Agentic Systems beyond localhost

Behind the surface

Artur Skowroński

VIRTUSLAB

Artur Skowronski
Head of Java / Kotlin Development

VIRTUSLAB

SCALA 3.7.0 RELEASED!

Wednesday 7 May 2025

Wojciech Mazur, VirtusLab

RELEASE

JVM Weekly

Home | Notes | Foojay.io Community Newsletter | Archive | About

"Optimizing the GC when Migrating Cloud Workloads to Arm" with Kieran Hejma...

Today, we have a guest post from Kieran Hejmadi, an Software and Academic Ecosystem Manager in Arm.

MAY 1 · ARTUR SKOWRONSKI AND KIERAN HEJMADI

Latest | Top | Discussions

April - The Rest of the Story - JVM Weekly vol. 127

I know there is a lot of April left, not my fault, calendar is working against my publishing schedule this month...

APR 24 · ARTUR SKOWRONSKI

Java in AWS with Wojtek Gawroński - JVM Weekly vol. 126

I'm experimental mood recently, so today, we have the interview!

APR 17 · ARTUR SKOWRONSKI AND WOJTEK GAWROŃSKI

Make Final Mean Final... and the other JEPs - JVM Weekly vol. 125

It's been a while since we've seen JEPs! But you know, not the fully baked ones, not the ones that have advanced far i...

JVM Weekly

From the latest updates in JVM languages like Java, Kotlin, and Scala to emerging technologies like GraalVM and Quarkus, this newsletter covers a wide range of topics that are of interest to developers and tech enthusiasts.

✓ Subscribed

Recommendations MANAGE

- **Tales from the jar side** — Ken Kousen
- **Architecture Weekly** — Oskar Dudycz
- **Engineering Primer** — Bartek Antoniak
- **Tech Talks Weekly**

When someone tells me consulting sounds like googling with extra steps

I’m mostly involved in Software Consulting

TECH WEEK

Oct 06 - Oct 12 San Francisco

Oct 13 - Oct 19 Los Angeles

Find Tech Week events

Submit Event

Explore curated calendars for...

17 events
Hardtech

20 events
Future of Work

29 events
Founders

14 events
Hackathons

Explore all events

All Mo Tue Wed Thu Fri Sa Su

Event filters

Search /

Topics Clear

Applications Defense Fintech Climate
AR / VR Engineering Cybersecurity Hardware
GTM Women-focused B2B Gaming AI
Deep Tech Creators Crypto / Web3
International / Expansion Fundraising / Investing
B2C / Consumer Media / Entertainment SaaS
Healthcare / Healthtech HR / Hiring Infrastructure
- See less

Neighborhood

Invite Only · Serverless AI, Inc.
Agent runtime and Investor happy hours - Infrastructure for the future of AI
Mon Oct 6 · 6:00 am
Hayes Valley

Virio
Virio Run Club
Mon Oct 6 · 6:30 am
Embarcadero

Delphi, D.E. Shaw
Delphi x D.E. Shaw Ventures Biking & Coffee
Mon Oct 6 · 7:00 am
Marina

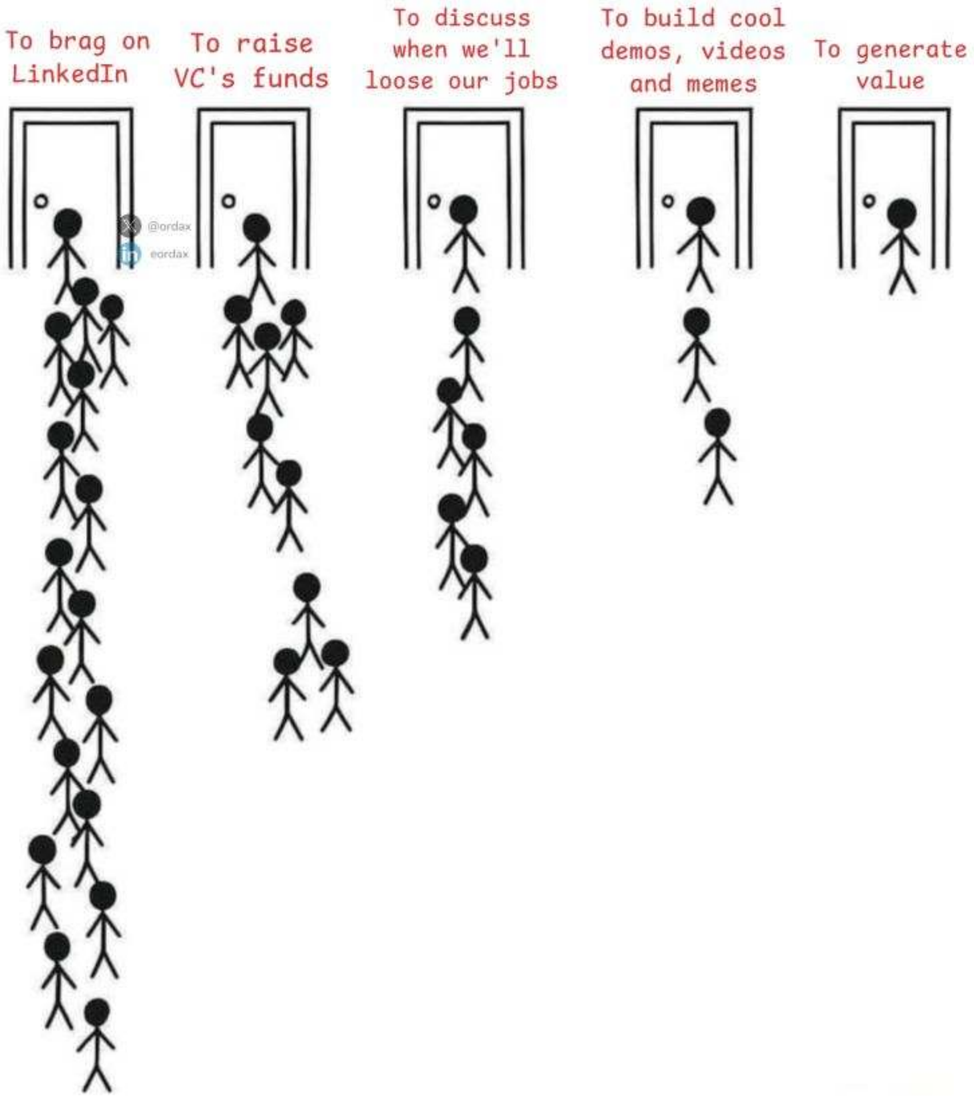
TrueYou
Founders Cold Plunge
Mon Oct 6 · 7:00 am
SOMA

Sponsored · AWS, Seed AI
Founders Shaping the Future: California M&A Policy Path Forward
Mon Oct 6 · 10:30 am
Chinatown

Walnut AI
Agents Meet SF Cohort: Meet Your Next Partner
Mon Oct 6 · 8:00 am
SOMA

Assenteo
Trust on Toast: AI Breakfast
Mon Oct 6 · 8:00 am
Downtown (SF)

How people is using Agentic AI



Cole Stryker
Staff Editor, AI Models
IBM Think

What is agentic AI?

Agentic AI is an artificial intelligence system that can accomplish a specific goal with limited supervision. It consists of AI agents—machine learning models that mimic human decision-making to solve probl
system, each agent performs a specific subta
their efforts are coordinated through AI orche

Unlike traditional AI models, which operate within prede
intervention, agentic AI exhibits autonomy, goal-driven b
refers to these models' agency, or, their capacity to act ir

Topic > Agentic AI

What is agentic AI?

Agentic AI is an advanced form of artificial intelligence focused on autonomous decision-making and action. Unlike traditional AI, which primarily responds to commands or analyzes data, agentic AI can set goals, plan, and execute tasks with minimal human intervention. This emerging technology has the potential to revolutionize various industries by automating complex processes and optimizing workflows.

Contact sales

What is Agentic AI?

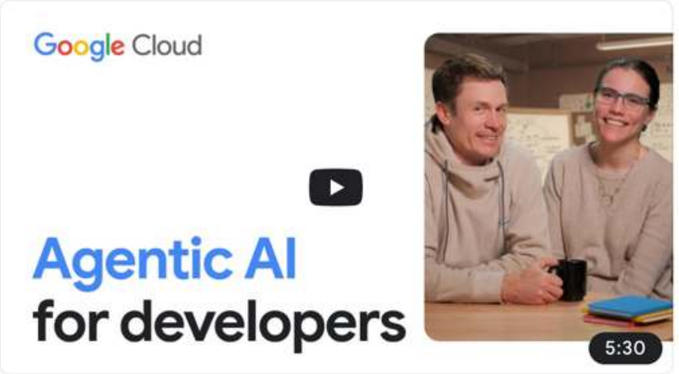
Agentic AI is an autonomous AI system that can act independently to achieve pre-determined goals. Traditional software follows pre-defined rules, and traditional artificial intelligence also requires prompting and step-by-step guidance. However, agentic AI is proactive and can perform complex tasks without constant human oversight. "Agentic" indicates agency — the ability of these systems to act independently, but in a goal-driven manner.

AI agents can communicate with each other and other software systems to automate existing business processes. But beyond static automation, they make independent contextual decisions. They learn from their environment and adapt to changing conditions, enabling them to perform sophisticated workflows with accuracy.

What Is Agentic AI?

Agentic AI uses sophisticated reasoning and iterative planning to autonomously solve complex, multi-step problems.

October 22, 2024 by Erik Pounds



Agentic AI: Workflows versus agents

t Blueprints was changed to NVIDIA Blueprints in
name have been updated in this blog.

OpenAI cofounder Andrej Karpathy says it will take a decade before AI agents actually work

By Lakshmi Varanasi + Follow

OpenAI cofounder Andrej Karpathy is unimpressed with the state of AI agents. San Francisco Chronicle/Hearst Newspapers via Getty Images/Contributor/Getty Images

Alex Barády · 2nd
ENDGAME Founder | AI Entrepreneur, Executive Advisor, & I...
3d · + Follow ···

OpenAI Co-founder just dropped a bombshell:

"AI agents just don't work. It's a slop" ...more

354

Sarang Fegde · 3rd+
Founder of Hypelinks | Ex-Amazon | Post Exit Founder | Inve...
Visit my website
2d · + Follow ···

In the recent interview, Andrej Karpathy (ex-Tesla, ex-OpenAI) challenges the hype around "AI agents" and backs a slower, more structural build-out into the next 5-10 years. Here are my take aways.

