AI AGENT OBSERVABILITY

How telemetry works for AI Agents?

Bhavishya Pandit

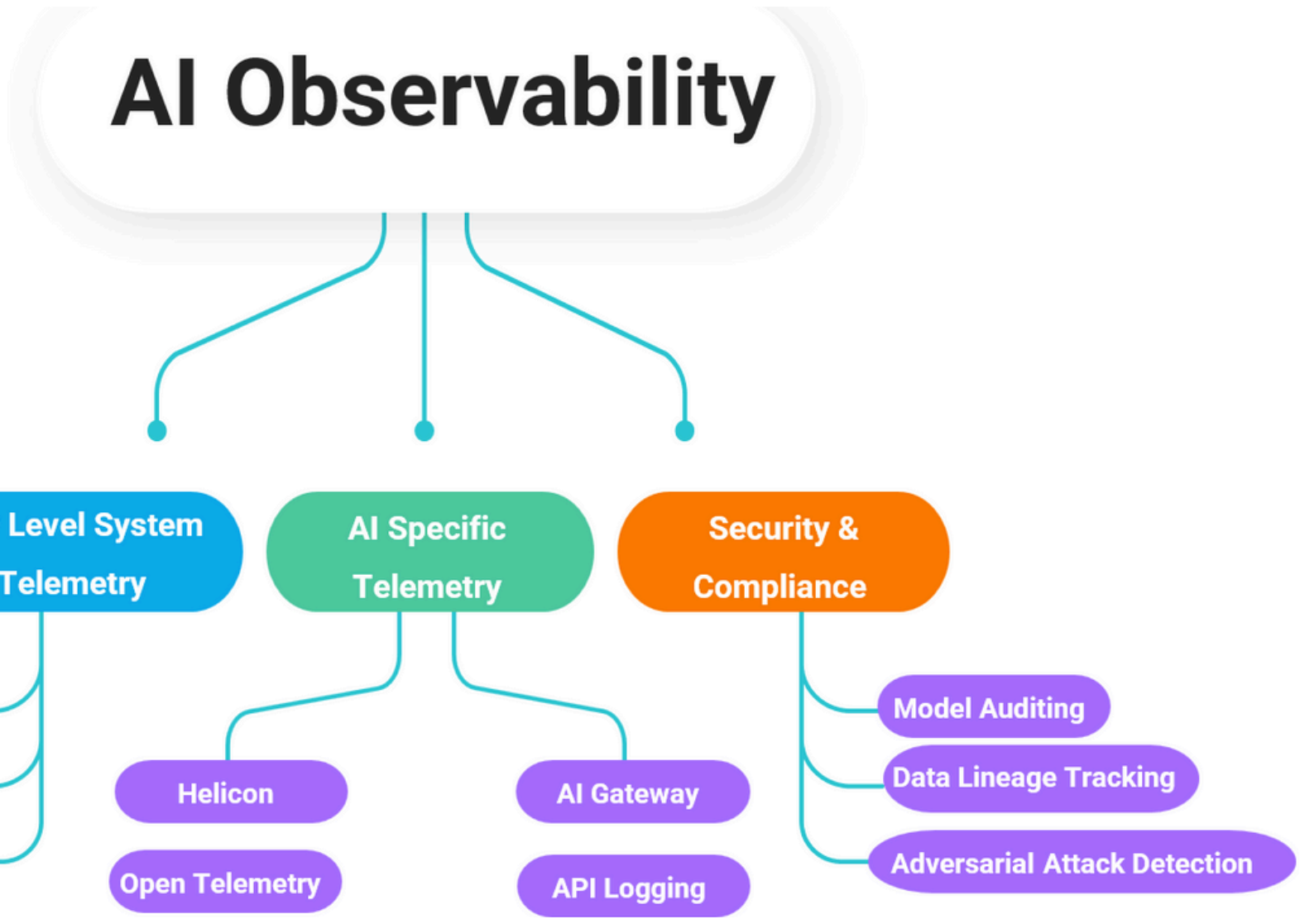
Introduction

63% of AI failures originate from undetectable reasoning errors (MIT 2024)

Agents fail differently than software:

- Silent tool failures (APIs return **200** but wrong data)
- Logic loops (wasting **$1000s** in LLM calls)
- Context drift (**forgetting** original user intent)

