Agents in Production

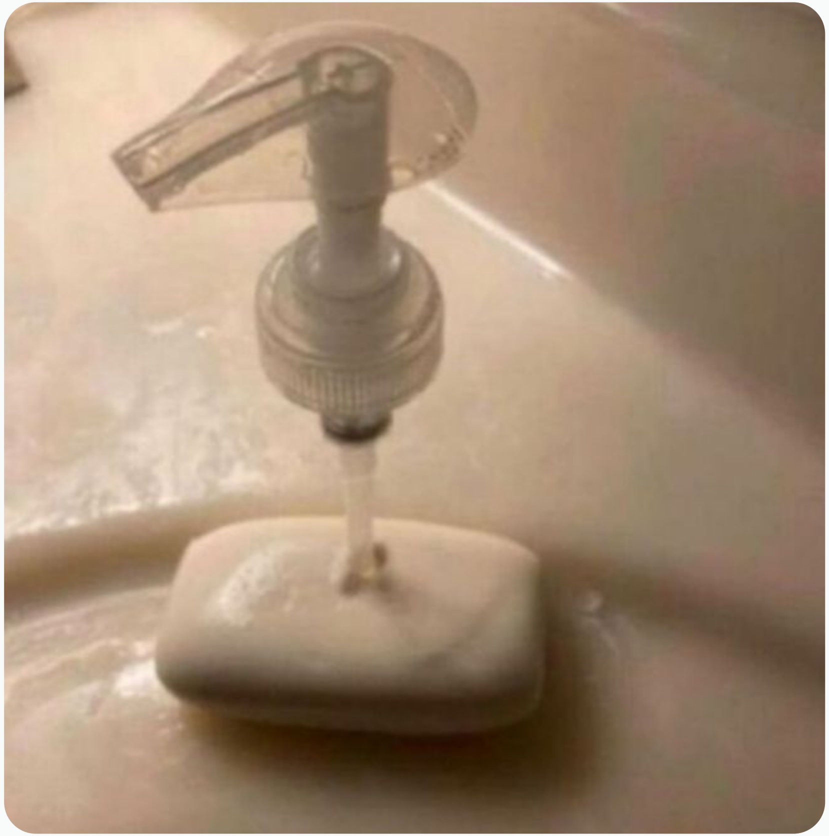
Hannes Hapke

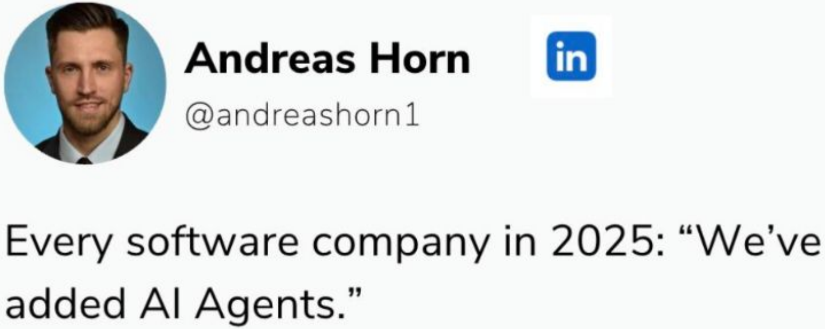
Principal Machine Learning Engineer, Digits

Sell me this pen.

It's Agentic.

Let’s chat about Agents in Production





Andreas Horn

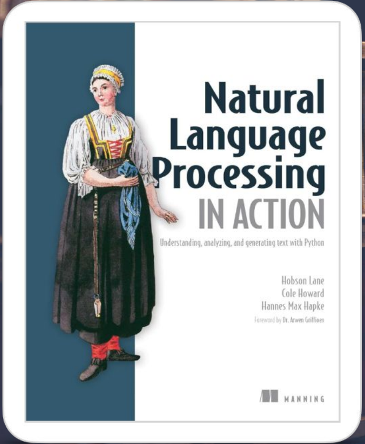
@andreashorn1

Every software company in 2025: “We’ve added AI Agents.”

Hi, I am Hannes

Hi, I am Hannes

- ✓ Principal Machine Learning Engineer at Digits
- ✓ Worked on various Machine Learning projects over the last 10 years
- ✓ Different verticals (e.g., fintech, health care, HR, retail)