Linas Beliūnas · 2nd
Building a Safer Internet with AI 🍵 | Scouting for top startu...
View my services
5d · Edited · + Follow ···

OpenAI co-founder Andrej Karpathy calls AI Agents slop:

"Overall, the models are not there. And I feel like the industry [...] it's ...more

1,871 246 comments · 172 reposts

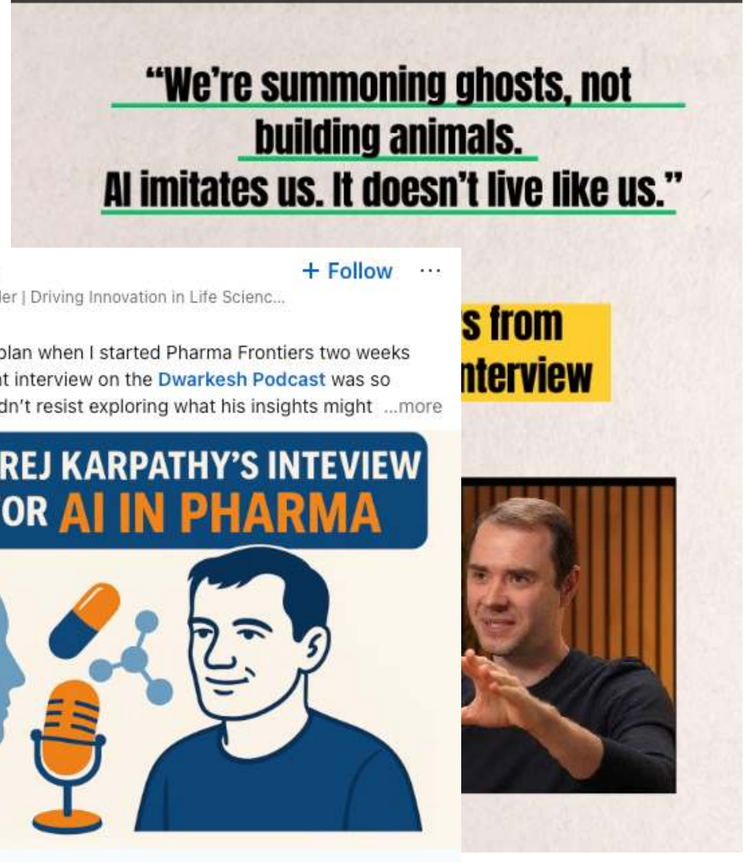
Like Comment Repost Send

Tijana Zivkovic · 2nd
AI for Leaders | Certified AI Educator | AI edtech startup for...
Visit my website
4h · + Follow ···

Karpathy just gave a masterclass in AI realism.

7 lessons from Andrej Karpathy's most grounded interview yet:

Andrej Karpathy's interview · 9 pages



Kevin Ray · 2nd
AI & Data Science Leader | Driving Innovation in Life Scienc...
8h · + Follow ···

Whilst maybe not part of the plan when I started Pharma Frontiers two weeks ago, Andrej Karpathy's recent interview on the Dwarkesh Podcast was so thought-provoking that I couldn't resist exploring what his insights might ...more

Pharma Frontiers + Subscribe

What Andrej Karpathy's Interview Means for AI in Pharma

Kevin Ray

Zgodnie z prawami fizyki
trzmiel nie może latać

Na całe szczęście
trzmiel nie zna praw fizyki

Zgodnie z prawami fizyki
trzmiel nie może latać

Na całe szczęście
trzmiel nie zna praw fizyki

Więc ma wy*ebane

Two generals problem

SOME CONSTRAINTS AND TRADEOFFS
IN THE DESIGN OF
NETWORK COMMUNICATIONS*

E. A. Akkoyunlu
K. Ekanadham
R. V. Huber†
Department of Computer Science
State University of New York at Stony Brook

A number of properties and features of interprocess communication systems are presented, with emphasis on those necessary or desirable in a network environment. The interactions between these features are examined, and the consequences of their inclusion in a system are explored. Of special interest are the time-out feature which forces all system table entries to "die of old age" after they have remained unused for some period of time, and the insertion property which states that it is always possible to design a process which may be invisibly inserted into the communication path between any two processes. Though not tied to any particular system, the discussion concentrates on distributed systems of sequential processes (no interrupts) with no system buffering.

Key Words and Phrases: interprocess communication, computer networks, ports.

CR Categories: 3.81, 4.32, 4.39

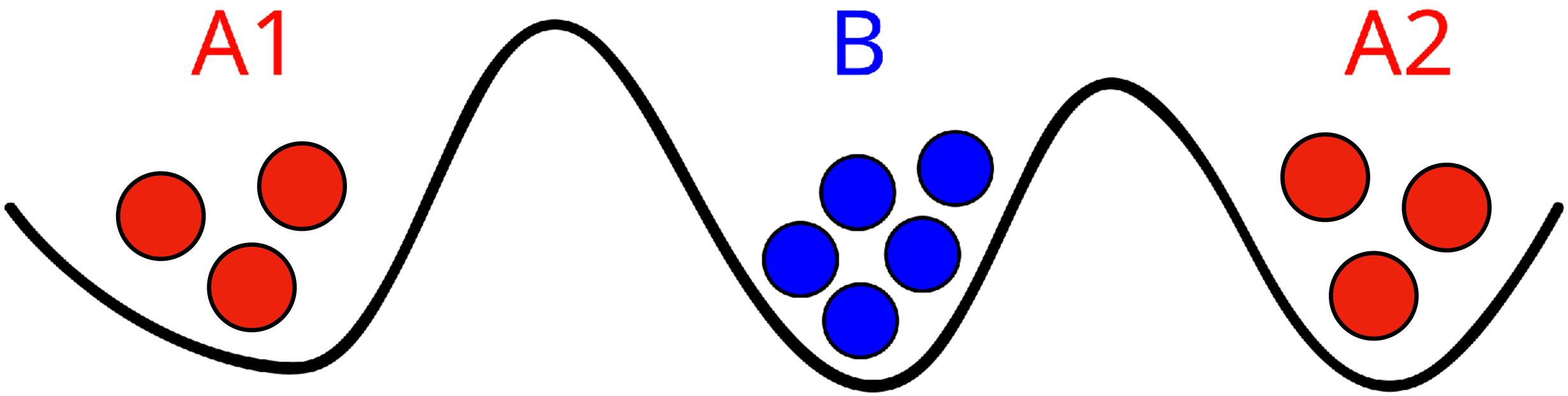
1. Introduction

The design of an interprocess communication mechanism (IPCM) usually starts with a description of the desired behavior of the system and the services to be provided. In selecting the features to be incorporated into the IPCM, the greatest amount of care is required, for these features are interdependent to a great degree, and it is crucial that the design process start with a complete, detailed specification of the system to be designed, with the consequences of each decision fully explored and understood. The temptation of piecemeal design is to be avoided at all costs.

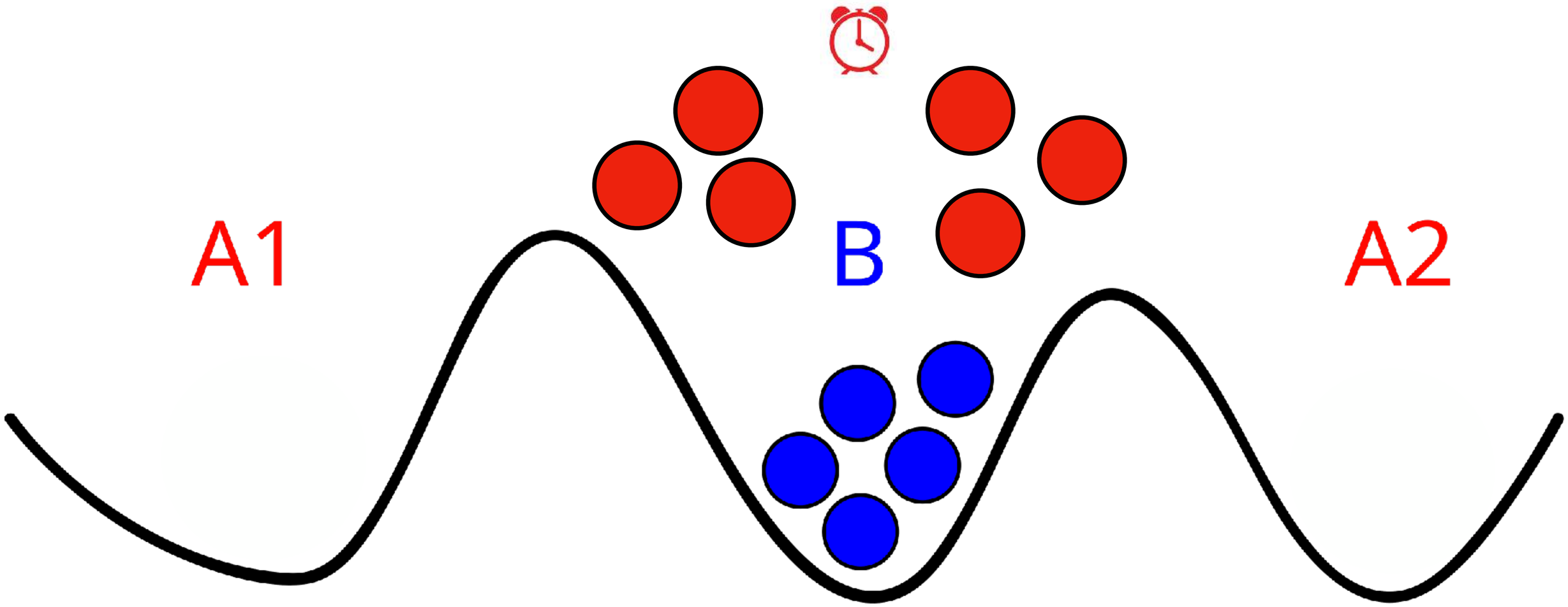
results in some horrendous code 'patched' into the system, and much elegance is lost. The resulting system is harder to implement, verify, understand, debug, and maintain. These are the questions which extract a "well, we didn't actually implement it that way," response from system designers.

