System Design

AI Agents System Design

How to build scalable agentic systems?

Follow
Bhavishya Pandit

Repost

Introduction

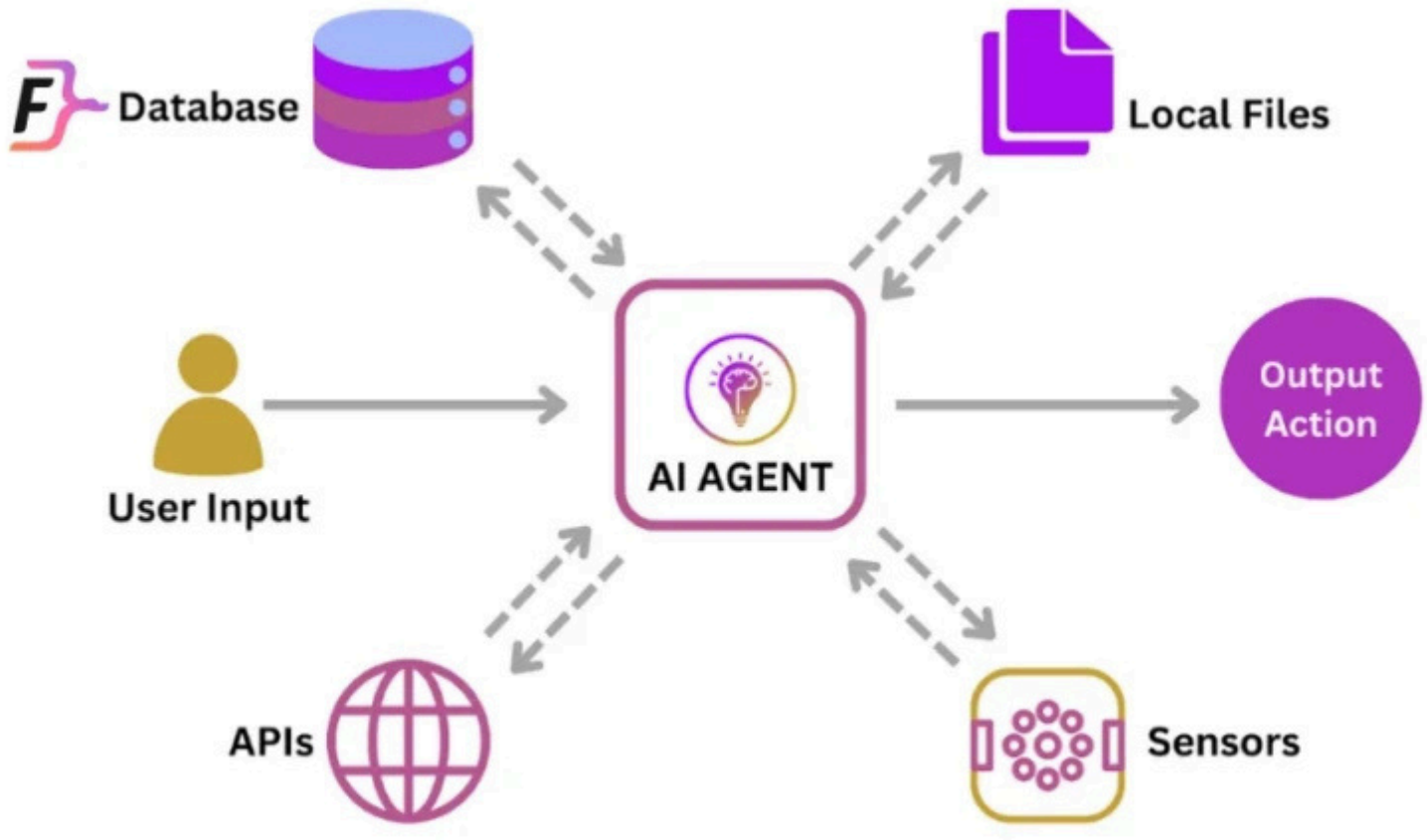
Modern AI apps are moving from single-turn Q&A models to systems that can reason, decide, and act.

All of this is possible due to **AI Agents**!

They enable AI to:

- Handle dynamic queries
- Fetch live data
- Take actions like updates or alerts

Source



Agentic design is not one-size-fits-all. Let's explore how it works.