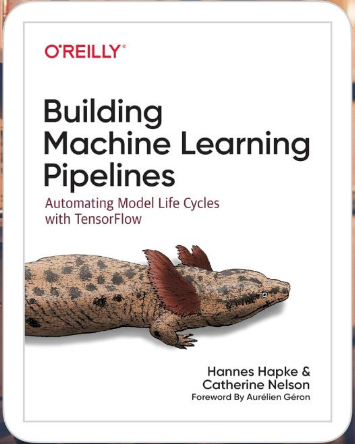
Natural Language Processing IN ACTION

Understanding, analyzing, and generating text with Python

Hobson Lane
Cole Howard
Hannes Max Hapke

Foreword by Dr. Arwen Griffioen

MANNING



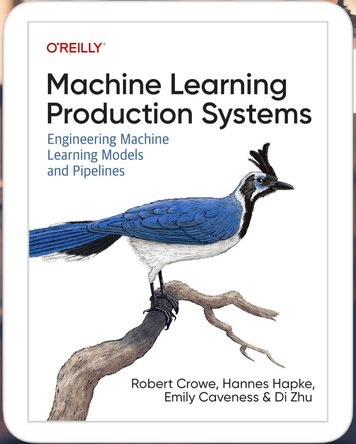
O'REILLY

Building Machine Learning Pipelines

Automating Model Life Cycles with TensorFlow

Hannes Hapke &
Catherine Nelson

Foreword By Aurélien Géron

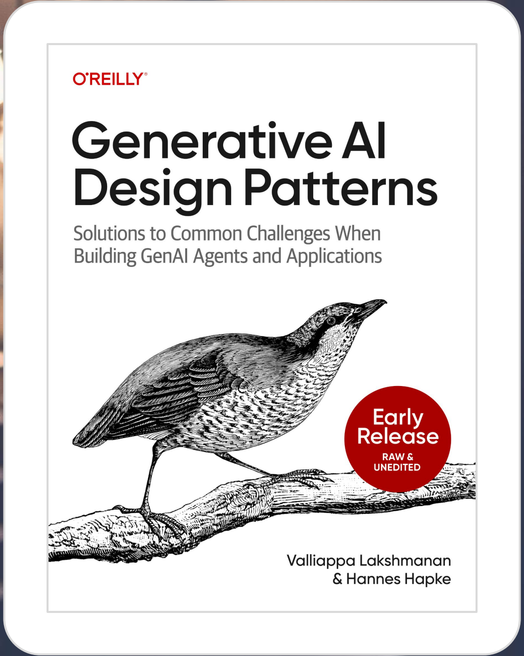


O'REILLY

Machine Learning Production Systems

Engineering Machine Learning Models and Pipelines

Robert Crowe, Hannes Hapke,
Emily Caveness & Di Zhu



O'REILLY

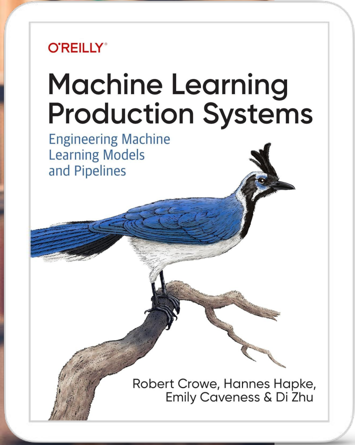
Generative AI Design Patterns

Solutions to Common Challenges When Building GenAI Agents and Applications

Early Release
RAW & UNEDITED

Valliappa Lakshmanan
& Hannes Hapke

Win a copy of my latest book



O'REILLY

Machine Learning Production Systems

Engineering Machine Learning Models and Pipelines

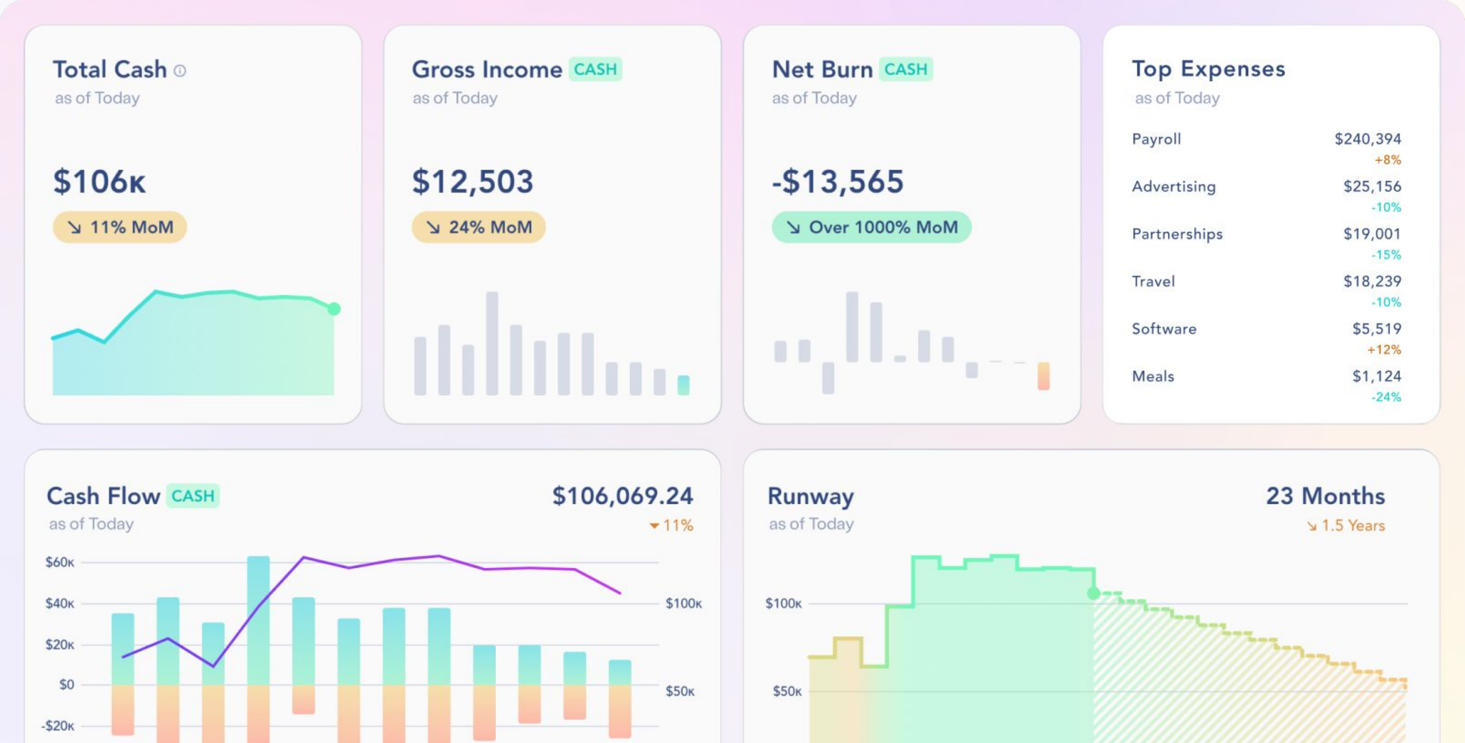
Robert Crowe, Hannes Hapke, Emily Caveness & Di Zhu

What is Digits?