Unfortunately, with few exceptions [6, 10, 15], there is little guidance to be found in the published literature on this important point - how to arrive at a consistent and elegant design. This paper is a modest attempt to help fill this gap. The paper will address itself to general concepts rather than to the specifics of a particular de-

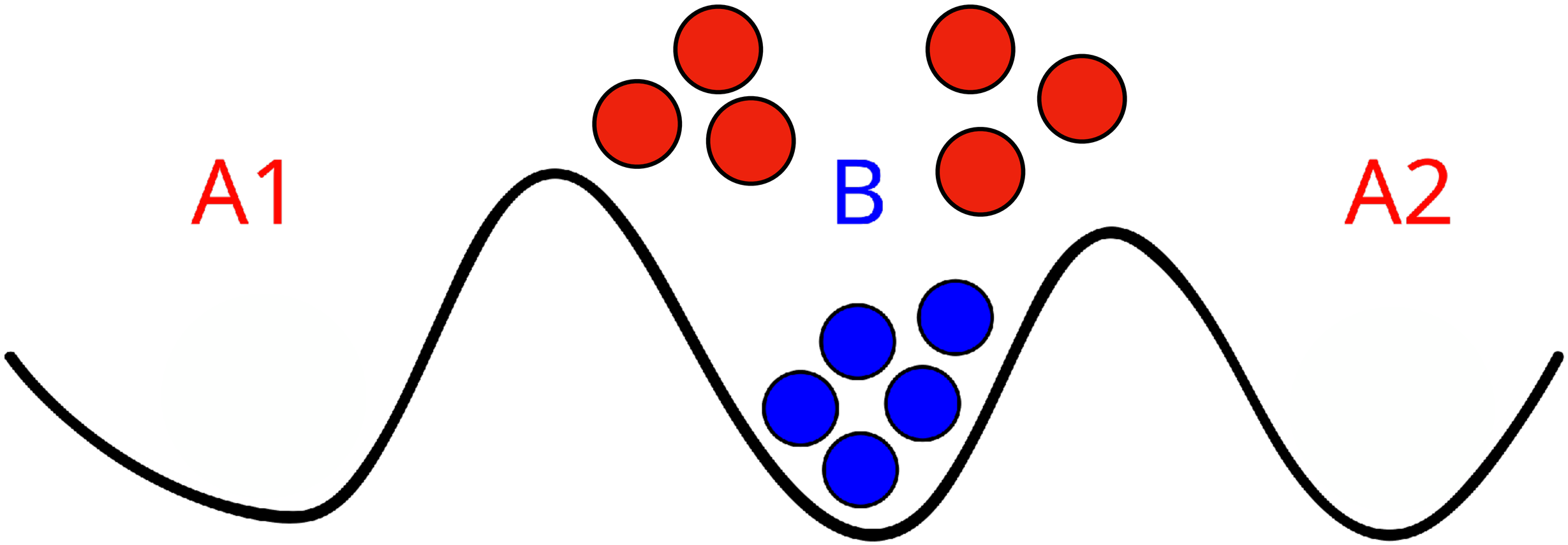
Two generals problem



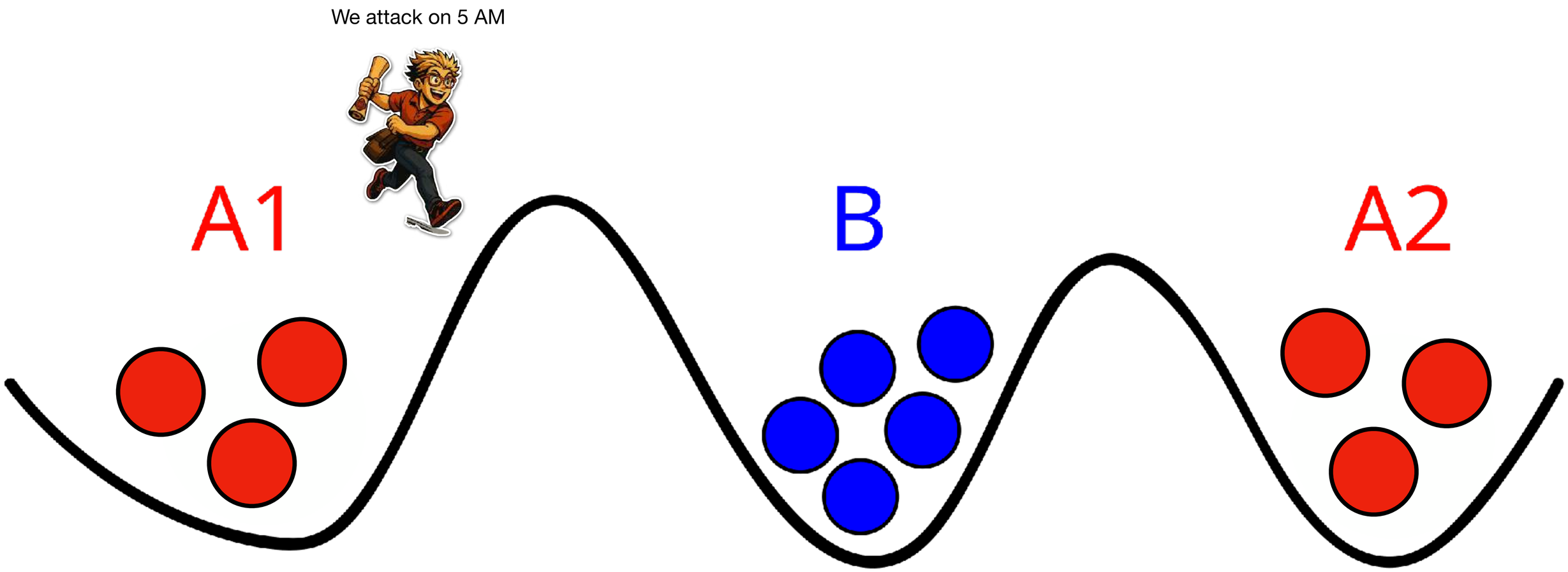
Two generals problem



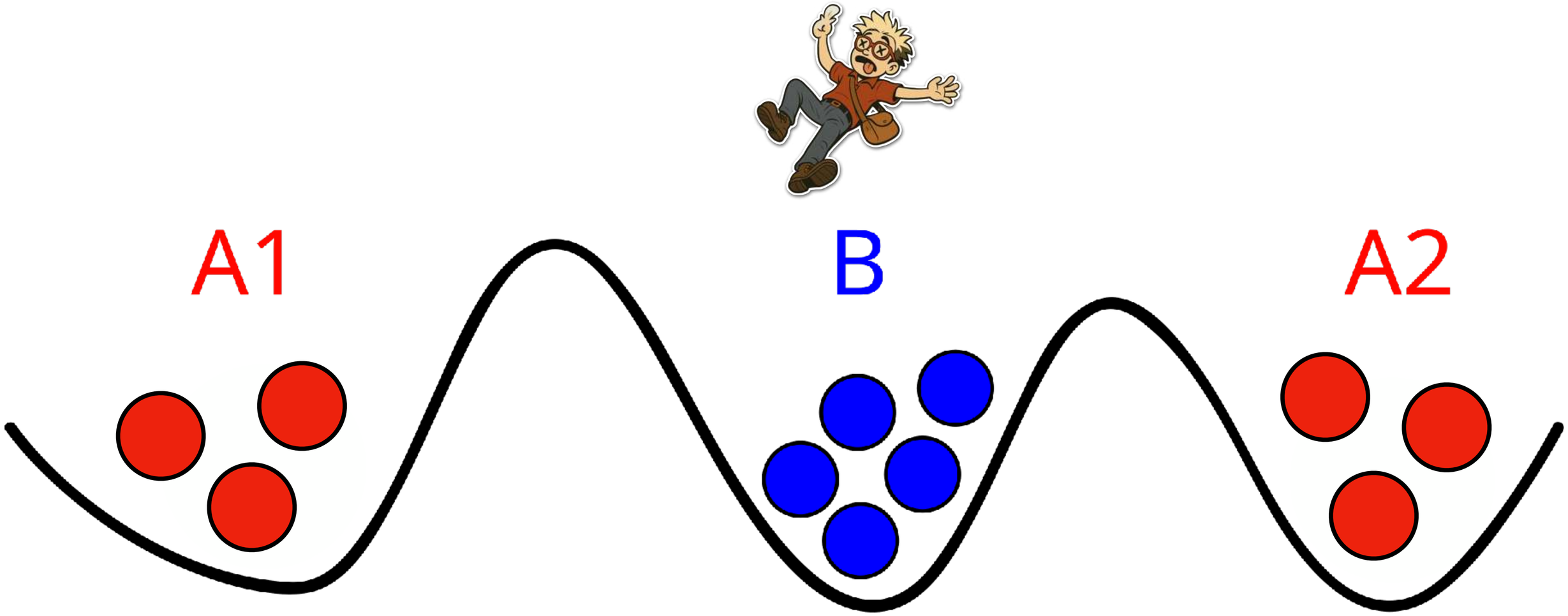
Two generals problem



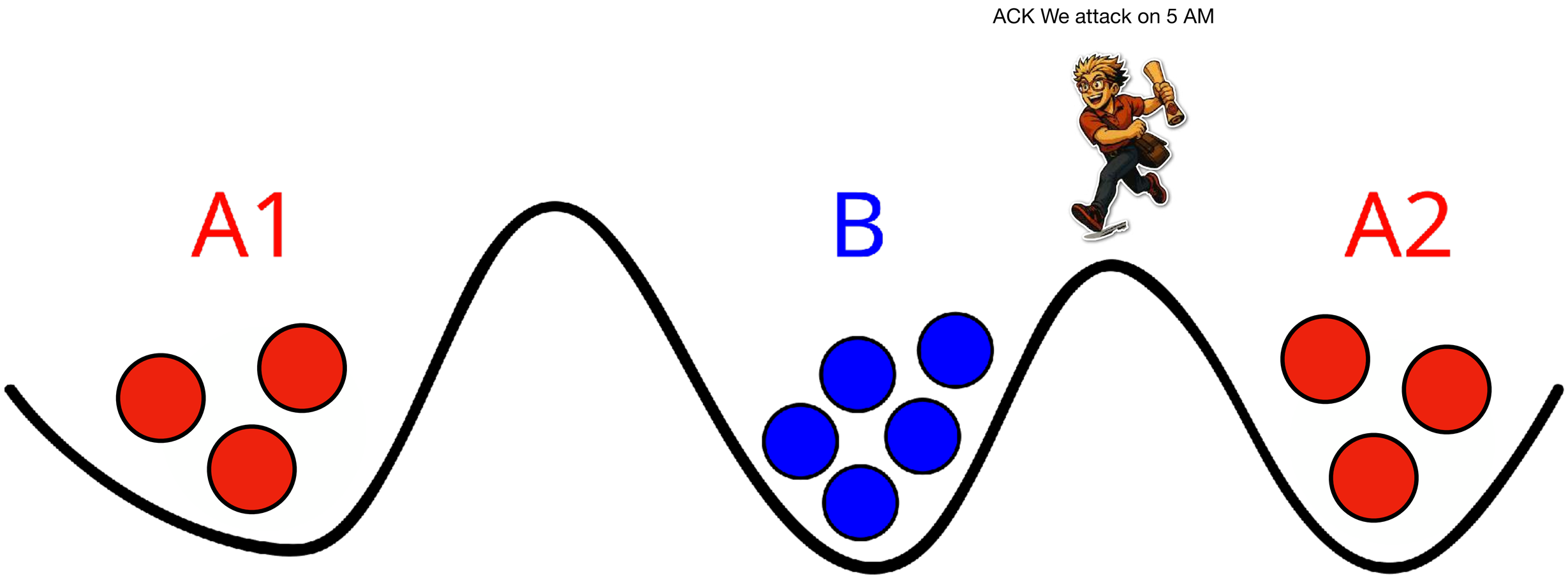
Two generals problem



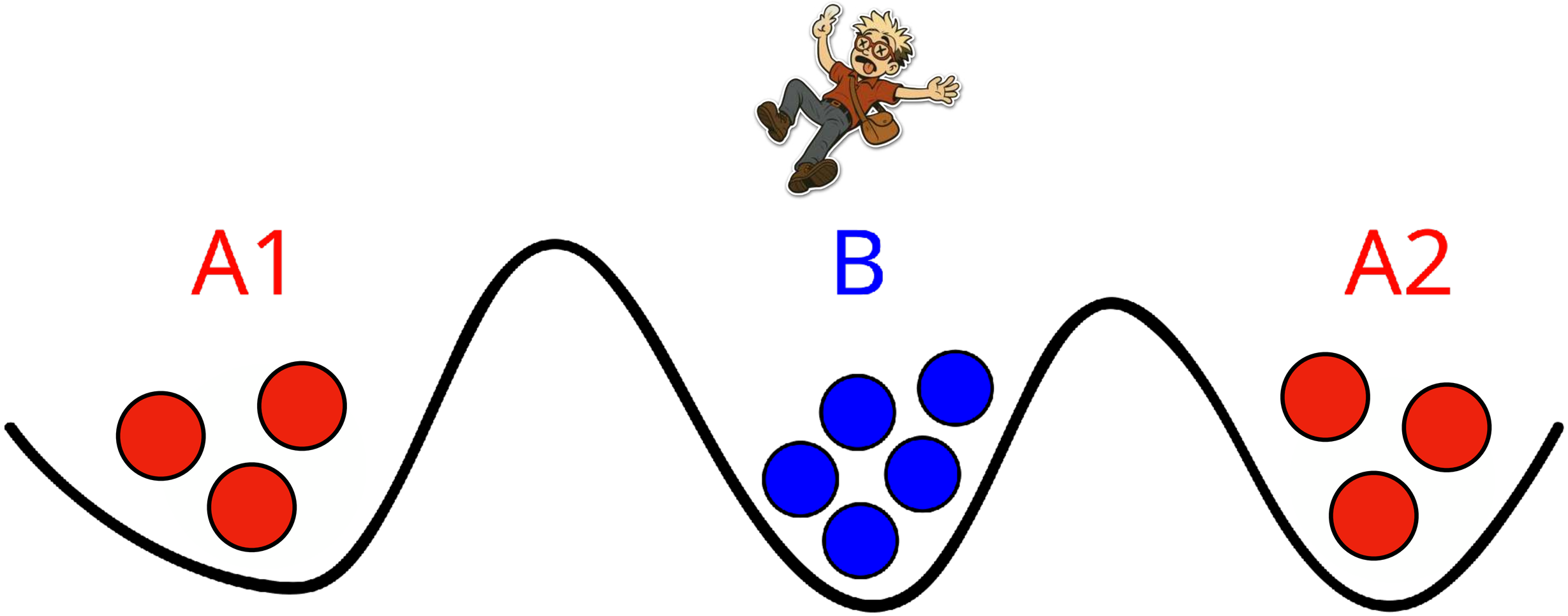
Two generals problem



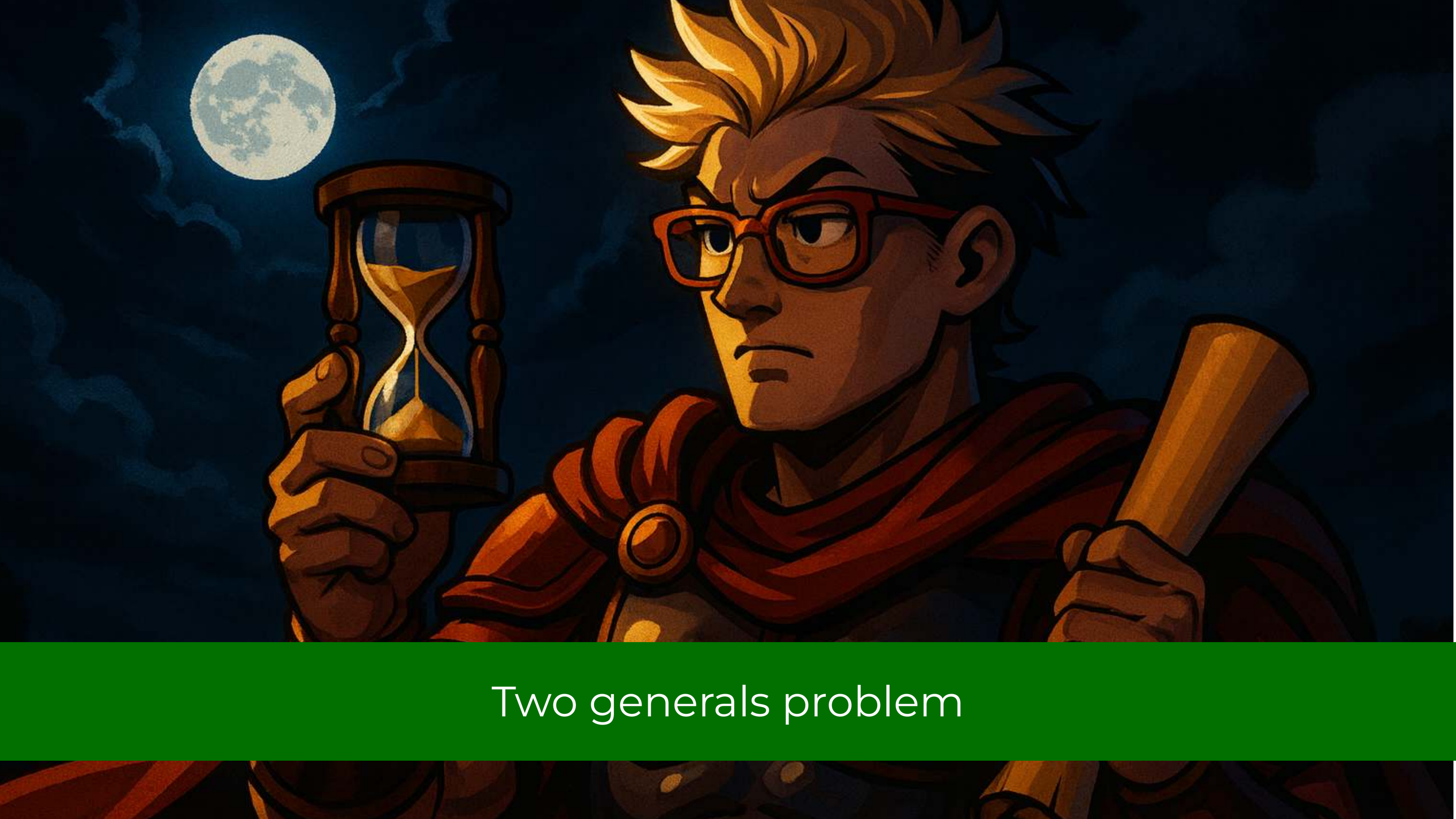
Two generals problem