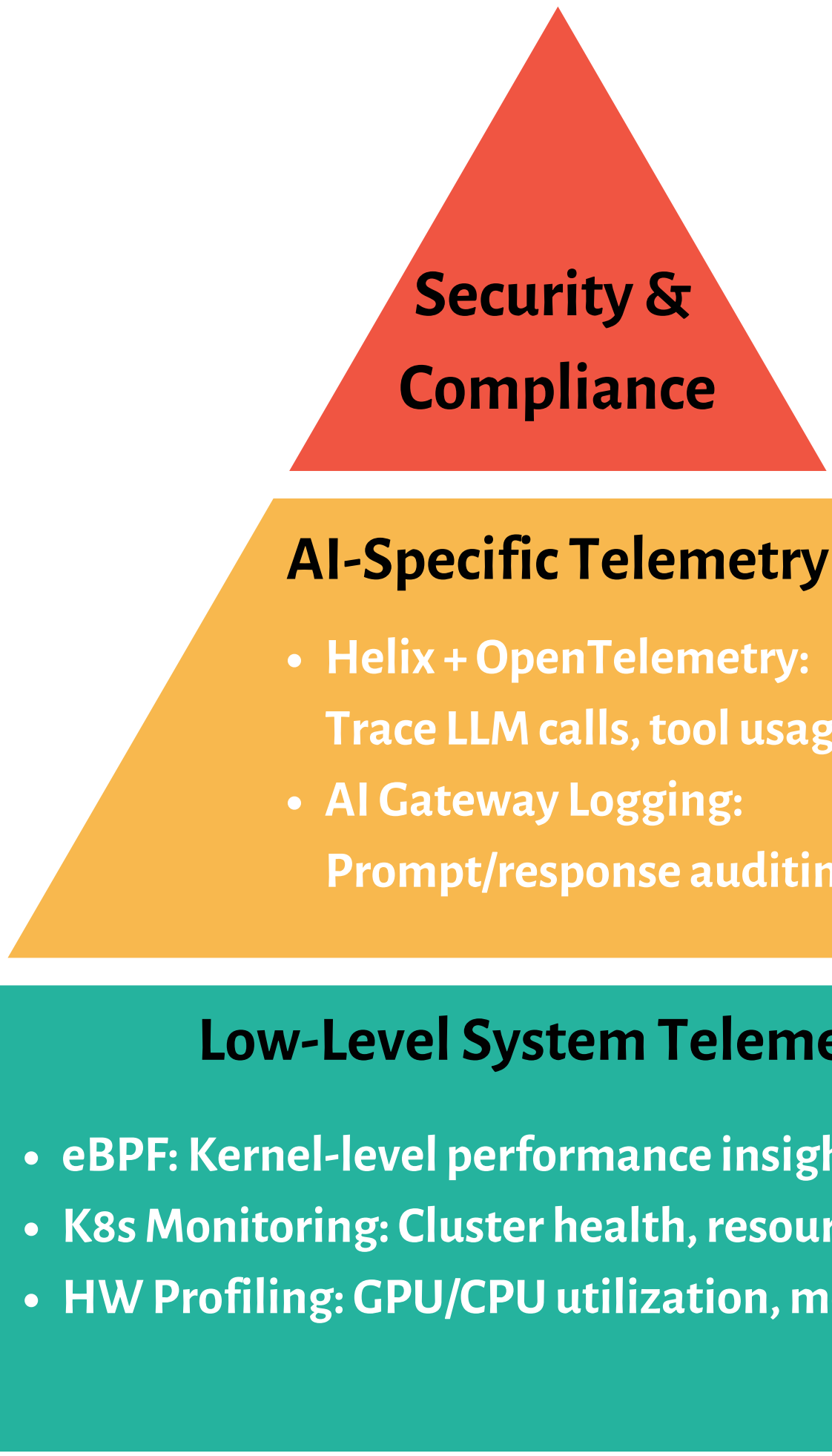
Traditional monitoring shows surface-level metrics like errors and latency, but completely misses the why behind your agent's actions. When something goes wrong, you're left guessing.



Without full observability into every step - prompts, tool calls, memory changes, and decision paths - you're essentially debugging blindfolded. Let’s discuss about observability of AI Agents in detail-

AI Observability Stack

We can break the complete observablity stack in 3 layers -



- Model Auditing
- Data Lineage
- Spot prompt injections, jailbreaks

What To Track?

Observability for Agents differs from standard practice, we need to implement special metrics for better understanding.

Cost Per Run

- Token usage
- API calls
- Tool expenses

Real User Signals

- Thumbs-up/down
- Behavioral Cues

Accuracy

- Agent task completion accuracy
- Agent Obedience

Automated Safeguards

- LLM/Agent Guardrails
- RAGAS (Evaluation Library)

LLM/Agent as a judge

Another Agent/LLM to rate performance of or LLM responses

Offline and Online Evaluation

AI Agents are evaulated in two categories that complement each other-

OFFLINE EVALUATION	ONLINE EVALUATION
Controlled tests with perfect questions	Real users ask unpredictable things
✅ Verifies basic competency ✅ Catches glaring errors ✅ Sets benchmarks	✅ Reveals user perception ✅ Uncovers edge cases ✅ Detects slow degradation
Limitation: Only tests what you anticipate	**Challenge:** No "right answers" - harder to measure
Best for: - Pre-launch safety checks - CI/CD pipelines - Compliance validation	**Best for:** - Continuous improvement - Model drift detection - A/B testing new versions

3 Step Agent Observability

A simple 3 step process to add observability to your AI Agent workflow:

1

- **Build test sets that mirror real user goals**
- **Include "adversarial examples" (ways users might break it)**
- **Automate testing in your CI/CD pipeline**

This will help in finding flaws in controlled environment before actual launch.