Follow
Bhavishya Pandit

Repost

Core Components of Agentic Design

Large Language Models (LLMs) are powerful, but by default, they're limited to what they were trained on.

Agentic systems extend this by adding:

1 Memory

Context awareness (e.g., conversation history)

2 Tool calling

Executing code, querying databases, etc.

3 Decision-making

Reasoning before acting

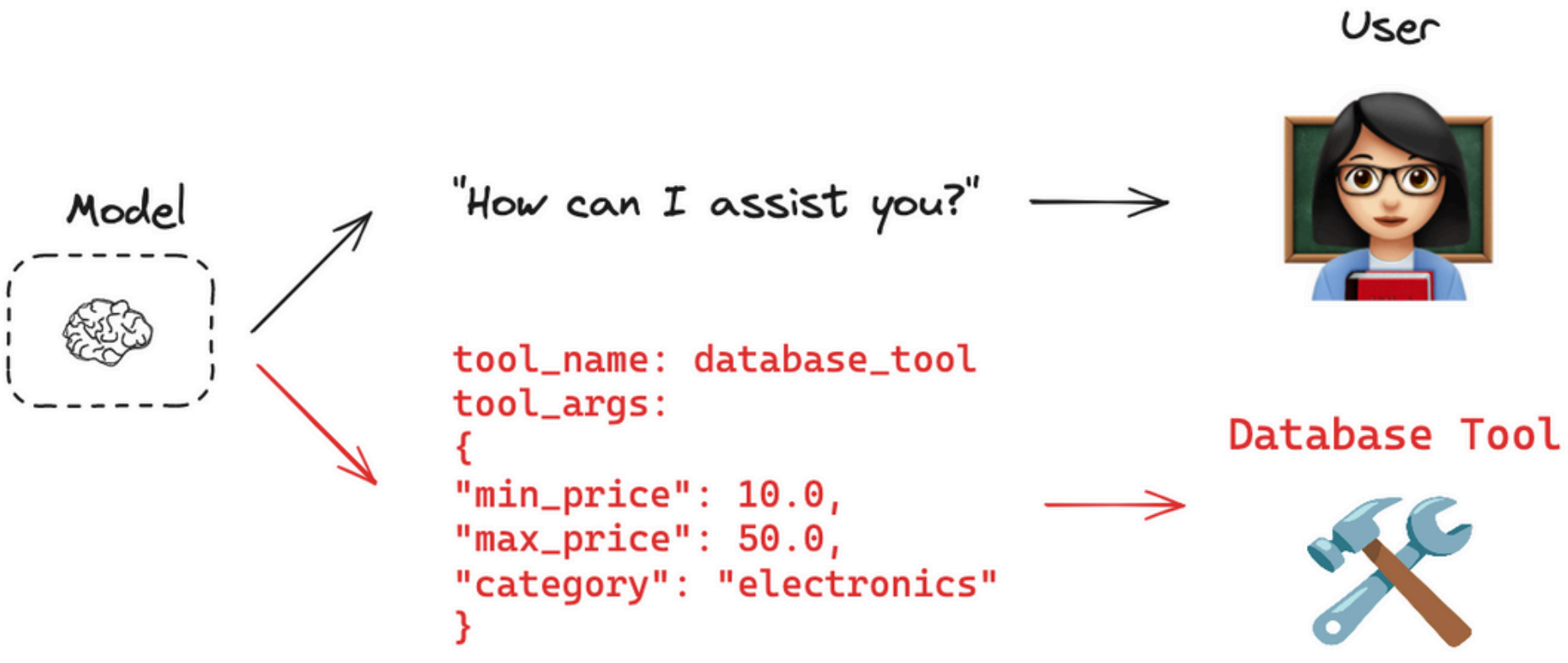
Think of an agent as a smart coordinator that combines LLM reasoning with external capabilities to solve real problems.

Tool Calling - Foundation of any agent

Tool calling allows agents to “talk” to the outside world.

This includes:

- **Data retrieval:** SQL queries, CRM lookups, vector search
- **Action-taking:** Send an email, update a record, run analytics
- **Custom logic:** Python functions, business rules, or APIs



Source

Without tool calling, agents remain passive. With it, they become interactive and useful.

Now lets talk about the different design systems of an agentic system ->

Follow
Bhavishya Pandit

Repost

1 Deterministic Chains

The most basic form of an agentic system is the deterministic chain.