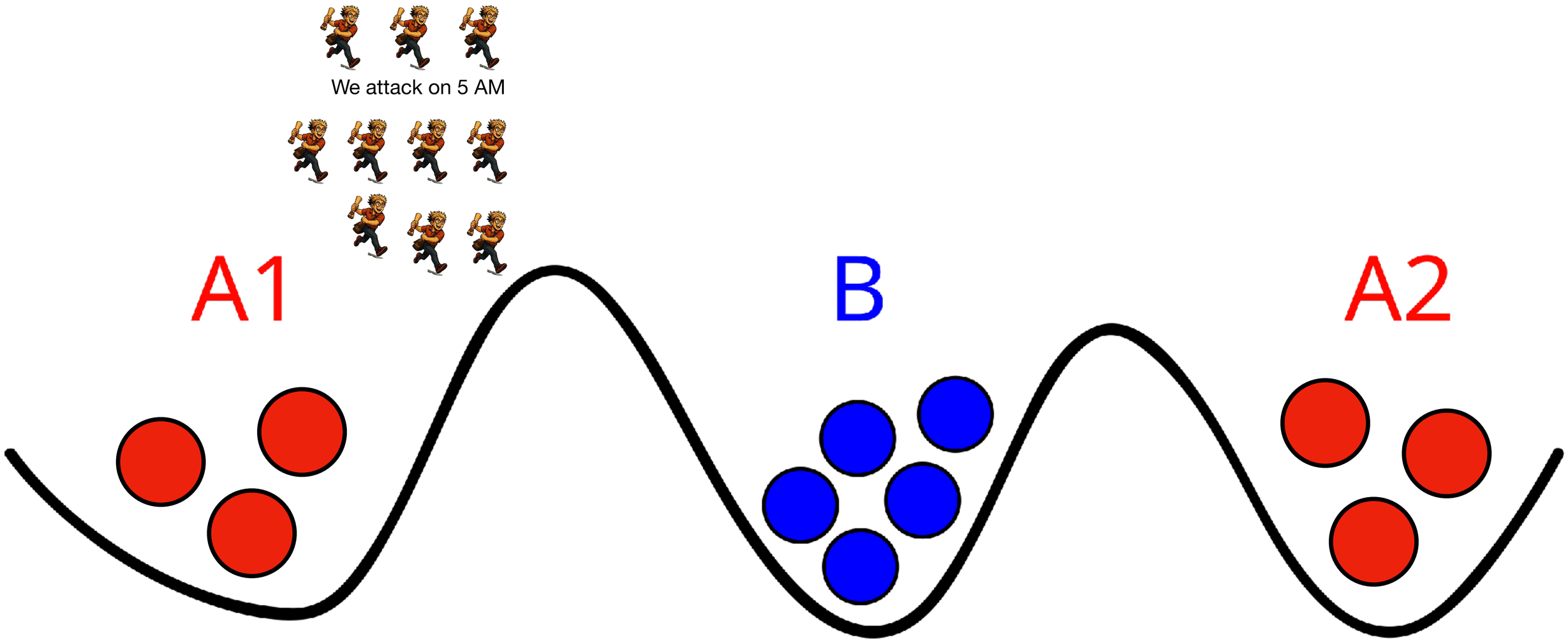
Two generals problem



Two generals problem



Two generals problem



Two generals problem

The point I'm trying to make

The way I explain it

OTHER PEOPLE WHEN THERE IS HYPE ON THE MARKET:

ME:

It always leads to a deeper understanding. It's the only way to build. If I can't build it, I don't understand it. That's a Feynman quote, I believe. I 100% have always believed this very strongly, because there are all these micro things that are just not properly arranged and you don't really have the knowledge. You just think you have the knowledge. So don't write blog posts, don't do slides, don't do any of that. Build the code, arrange it, get it to work. It's the only way to go. Otherwise, you're missing knowledge.

