deterministic process
- Everything is pre-planned and scripted

Our Hybrid Approach: Best of Both Worlds



Why Hybrid?

Our system combines workflow structure with agent intelligence:

- Outer structure: Deterministic workflow (Architecture → Diagram → NFR Review)
- Inner logic: LLM acts as domain-specific mini-agent within each step
- LLM performs intelligent reasoning but doesn't choose tools dynamically

Advantages of Hybrid Approach

Pure Agents Have Challenges

Harder to control and debug

Prone to hallucinating tool usage

Can get stuck in loops

Less predictable timing

Our Hybrid Approach Delivers

✓ Stability and predictability

✓ Clear user experience

✓ Statefulness (LangGraph + MemorySaver)

✓ Two expert-agent steps (Architecture + NFR)

✓ Zero chance of agent loops



Conclusion

Our AI design tool helps you create architecture designs much faster through smart automation. It also makes sure your designs are high-quality by using the best methods, and it makes it easy for teams to work together. You can get your designs in various helpful formats, showing just how practical generative AI is for IT services. Ultimately, it brings big benefits to architecture design.

Thanks!
Any Questions? Let's Discuss!