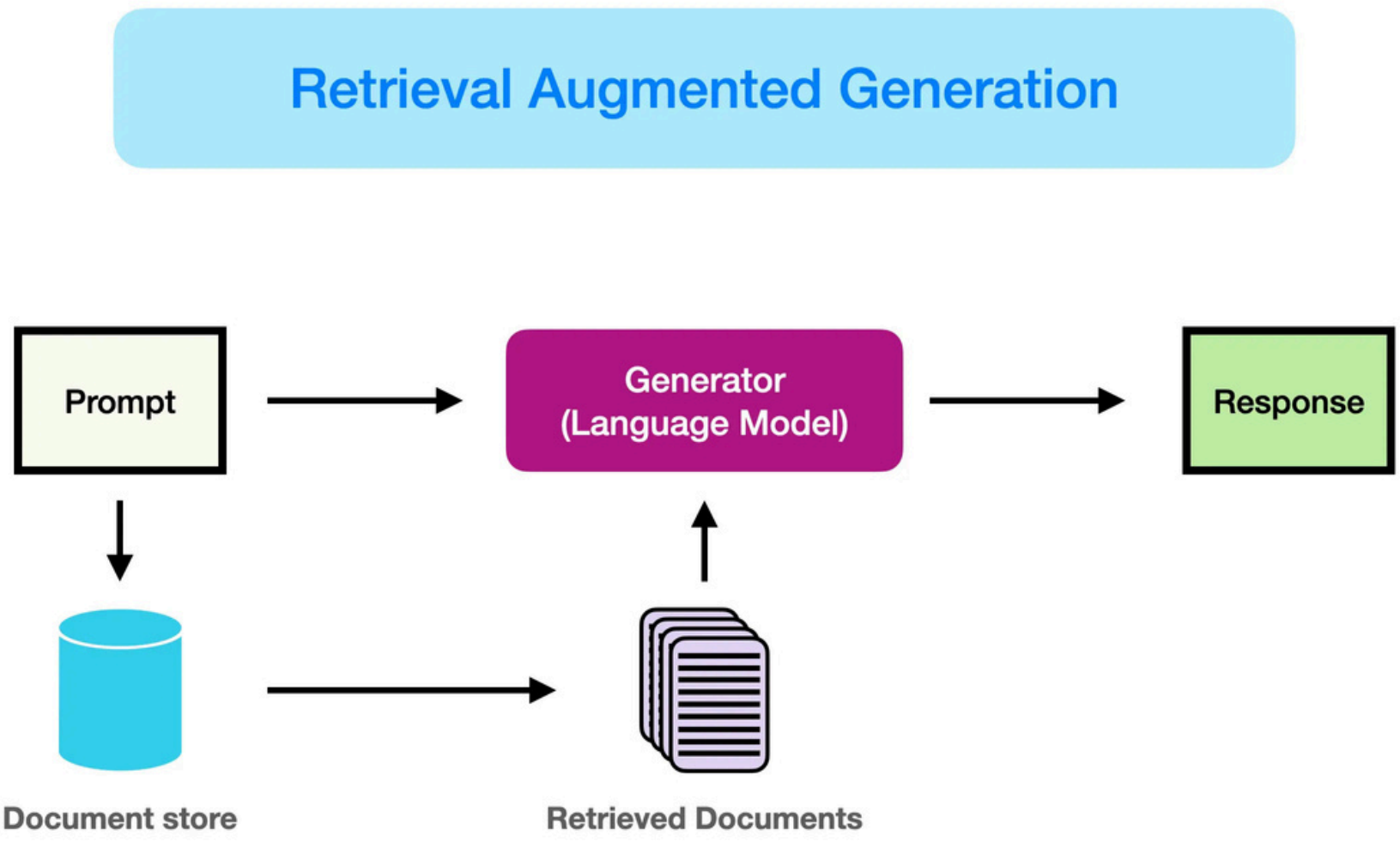
This design follows a fixed sequence of operations, where every step, tool call, and logic path is pre-defined by the developer.

✅ Every tool call happens in the same order

✅ The system behaves predictably for any given input

❌ No decisions, branching, or conditional logic

A common example is the Retrieval-Augmentation-Generation (RAG) workflow:



Follow
Bhavishya Pandit

Repost

2 Single-Agent Systems

In a single-agent system, one agent manages the entire flow — coordinating reasoning, tool usage, and responses in a dynamic and context-aware manner.