160 Slides...
45 minutes

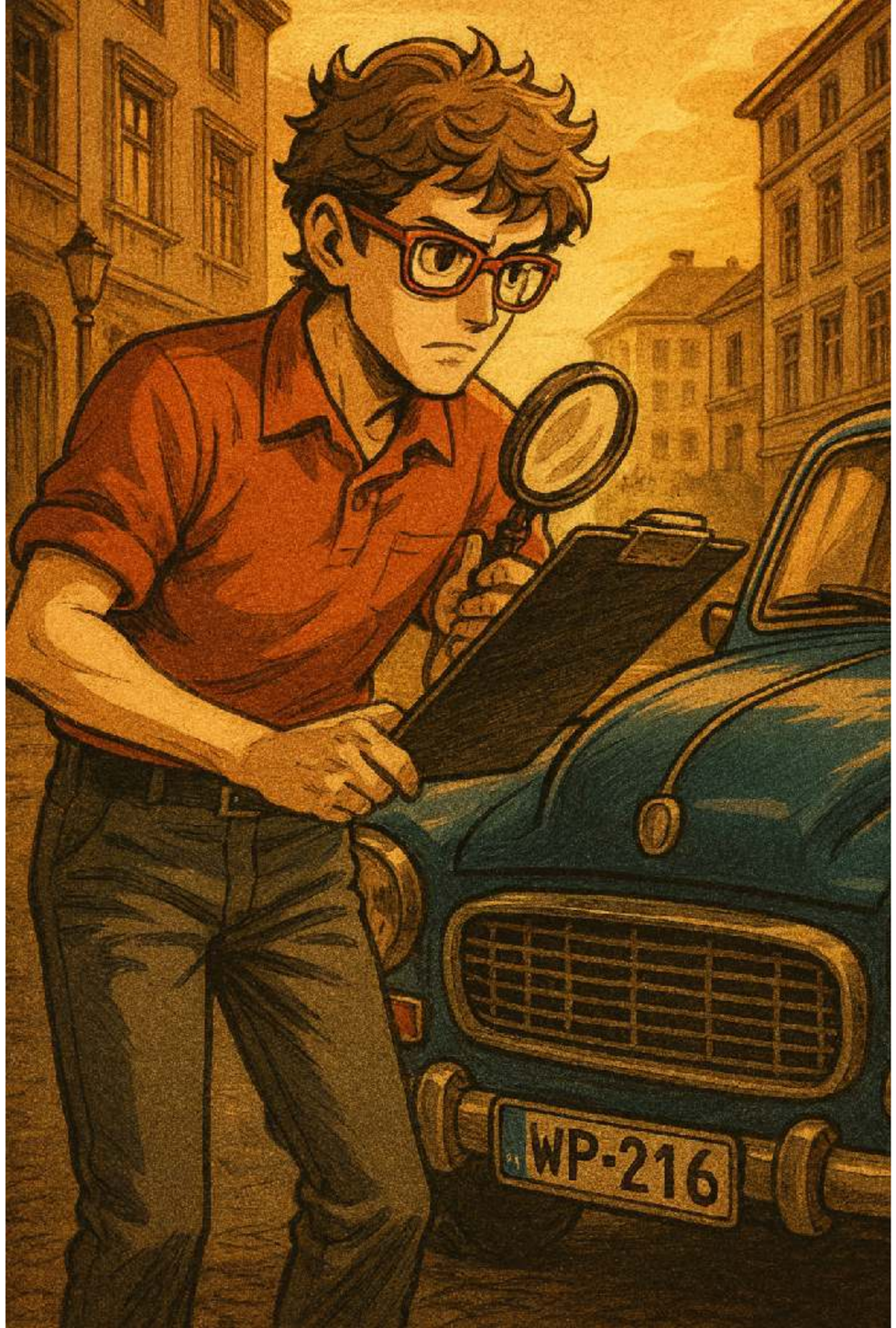
Still the helicopter view

Let’s start with InsureTech itself

Underwriting is simply the assessment of risk and a decision about “on what terms you get to take part” - done by a specialist or a company.

The outcome of underwriting is usually one of the following decisions:

- We accept and specify the terms (price/premium, limit, exclusions),
- We accept conditionally (e.g., a higher price or additional collateral),
- We decline (risk is too high)

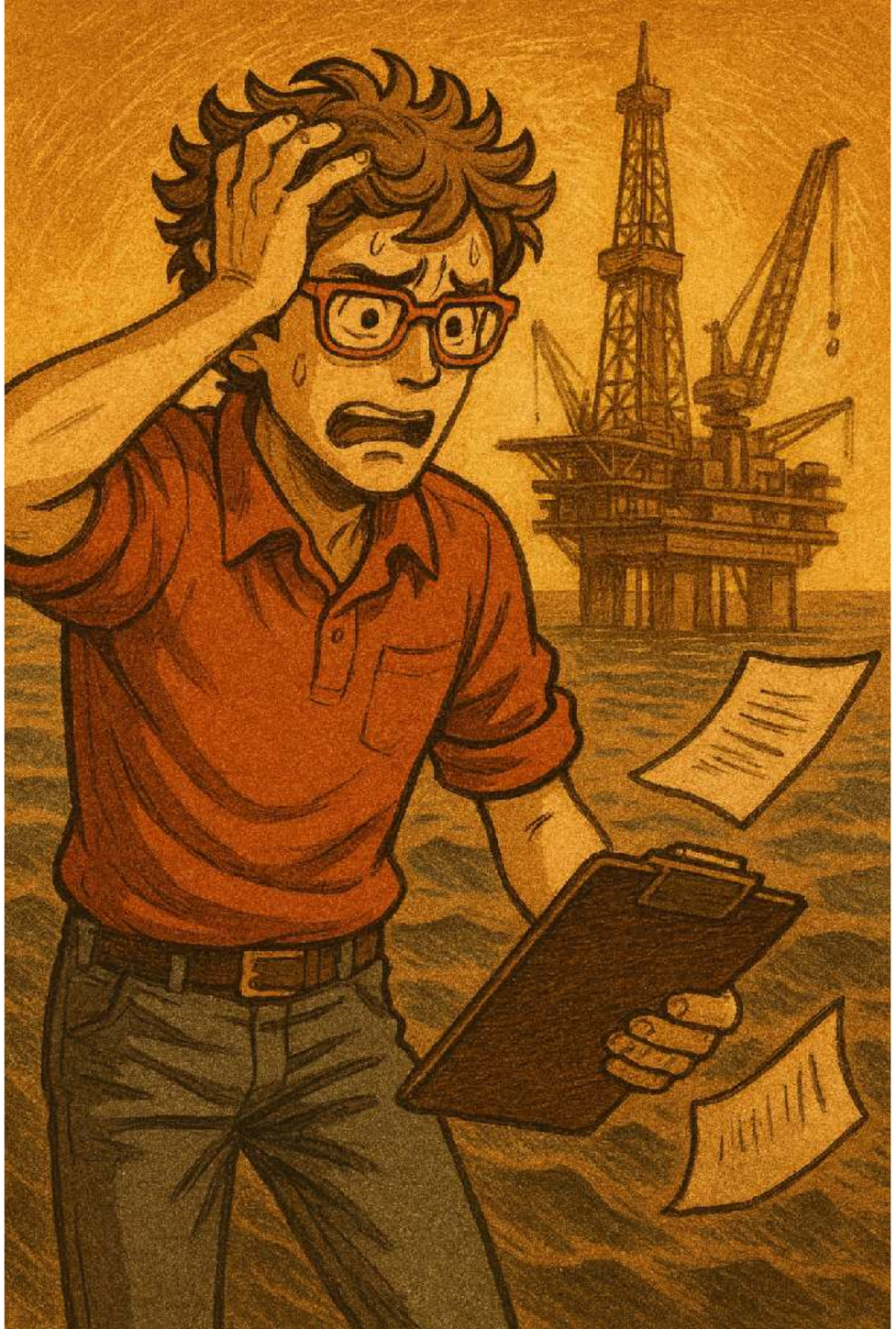


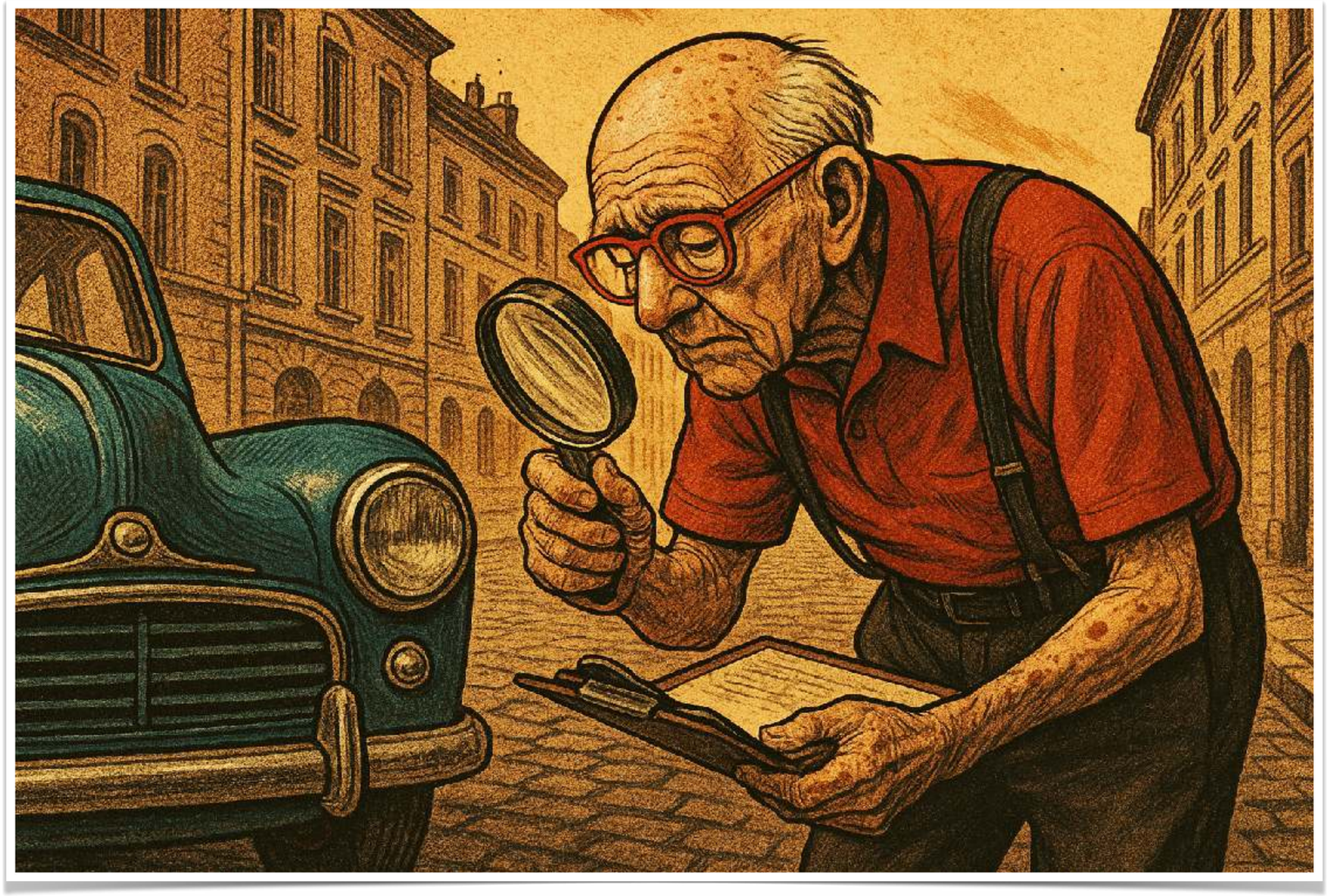
Standard (home / car)

- High-volume, standardized risk
- Simple pricing.
- Broad off-the-shelf packages
- Little customization.
- Lots of historical data → stable pricing.
- Simple underwriting (questionnaire, scoring)
- Standard claims handling, typical repair/service networks

Specialized insurance (e.g., cyber, D&O, marine, events, fine art, professional liability)

- Unique/niche risks, often with complex causality chains.
- Tailor-made policies with hand-negotiated clauses.
- Less data → higher uncertainty → higher or variable rates.
- Deep underwriting (audits, due diligence, process/IT assessment).
- Specialist claims handling (industry experts, forensics).
- Distributed via specialized brokers; limited market capacity.