2

- **First 2 weeks: Run in shadow mode**
- **Compare new vs old version outputs**
- **Look for differences, not just errors**

This step compares real-world behavior without risking users

3

- **Weekly: Add top production failures to test sets**
- **Monthly: Review evaluation criteria (user needs evolve)**
- **Quarterly: Stress-test with fresh edge cases**

This continuously tests and refines your AI agent to systematically eliminate flaws and enhance performance.

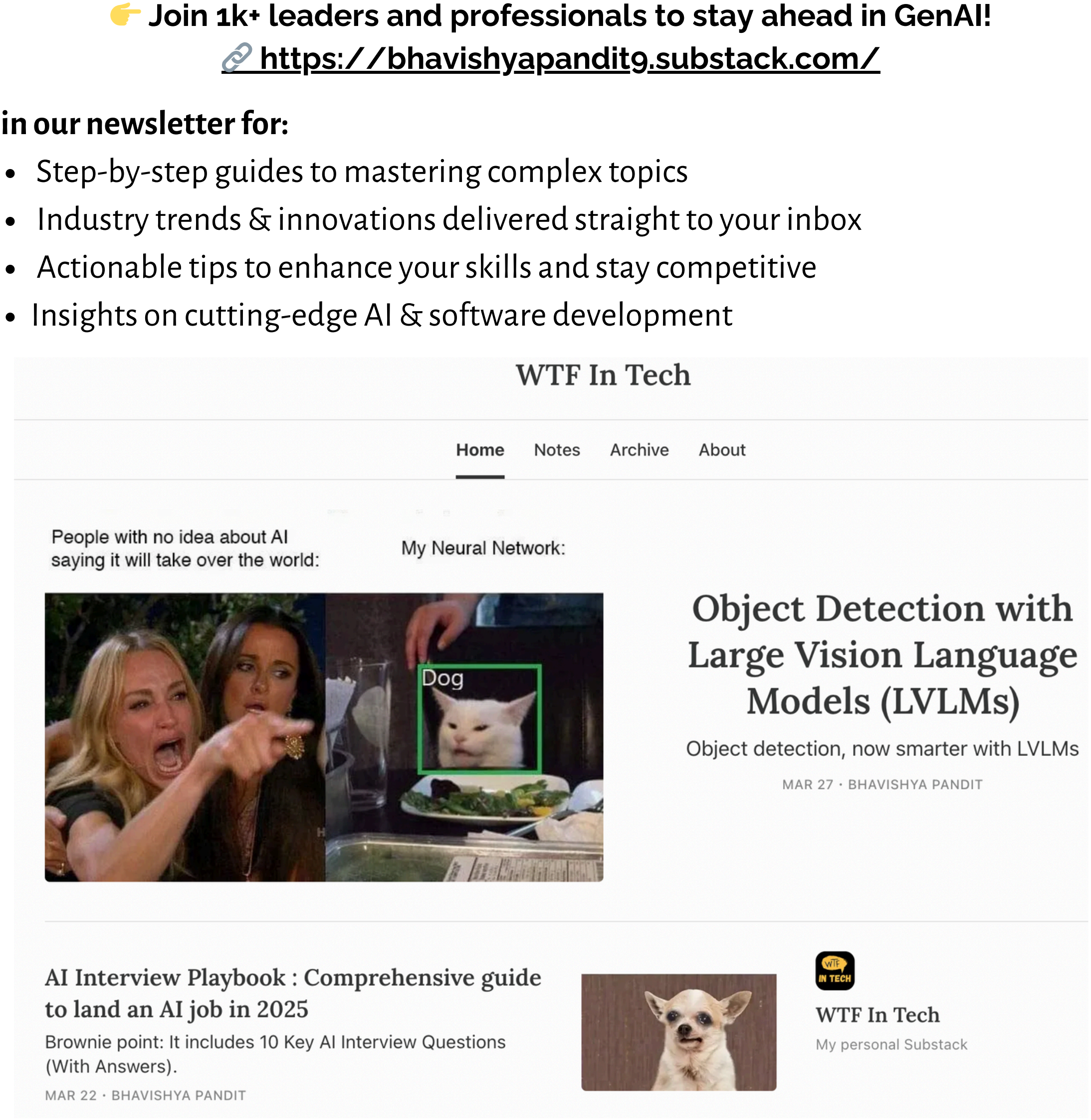
Stay Ahead with Our Tech Newsletter! 🚀

👉 **Join 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