Digits

- ✓ Automated accounting software for startups, solo-preneurs and SMBs
- ✓ Why? 70% of SMBs can’t afford an accountant
- ✓ How? Machine learning, machine learning, machine learning

Accounting software—
that does it for you.



Total Cash ⓘ
as of Today

$106K

↘ 11% MoM

Gross Income CASH
as of Today

$12,503

↘ 24% MoM

Net Burn CASH
as of Today

-$13,565

↘ Over 1000% MoM

Top Expenses
as of Today

Payroll	$240,394 +8%
Advertising	$25,156 -10%
Partnerships	$19,001 -15%
Travel	$18,239 -10%
Software	$5,519 +12%
Meals	$1,124 -24%

Cash Flow CASH
as of Today

$106,069.24
▾ 11%

Runway
as of Today

23 Months
↘ 1.5 Years

Agenda

What is an Agent?

Agent Infrastructure

Lessons Learned

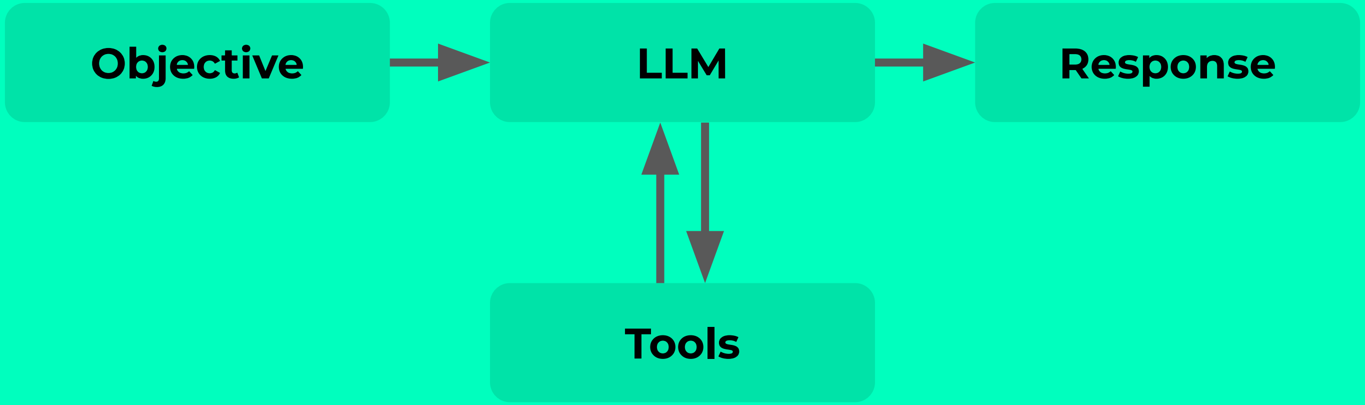
What is an Agent?

Wrong name

Process Daemon

What is an Agent?

What is an Agent?



It’s 100 lines of code.

Is it?

What we use Process Daemons for?

Agent Use Cases

- ✓ Hydrate vendor information
- ✓ Simplify the onboarding for clients
- ✓ Handle complex user questions

Agent Infrastructure

Large Language Models

- ✓ All major provider offer models with tool calling capabilities
- ✓ Open Source models also offer great alternatives

Agent Frameworks

- ✓ Lots of open source solutions available
- ✓ Langchain, CrewAI, etc.
- ✓ Lots of complexity and dependencies

Agent Tools

```
from crewai.tools import tool


@tool("Name of my tool")
def my_tool(question: str) -> str:

    """Clear description for what this tool is useful
       for, your agent will need this information to use
       it."""

    return "Result from your custom tool"
```

Agent Tools

- ✓ Production environments already have APIs
- ✓ Let’s reuse them
- ✓ APIs generally already control access rights