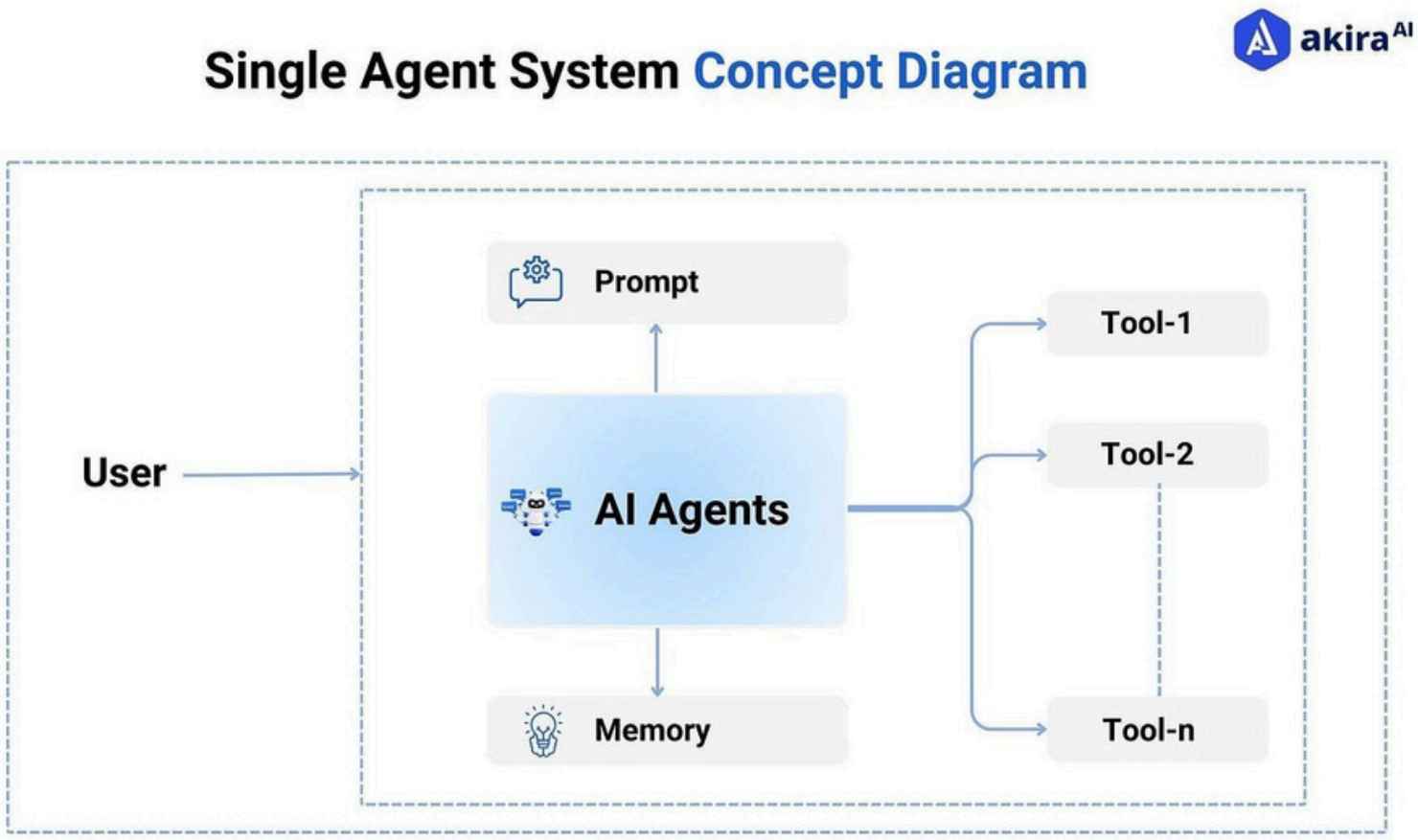
Unlike deterministic chains, the agent doesn't just follow a fixed script. It can:

✅ Accept inputs and relevant context (e.g., past conversations, user history)

✅ Decide when to call tools, and which tools to use

✅ Loop through reasoning steps — making multiple LLM/tool calls if needed to reach the desired outcome