Deemed as "Grey Hair" business

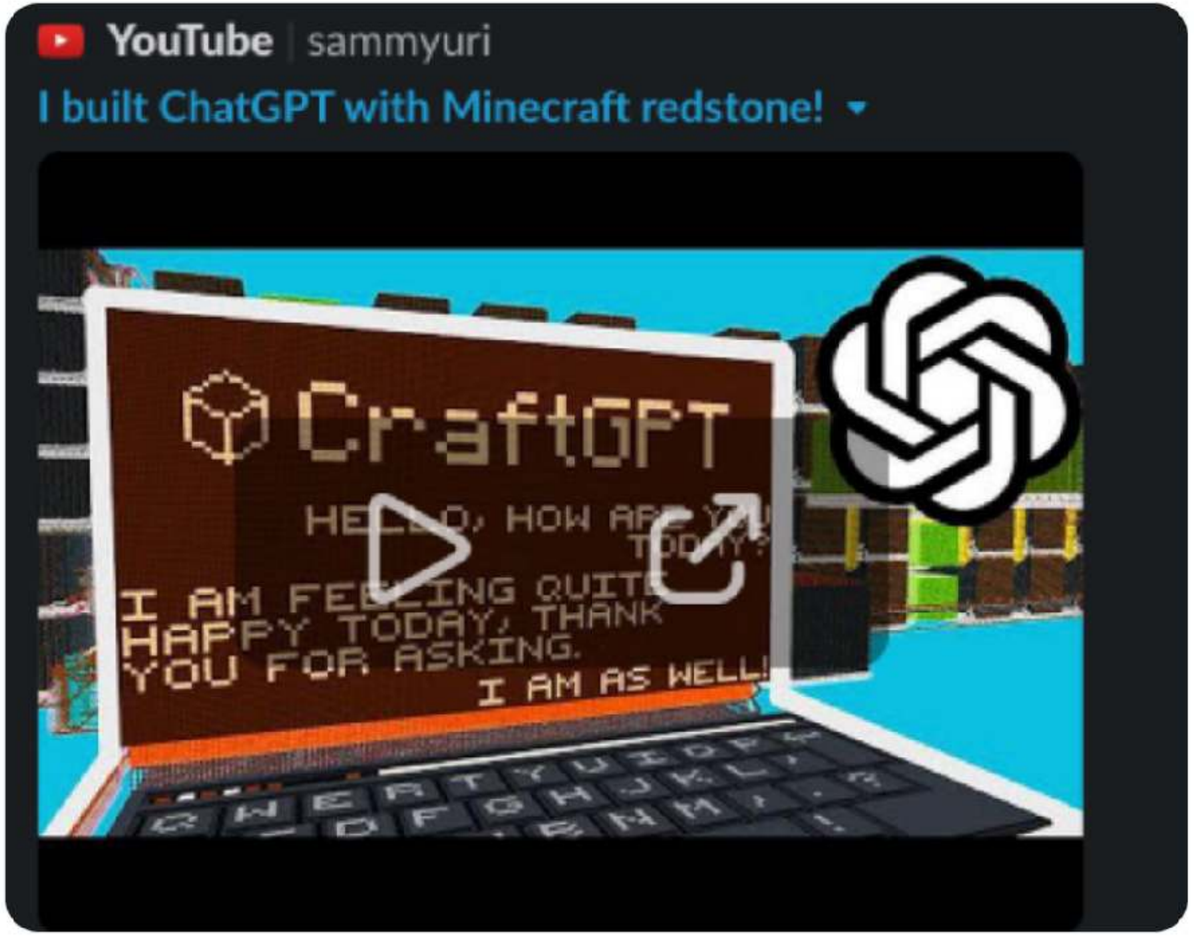
half the workforce retires by 2028

Artur Skowronski
@ArturSkowronski

Promuj

Me: I can’t find a job in the industry. I wonder what my competition is doing.
Competition:

Przetłumacz wpis



US Insurance Tech Market Map

Health

oscar, Clover, bright health, DevotedHealth, GUSTO, Namely, Collective Health, decisely, JUSTWORKS., LUMITY, Stride Health, Hixme, bind, GoHealth, JOANY, HealthSherpa, GRAVIE, SimplyInsured, allay, beam, emerge, GetInsured, HealthCare.com, friday

Home, condo & renters

Jetty, l·ippo, kin., SURE., Rhino, LeaseLock, TheGuarantors, Matic, gabi, swyft, Covered, Quilt

Auto

metromile, ROOT, the zebra, INSHUR, EVERQUOTE, clearcover, COVERHOUND, insurify, go, Goji

Title

Qualia, SPRUCE, STATES TITLE

Pet

FIGO

Drone

verifly

Personal item

Cover, trov

Small commercial

nextinsurance, coverwallet, EMBROKER, insureon, JONES, FOUNDERSHIELD, CORVUS, PIE INSURANCE, Huckleberry, QBIS, Slice

Cyber

PALADIN Cyber, Coalition, a b

Life & annuity

Ladder, bestow, tomorrow, Health I.Q., Fabric, policy genius, MIRA, ensurem, covr, BLUEPRINT INCOME

Crop

CROP PRO, WorldCover

B2B

P&C

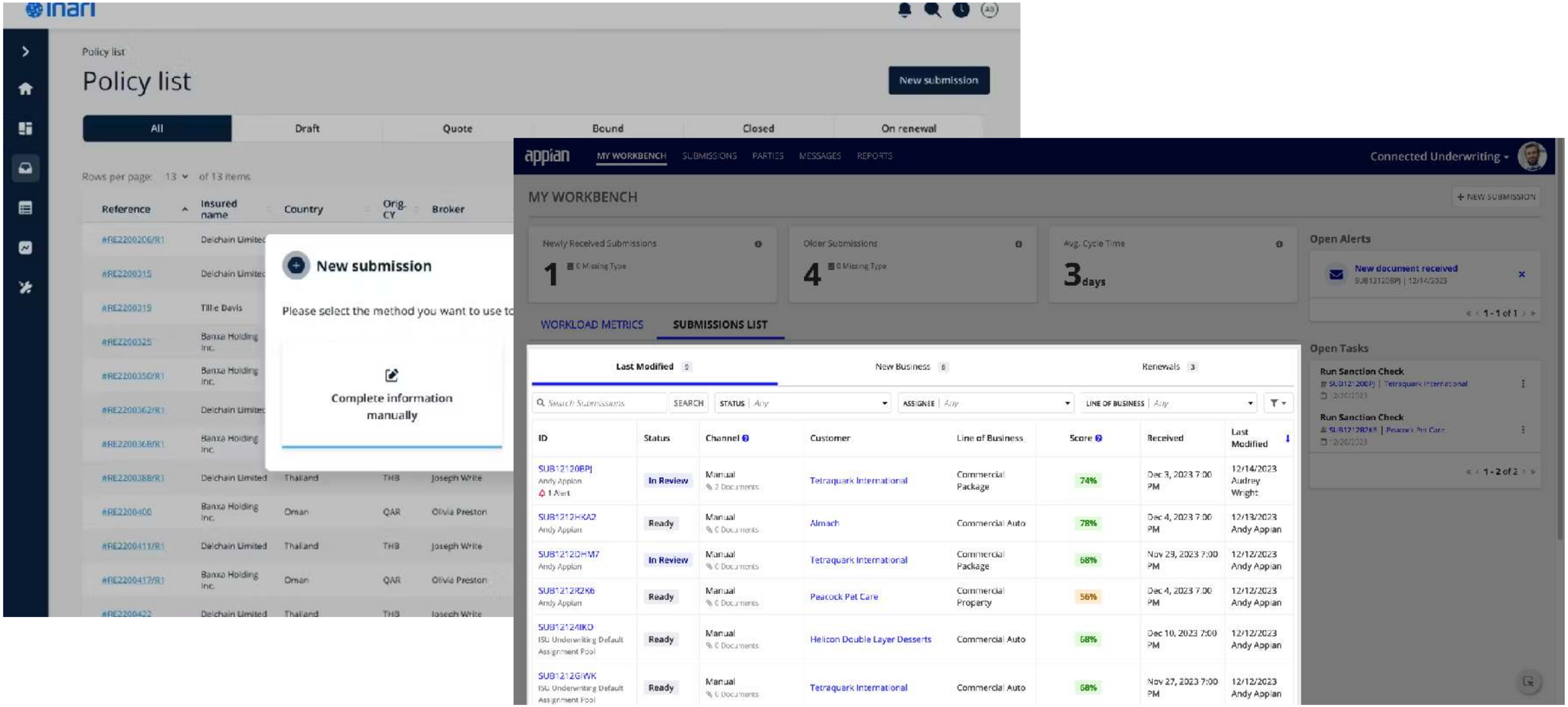
snapsheet, Groundspeed, CARPE DATA, Weather Analytics, datacubes, Zeguro, ONE INC, TrueMotion, CyberCube, Converge, CLARA, Indio, ZENDRIVE, TENSORFLIGHT, ledger Investing, riskgenius, socotra, PILLAR, Praedicat, Cyberwrite