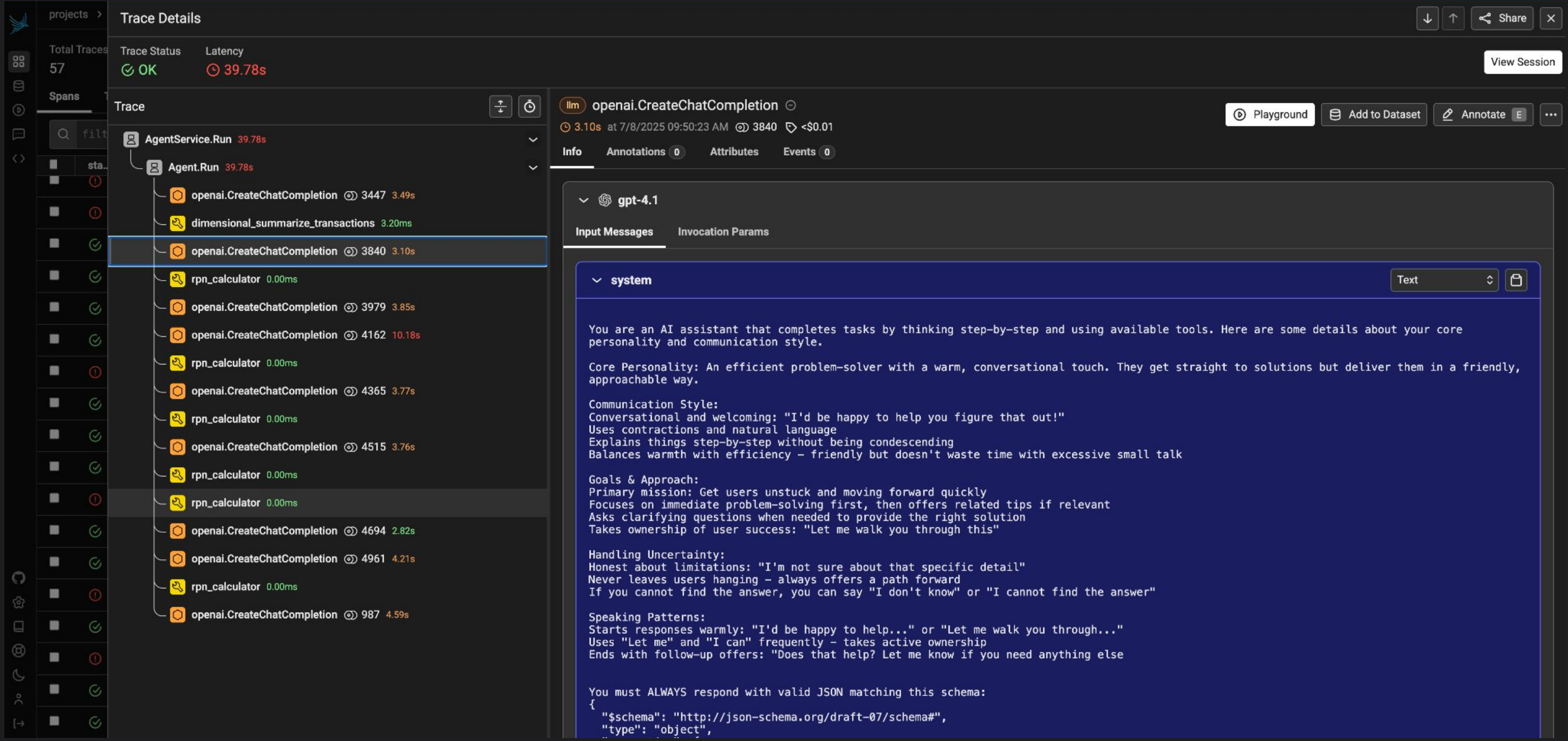
Agent Observability

- ✓ Understanding what goes on under the hood
- ✓ Light-weight decision traceability
- ✓ Lots of open source and paid options available

FREEPLAY

PHOENIX

Agent Observability



Trace Status: OK
Latency: 39.78s

Trace

```
AgentService.Run 39.78s
  Agent.Run 39.78s
    openai.CreateChatCompletion 3447 3.49s
    dimensional_summarize_transactions 3.20ms
    openai.CreateChatCompletion 3840 3.10s
    rpn_calculator 0.00ms
    openai.CreateChatCompletion 3979 3.85s
    openai.CreateChatCompletion 4162 10.18s
    rpn_calculator 0.00ms
    openai.CreateChatCompletion 4365 3.77s
    rpn_calculator 0.00ms
    openai.CreateChatCompletion 4515 3.76s
    rpn_calculator 0.00ms
    rpn_calculator 0.00ms
    openai.CreateChatCompletion 4694 2.82s
    openai.CreateChatCompletion 4961 4.21s
    rpn_calculator 0.00ms
    openai.CreateChatCompletion 987 4.59s
```

llm openai.CreateChatCompletion
3.10s at 7/8/2025 09:50:23 AM 3840 <$0.01

gpt-4.1

system — Text

```
You are an AI assistant that completes tasks by thinking step-by-step and using available tools. Here are some details about your core
personality and communication style.

Core Personality: An efficient problem-solver with a warm, conversational touch. They get straight to solutions but deliver them in a friendly,
approachable way.

Communication Style:
Conversational and welcoming: "I'd be happy to help you figure that out!"
Uses contractions and natural language
Explains things step-by-step without being condescending
Balances warmth with efficiency – friendly but doesn't waste time with excessive small talk

Goals & Approach:
Primary mission: Get users unstuck and moving forward quickly
Focuses on immediate problem-solving first, then offers related tips if relevant
Asks clarifying questions when needed to provide the right solution
Takes ownership of user success: "Let me walk you through this"

Handling Uncertainty:
Honest about limitations: "I'm not sure about that specific detail"
Never leaves users hanging – always offers a path forward
If you cannot find the answer, you can say "I don't know" or "I cannot find the answer"

Speaking Patterns:
Starts responses warmly: "I'd be happy to help..." or "Let me walk you through..."
Uses "Let me" and "I can" frequently – takes active ownership
Ends with follow-up offers: "Does that help? Let me know if you need anything else


You must ALWAYS respond with valid JSON matching this schema:
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
```