Source



Follow
Bhavishya Pandit

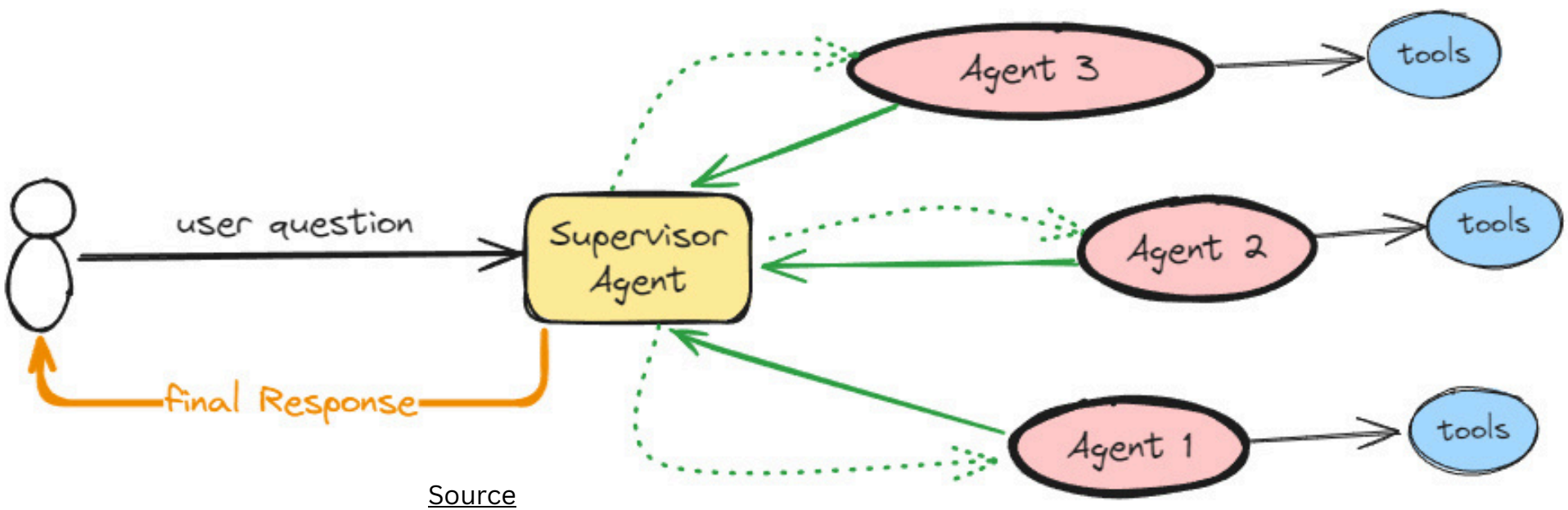
Repost

3 Multi-Agent Systems

In large-scale or cross-functional applications, a single agent may become overloaded or inefficient.

That's when a multi-agent system becomes essential.

Instead of one agent handling everything, the system is composed of multiple specialized agents, each responsible for a specific task or domain.



Source

Each agent:

✅ Focuses on a dedicated area (e.g., support, analytics, recommendations)

✅ Uses custom prompts, context, and tools relevant to its task

✅ Can collaborate or hand off to other agents when needed

Follow
Bhavishya Pandit

Repost

Choosing the Right Pattern

Not every AI application needs complex orchestration.

Choosing the right agent system design depends on your task complexity, domain diversity, and required flexibility.

Let's break it down:

PATTERN	WHEN TO USE	EXAMPLE USECASES
Deterministic Chain	- Fixed, repeatable task flows - High auditability is required - Low latency and predictable output - Logic rarely changes	FAQ bots Compliance workflows Basic RAG pipelines
Single-Agent System	- Varied queries within a single domain - Needs reasoning or retries - Seeks flexibility without complexity - Moderate sophistication	Helpdesk assistants Internal AI tools Smart form-fillers
Multi-Agent System	- Covers diverse business domains - Specialized logic/context per task - Needs modular, scalable architecture - Clear role separation	Enterprise AI platforms AI copilots AI-powered CRMs

