

12 Production Agent Systems

All Failed at the Same Layer

Production Failure Analysis: Layer 5 (Orchestration)



Get the code: community.nachiketh.in

The 7-Layer Agent Architecture

LAYER 1

System prompts, few-shot examples, context

LAYER 2

Function calling, tool definitions

LAYER 3

Vector databases, retrieval, context management

LAYER 4

Planning, decision-making, execution

LAYER 5

⚠️ Orchestration (THE MISSING LAYER)

Retry logic • Idempotency • Circuit breakers • Timeout management

LAYER 6

Logging, metrics, tracing, debugging

LAYER 7

API, web interface, integrations

11 out of 12 systems skipped Layer 5 entirely



Get the code: community.nachiketh.in

What Layer 5 Actually Does

✗ What Most Teams Skip

- No retry logic for failed API calls
- No idempotency for duplicate requests
- No circuit breakers for cascading failures
- No timeout management

✓ What Production Needs

- **Retry Logic:** Exponential backoff + jitter
- **Idempotency:** Prevent duplicate actions
- **Circuit Breakers:** Fail fast after threshold
- **Timeouts:** Aggressive (2-5s) with retries

Result:



Get the code: community.nachiketh.in

Failure #1: The Slack Message Storm

1,200

DUPLICATE MESSAGES

6

HOURS

Company:

Agent Task:

The Issue:



Switch to VS Code → Show wrong implementation



Get the code: community.nachiketh.in

Failure #2: The Stripe Duplicate Charge Disaster

847

DUPLICATE CHARGES

\$84,000

DUPLICATE AMOUNT

Company:

Agent Task:

The Issue:



Switch to VS Code → Show Stripe implementation



Get the code: community.nachiketh.in

Failure #3: The Database Write Cascade

3,400

DUPLICATE TICKETS

1

DAY

Company:

Agent Task:

The Issue:



Switch to VS Code → Show race condition code



Get the code: community.nachiketh.in

The Three Patterns You Need

1

Wait longer after each failure
+ Random jitter

2

Same key = same action
Deduplicated automatically

3

Fail fast after threshold
Prevent cascading failures



Switch to VS Code → Implementation deep dive



Get the code: community.nachiketh.in

Layer 5 Separates POC from Production

POC/Demo Systems

- Work in testing
- Fail in production
- No error handling
- Expensive demos

Production Systems

- Handle failures gracefully
- Prevent duplicate actions
- Fail fast when needed
- Ship to customers

The gap isn't the LLM. It's the infrastructure.



Get the code: community.nachiketh.in

Get the Complete Implementation



Download All Code Examples + PDF Guide

Join the community to get instant access to:

- All Python code from this video
- ProductionOrchestrator class (complete)
- PDF guide with implementation notes
- Production patterns library

[Join Community \(Free\)](#)



Get the code: community.nachiketh.in

Build Production Systems, Not Demos

Agentic AI Enterprise Bootcamp

Learn to build **all 7 layers** of production agent systems

What you'll build:

- production-grade agent systems (deployed)
- Complete orchestration layer (Layer 5)
- Observability & monitoring (Layer 6)
- Cost optimization & scaling strategies

For: Working Professionals with Coding Experience in Python, Senior ML Engineers, Software Architects, Tech Leads (3+ years experience)

Starts: February 15, 2026 • 8 Weeks

[Apply for Bootcamp](#)

Speaker notes



Get the code: community.nachiketh.in