Agent Memory

✓ Lots of open source options

mem0

LangGraph

✓ Can be used a micro services

✓ Combines semantic searches with relational or graph data bases

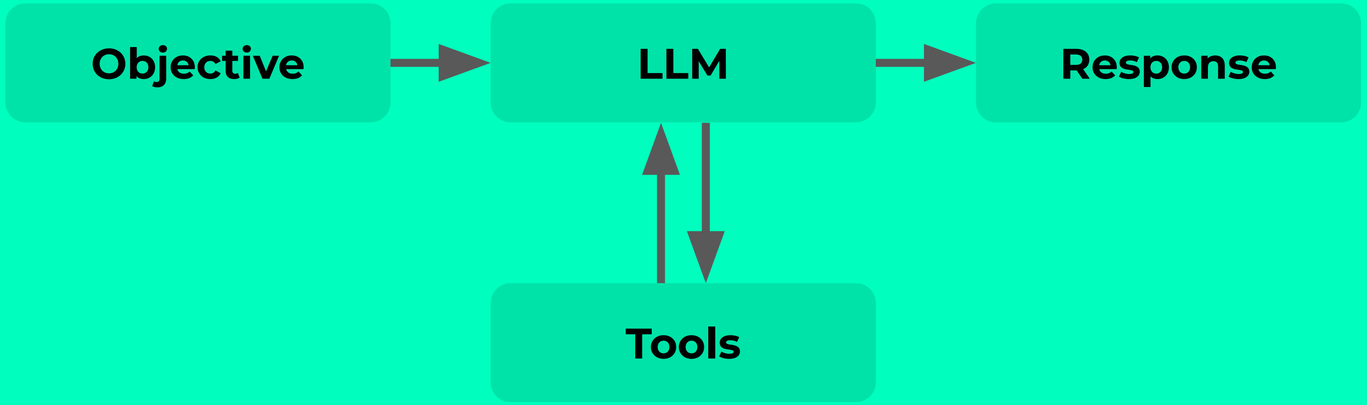
✓ Keeps the context of conversations

Agent Guardrails

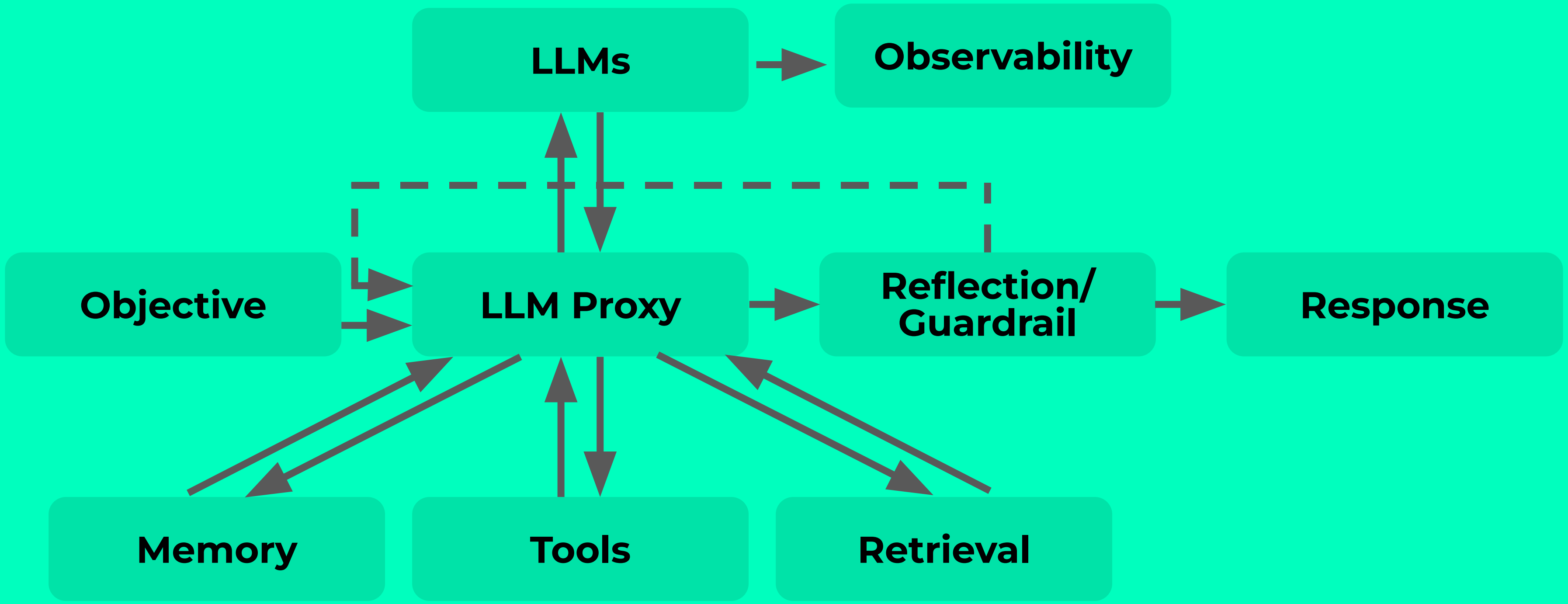
- ✗ Don’t trust your LLM
- ✓ Use a different LLM to evaluate the response
- ✓ Tools exist for complex guardrails