Life

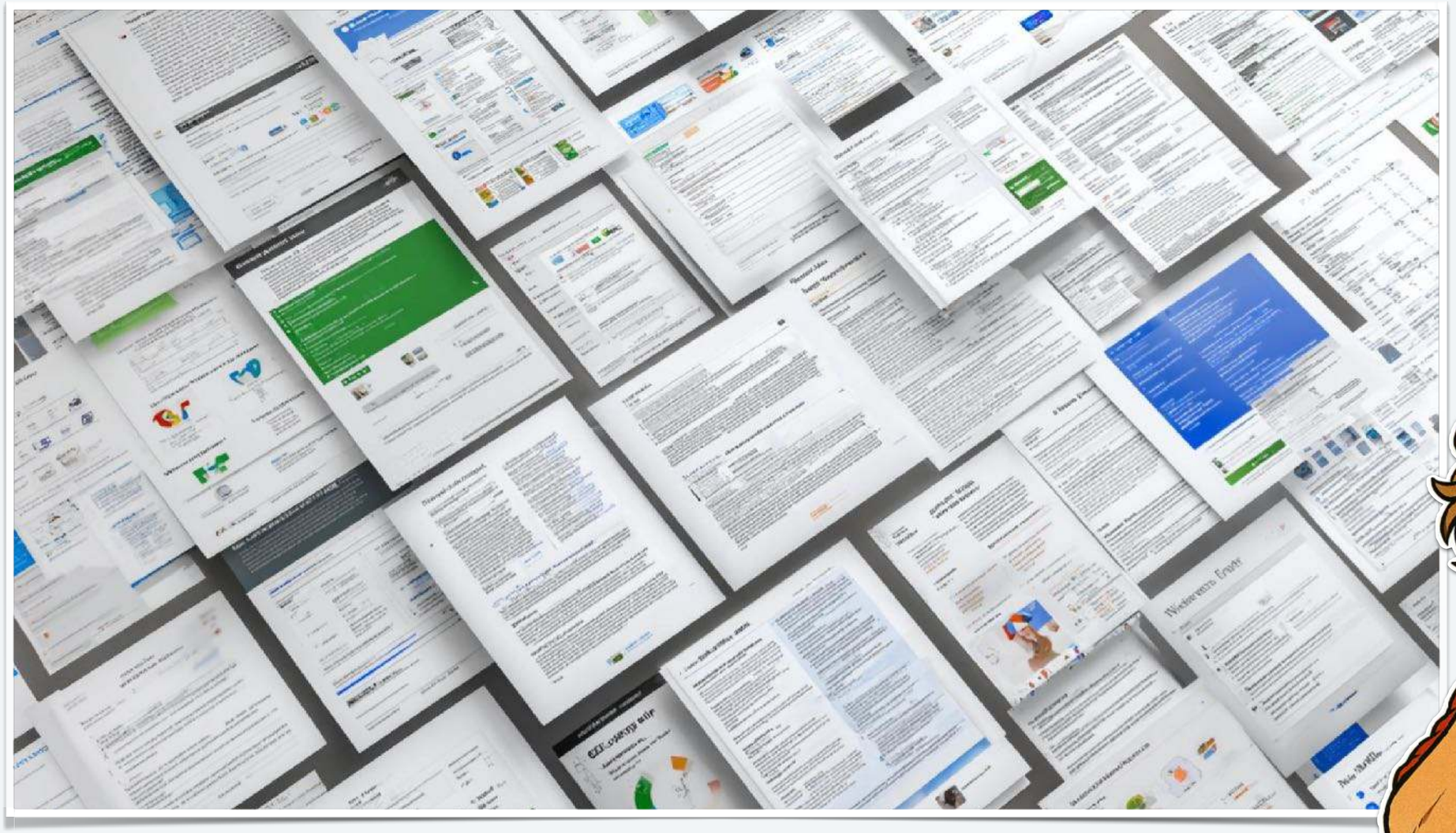
iPipeline, Atidot, vivametrica, sureify, ClaimVantage, Life.io, UNQORK, LAPETUS

Health

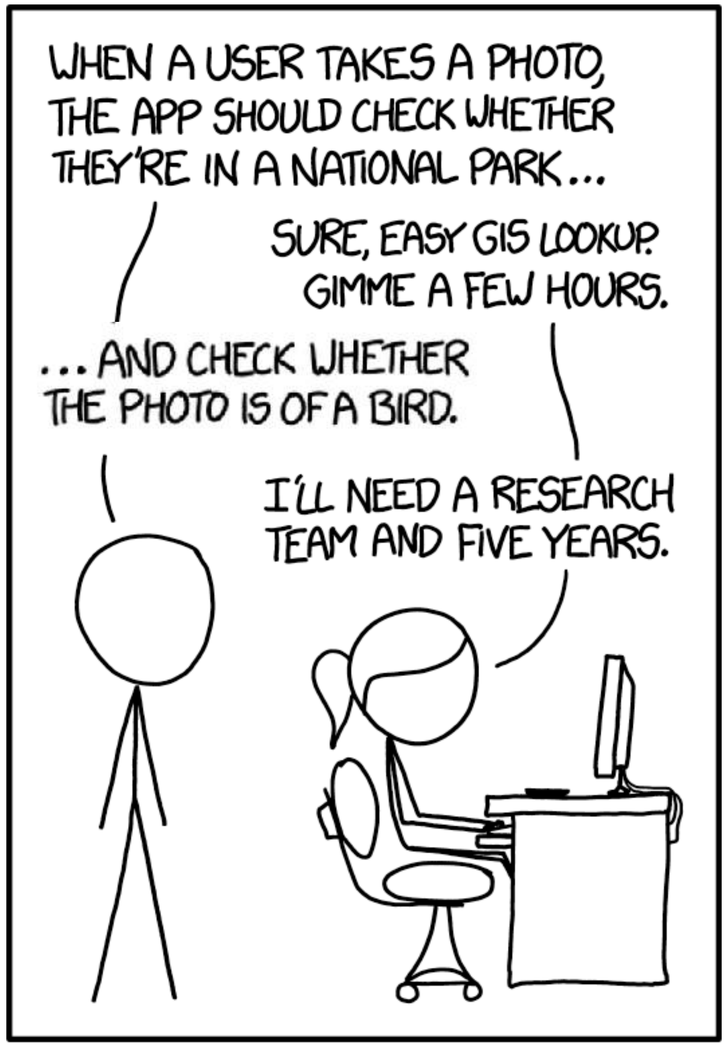
zipari, easecentral, flock, LimelightHealth, picwell, Maxwell | Health, benefix, wellthie, BeneStream



Underwriting Workbench



Reading a tons of PDFs



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

...AND CHECK WHETHER THE PHOTO IS OF A BIRD.

ChatGPT 5 Thinking

Is it a bird?

Yes — it's a bird. In fact, it looks like a male Painted Bunting (blue head, red belly, green back).

Problems good for agents

* Reading a lot of text from documents for humans
* Extracting informations (requiring some level of insight)
* Using the tools

What is an Agent?

We view agents as systems that independently accomplish tasks on behalf of users.

OpenAI

AI Agent is a program with a **predefined goal** that **uses a LLM** in a **continuous loop.**

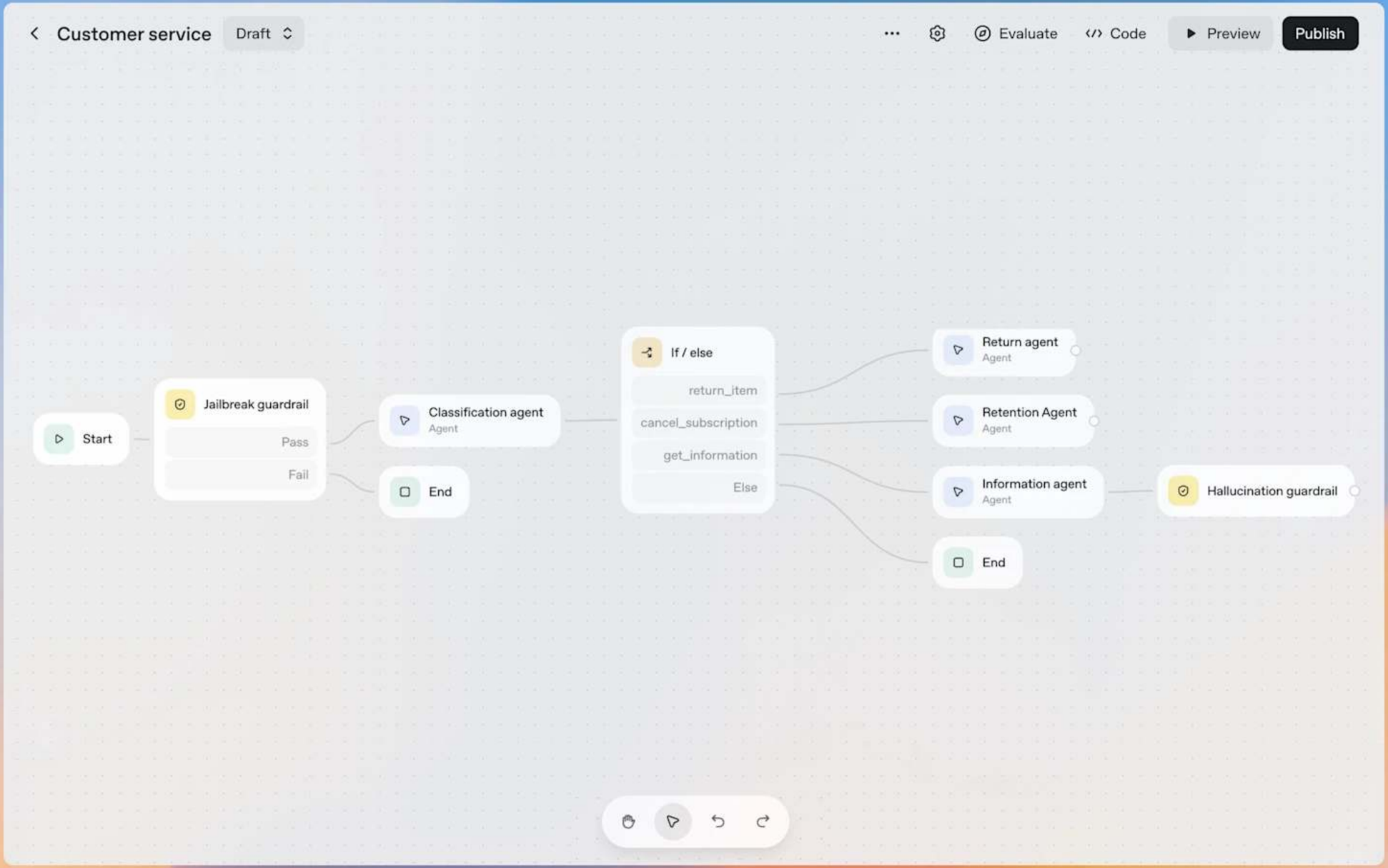
It **executes a plan** and **iterates until its objective,** such as data extraction or enrichment, is **achieved** or its **budget is exceeded.**

This **process is powered** by a set of **accessible tools,** like external APIs, IDEs, or web browsers.