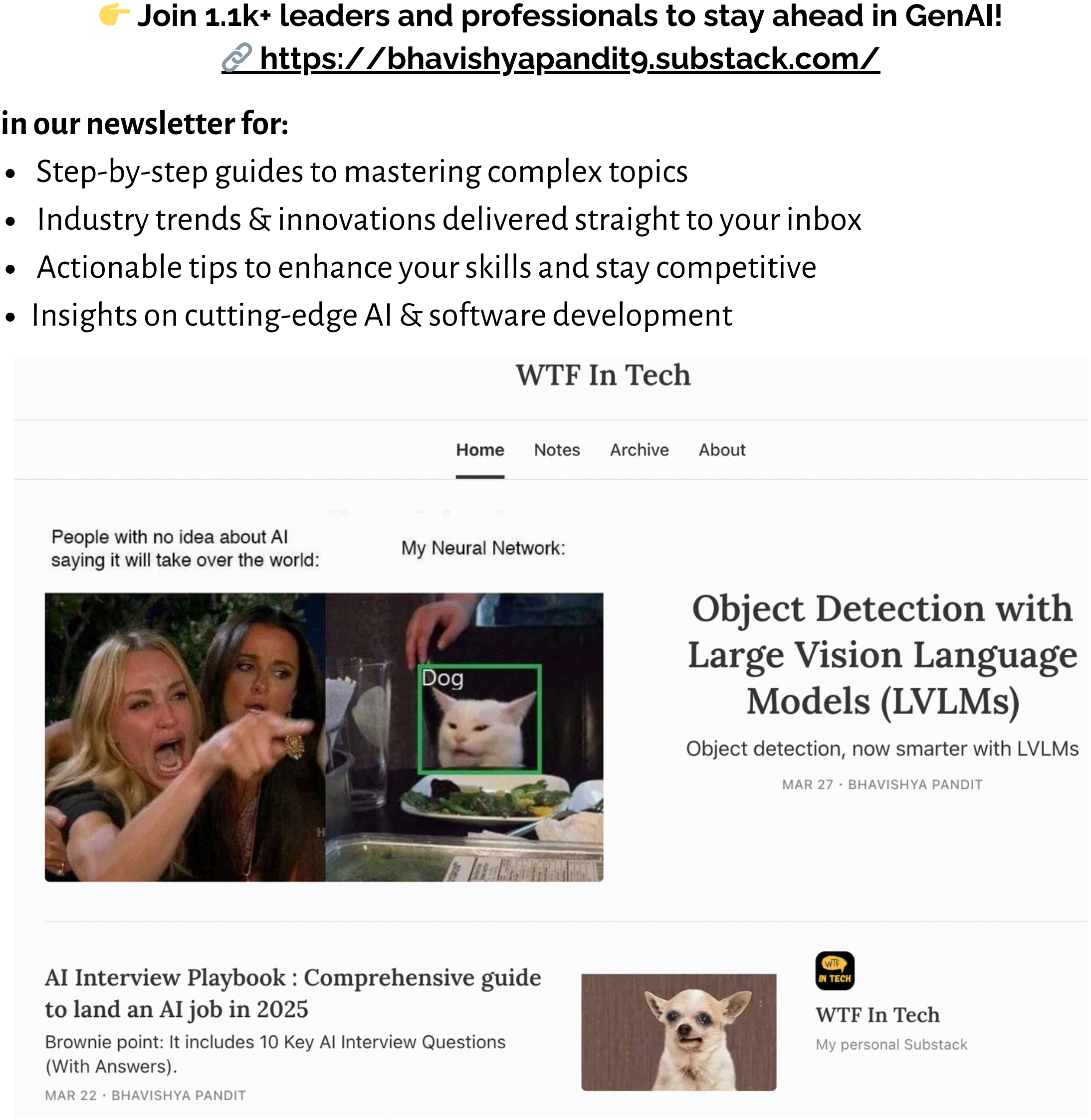
Stay Ahead with Our Tech Newsletter! 🚀

👉 **Join 1.1k+ leaders and professionals to stay ahead in GenAI!**

🔗 **https://bhavishyapandit9.substack.com/**

Join our newsletter for:

- Step-by-step guides to mastering complex topics
- Industry trends & innovations delivered straight to your inbox
- Actionable tips to enhance your skills and stay competitive
- Insights on cutting-edge AI & software development



💡 Whether you're a developer, researcher, or tech enthusiast, this newsletter is your shortcut to staying informed and ahead of the curve.

Bhavishya Pandit

Follow to stay updated on Generative AI

LIKE

COMMENT

REPOST