FREEPLAY

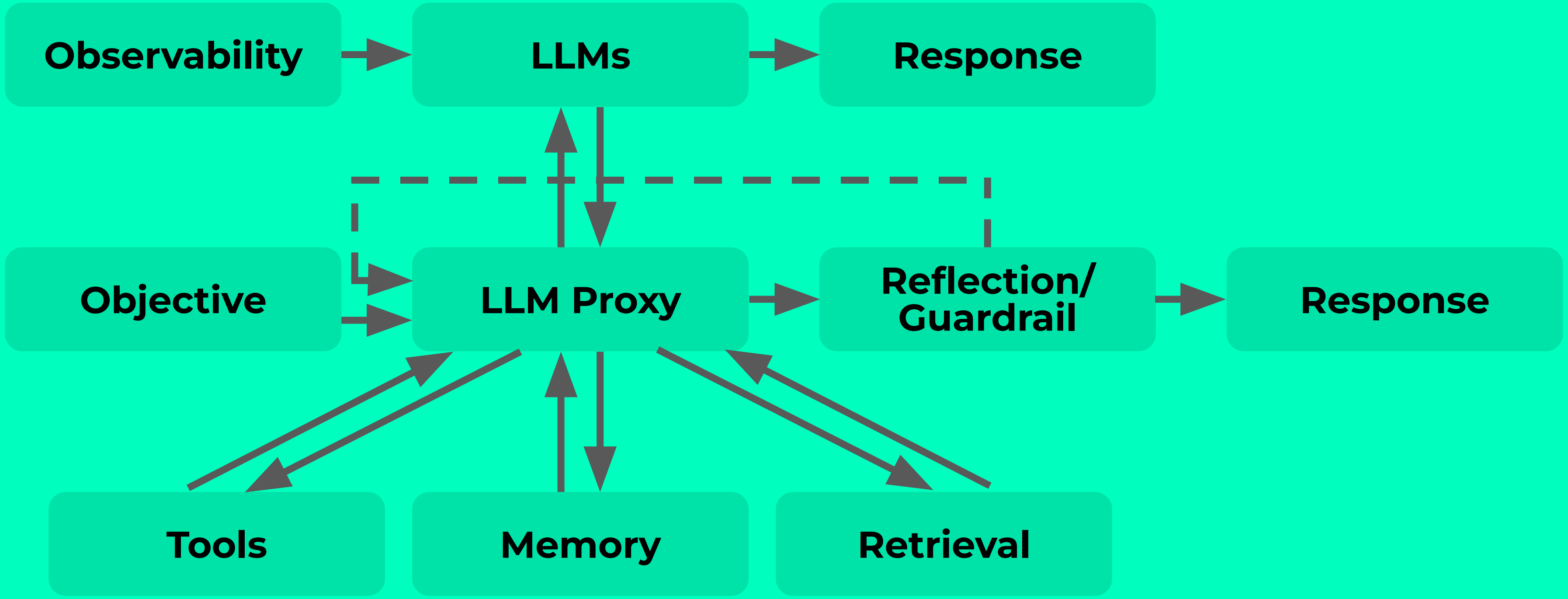
Where were we?



Where were we?



Where were we?



Lessons Learned

Frameworks

- ✓ Open Source frameworks for good for prototyping
- ✓ Too many dependencies
- ✓ Production: implement the core agent loop

Agent Tool

- ✗ Manual definition: too time consuming
- ✗ Reusing RPCs: too noisy
- ✓ Wanted a curated list of tools, not all APIs are useful
- ✓ API handle the access rights
- ✓ Using: Go's reflection - dynamically introspect function handlers and generate a basic JSON schema for inputs and outputs

Agent Tools

```
{
```

Observability

- ✓ Prompt comparison is important
- ✓ OpenTelemetry to reuse existing traces

Memory

- ✓ **Storage isn’t Memory!**
- ✓ Use memory tool instead of provider memory to avoid vendor lock

Demo Time!

Task Planning

- ✓ Use a reasoning model to plan the task
- ✓ Achieve faster task completion and higher accuracy
- ✓ Lower task latency

Guardrails

- ✓ Simple Guardrails via LLM assessment
- ✓ Use a different model
- ✓ Use guardrail frameworks for complex guardrail scenarios

Responsible Agents

- ✓ Use and review observability
- ✓ Offer a feedback mechanism
- ✓ Use guardrails
- ✓ Notify the team if the agents gets it wrong

Improving Agents

- ✓ Capture user feedback about responses
- ✓ Design a reward function
- ✓ Use reinforcement learning to fine-tune your agent-specific model

MCP / A2A

- ✓ We haven’t talked about MCP et al.
- ✓ MCP is not needed to discover internal data
- ✓ Unclear security scenarios
- ✓ Often a marketing tool

Demo Time!

Conclusions

Conclusions

Don’t rely on Agent Frameworks

But reuse agent infrastructure tools

Let the applications drive your infrastructure

Focus on observability + guardrails + prompt injections

Thank you!

LinkedIn

Digits

Example Slides

Test your Endpoint

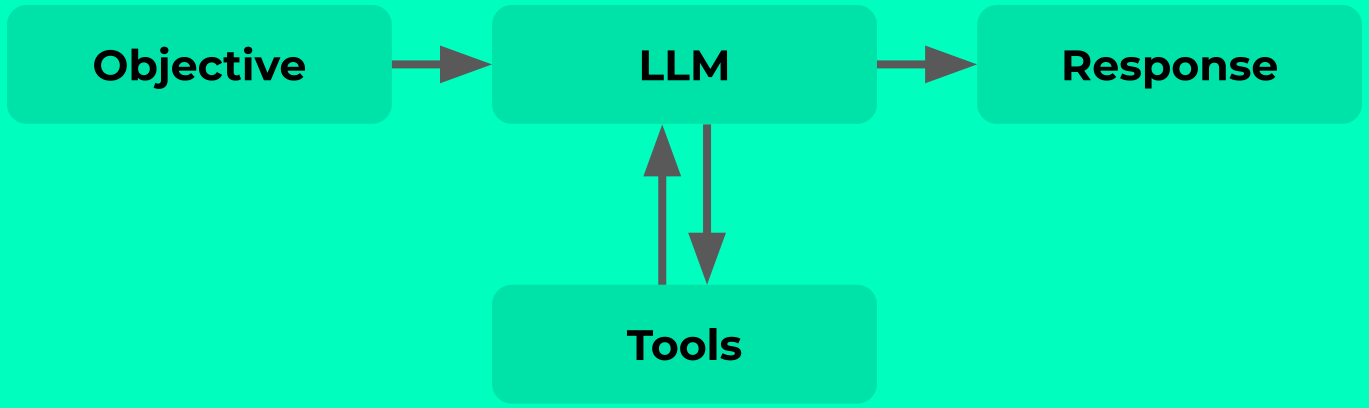
```
$ curl -s -X POST http://localhost:8000/v1/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer token-abc123" \
-d @- <<EOF
{
        "model": "google/gemma-2-2b-it",
        "prompt": "${PROMPT}",
      }
EOF
```

Creating your own private LLM API with vLLM and Kubernetes

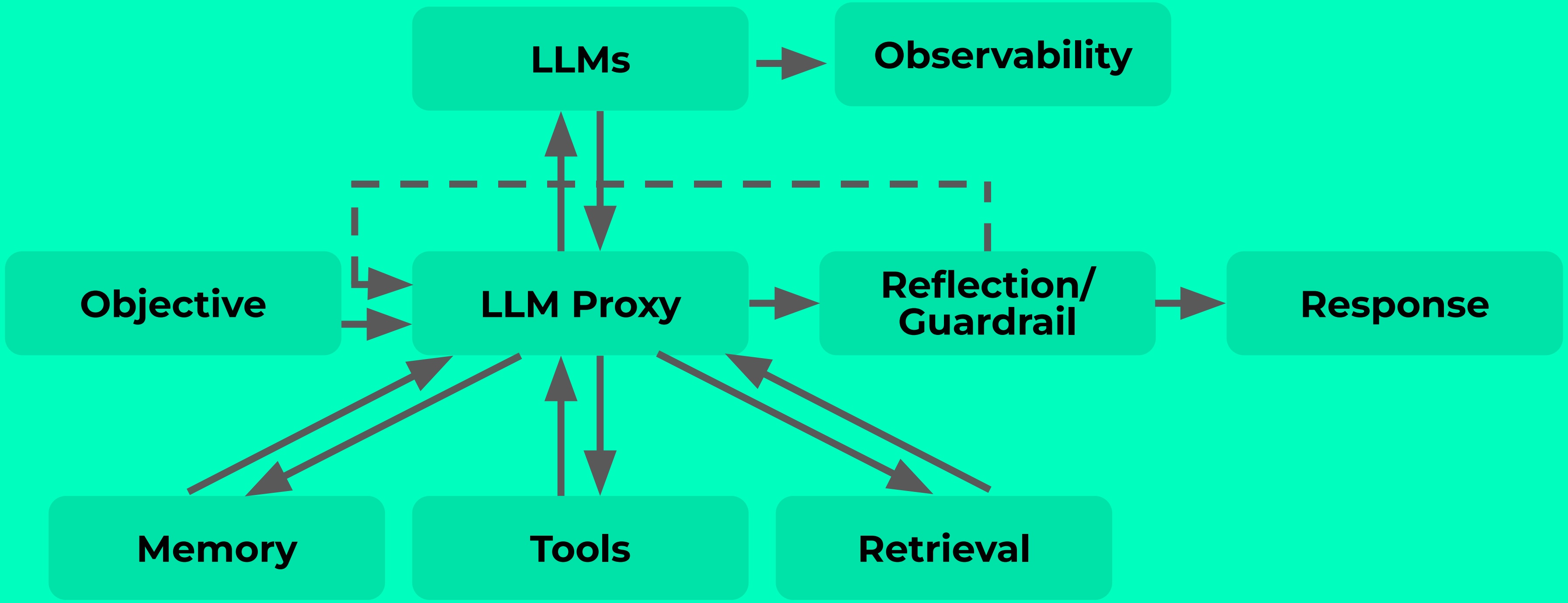
Alternatives to vLLM?

- Plain FastAPI + HuggingFace
- HuggingFace Text Generation Inference (TGI)
- SGL Project (github.com/sgl-project/sglang)
- TitanML Takeoff Server (titanml.co)

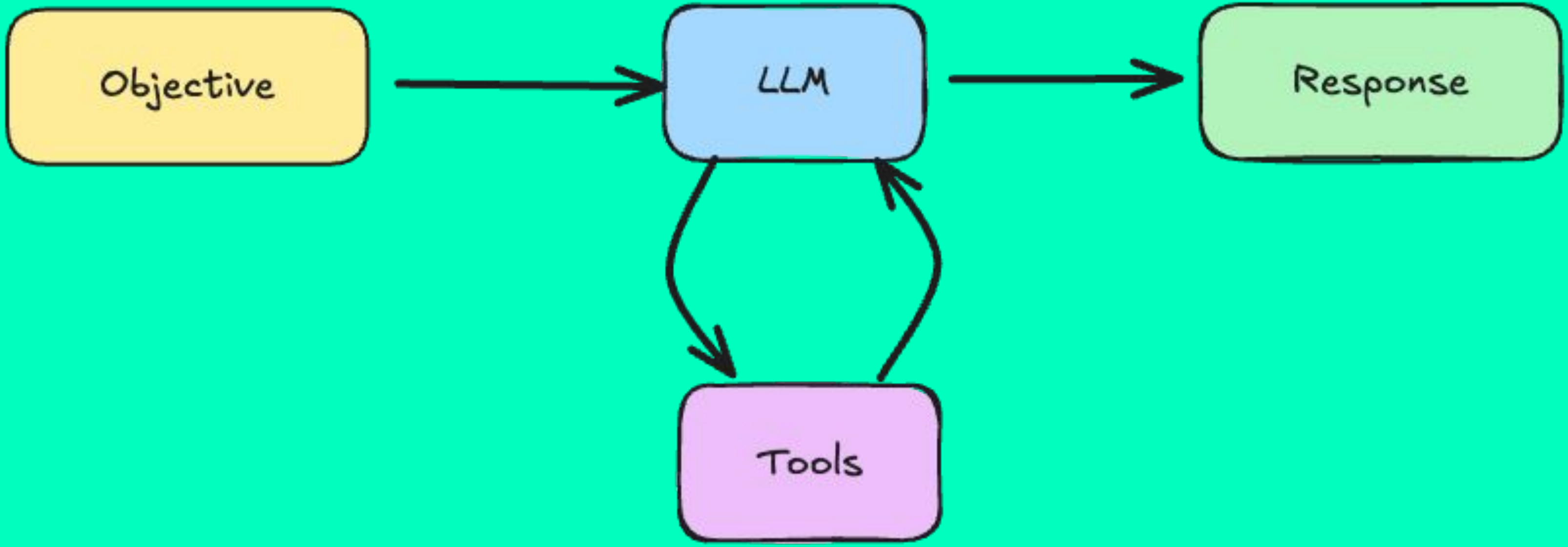
Where were we?



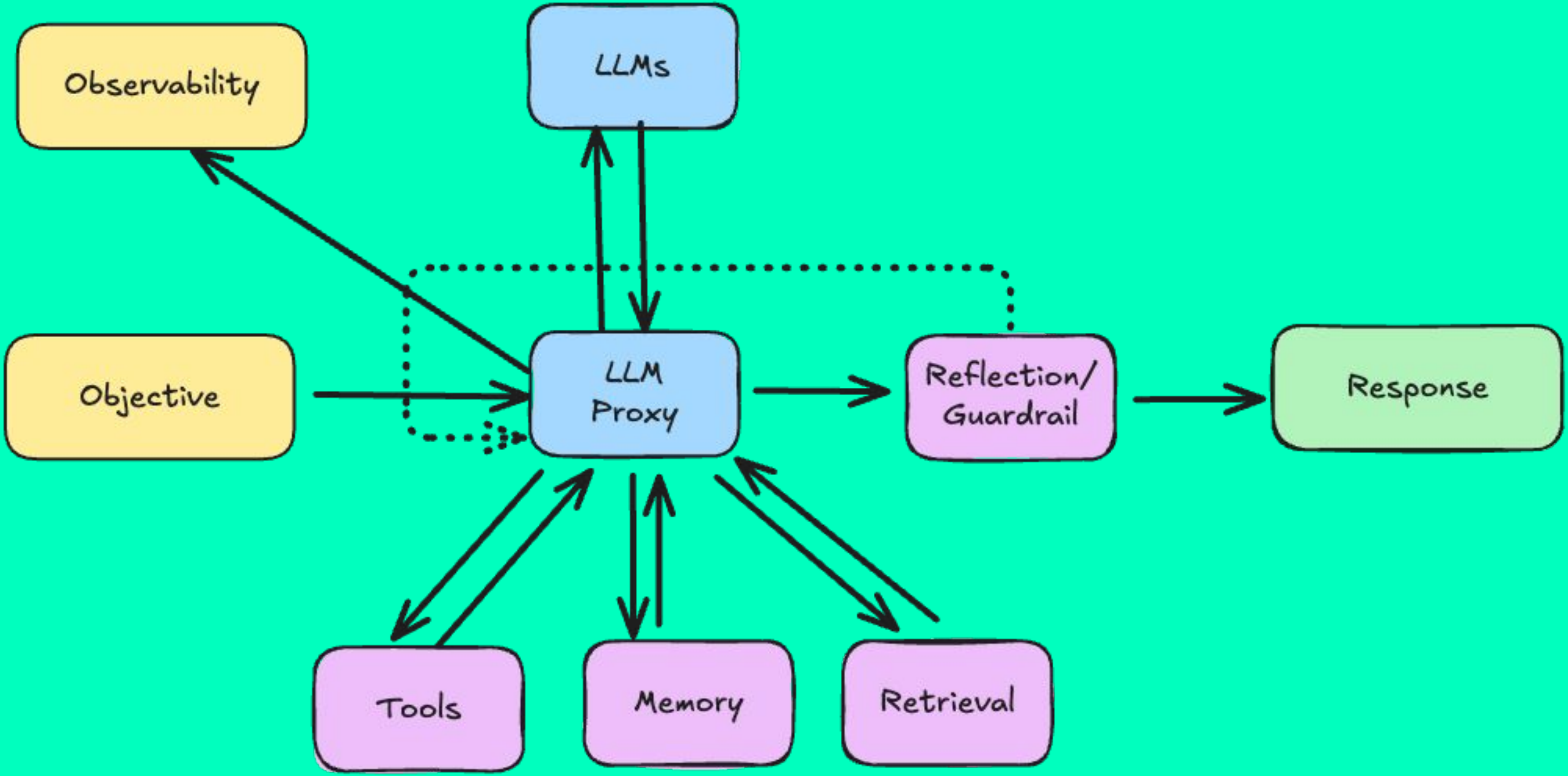
Where were we?



Where were we?



Where were we?



Tool Definition

```
{
  "properties": {
    ...
    "category_id": {
      "type": "string"
    },
    "category_type": {
      "type": "integer"
      // TODO enum here
    },
    "date": {
```