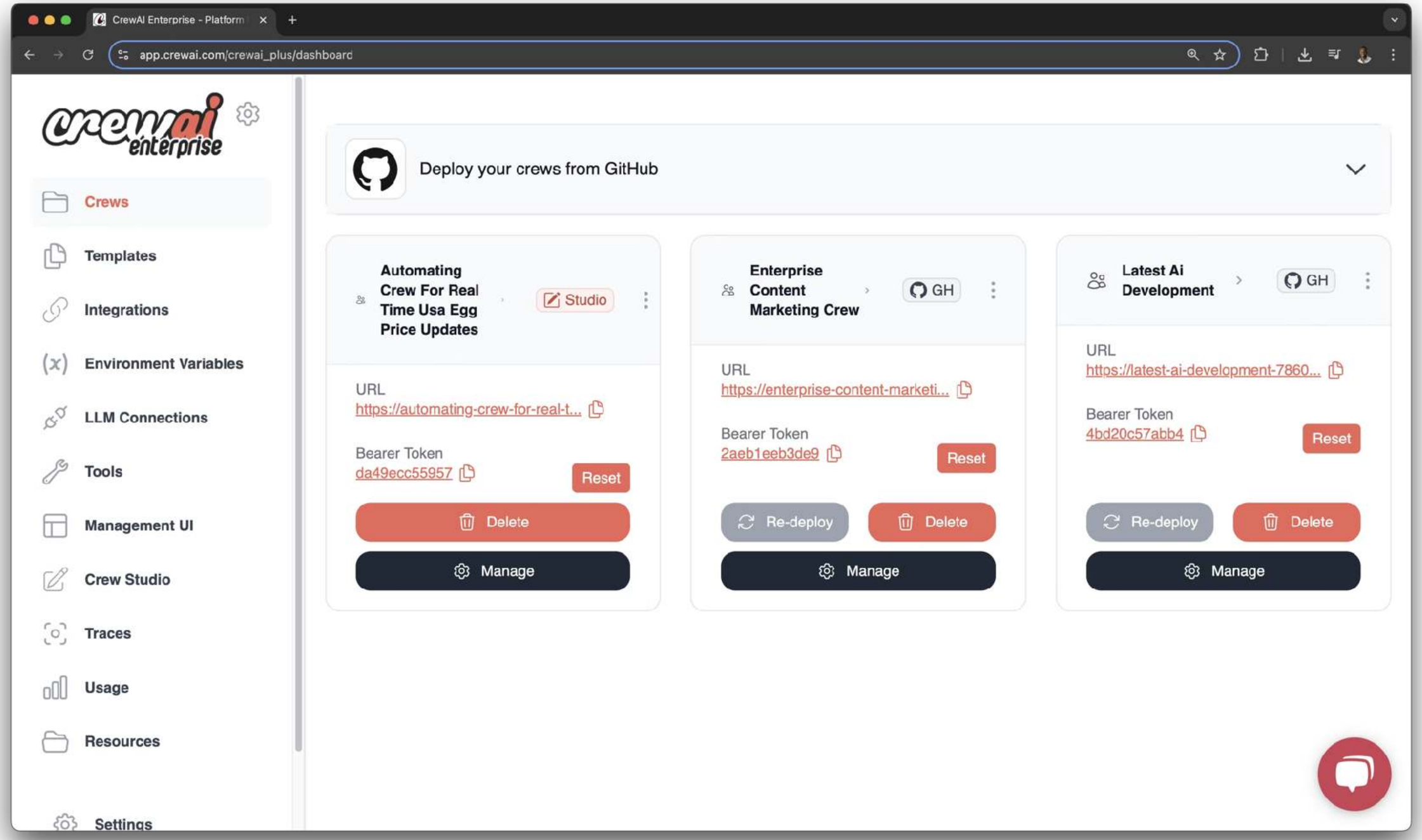
Each new LLM call **generates fresh insights,** which in turn **refine the agent's plan** and **guide its next action.**

Simon Willison

Agentic Frameworks


Customer service
Draft
Evaluate
Code
Preview
Publish
Start
Jailbreak guardrail
Pass
Fail
Classification agent
Agent
End
If / else
return_item
cancel_subscription
get_information
Else
Return agent
Agent
Retention Agent
Agent
Information agent
Agent
End
Hallucination guardrail

startup meme by memelogy.co


CrewAI Enterprise - Platform
app.crewai.com/crewai_plus/dashboard
crewai enterprise
Crews
Templates
Integrations
Environment Variables
LLM Connections
Tools
Management UI
Crew Studio
Traces
Usage
Resources
Settings
Deploy your crews from GitHub
Automating Crew For Real Time Usa Egg Price Updates
Studio
URL
https://automating-crew-for-real-t...
Bearer Token
da49ecc55957
Reset
Delete
Manage
Enterprise Content Marketing Crew
GH
URL
https://enterprise-content-marketi...
Bearer Token
2aeb1eeb3de9
Reset
Re-deploy
Delete
Manage
Latest Ai Development
GH
URL
https://latest-ai-development-7860...
Bearer Token
4bd20c57abb4
Reset
Re-deploy
Delete
Manage

Introduction
Tutorials
How-to guides
Conceptual guide
Ecosystem
LangSmith
LangGraph
Versions
v0.3
v0.2
Pydantic compatibility
Migrating from v0.0 chains
Upgrading to LangGraph memory
Release policy
Security Policy

Build an Agent

Open in Colab
Open on GitHub

LangChain supports the creation of agents, or systems that use LLMs as reasoning engines to determine which actions to take and the inputs necessary to perform the action. After executing actions, the results can be fed back into the LLM to determine whether more actions are needed, or whether it is okay to finish. This is often achieved via tool-calling.

In this tutorial we will build an agent that can interact with a search engine. You will be able to ask this agent questions, watch it call the search tool, and have conversations with it.

End-to-end agent

The code snippet below represents a fully functional agent that uses an LLM to decide which tools to use. It is equipped with a generic search tool. It has conversational memory - meaning that it can be used as a multi-turn chatbot.

In the rest of the guide, we will walk through the individual components and what each part does - but if you want to just grab some code and get started, feel free to use this!

End-to-end agent
Setup
Jupyter Notebook
Installation
LangSmith
Tavily
Define tools
Using Language Models
Create the agent
Run the agent
Streaming Messages
Streaming tokens
Adding in memory
Conclusion

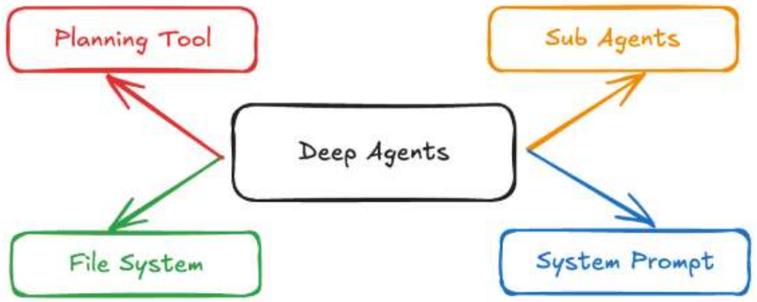
LangChain

README MIT license

🧠🤖Deep Agents

Using an LLM to call tools in a loop is the simplest form of an agent. This architecture, however, can yield agents that are “shallow” and fail to plan and act over longer, more complex tasks.

Applications like “Deep Research”, "Manus", and “Claude Code” have gotten around this limitation by implementing a combination of four things: a **planning tool**, **sub agents**, access to a **file system**, and a **detailed prompt**.



`deepagents` is a Python package that implements these in a general purpose way so that you can easily create a Deep Agent for your application.

Acknowledgements: This project was primarily inspired by Claude Code, and initially was largely an attempt to see what made Claude Code general purpose, and make it even more so.

Harrison Chase
hwchase17

GitHub All-Stars #1: deepagents – Architecture of Deep Reasoning for Agentic AI

Artur Skowronski
Head of Java/Kotlin Space

PUBLISHED: AUG 29, 2025 | 22 MINUTES READ

I've always claimed there's no better way to learn anything than by building something yourself... and the second-best way is reviewing someone else's code 😉.

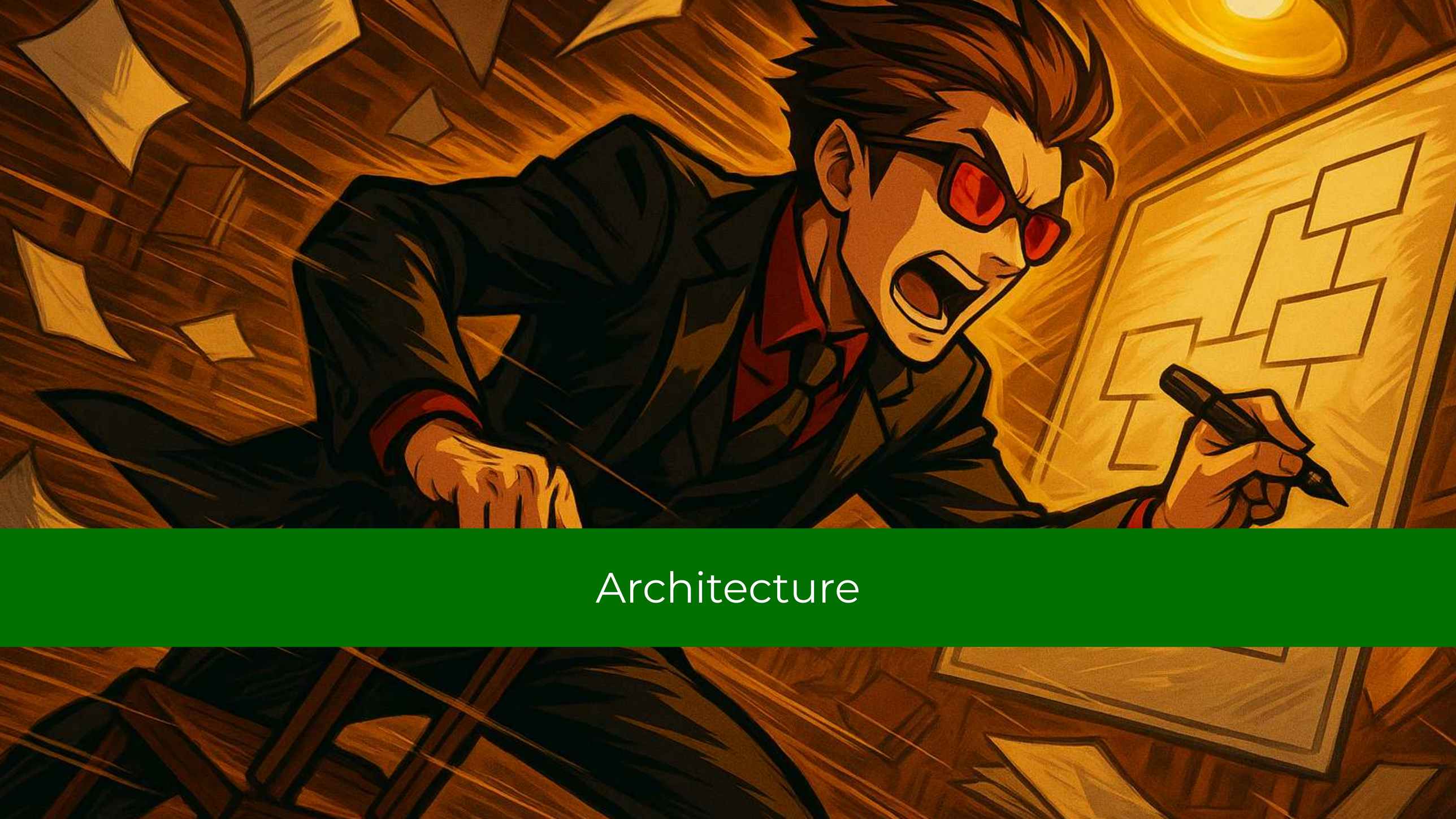
At Virtuslab, we recently had a bit of a sobering reflection. Our "collection" of GitHub starred projects had grown to a massive size, without bringing much real value to us or the wider community. So, we decided to change our approach: to add a bit of structure and start acting as chroniclers of these open-source gems. By doing this, we can understand them better - and discover the ones where we can actually contribute.

From now on, every Wednesday we'll pick one trending repo from the previous week and give it some attention: a tutorial, an article, or a code review - coming directly from its authors. We'll focus on what catches our eye: it could be a tool, a library, or anything the community has decided is worth publishing. Only one rule applies: it has to be new or relatively unknown. No rehashing the obvious projects that get thousands of stars after a big release (because honestly, who wants to hear another breakdown of Angular's architecture).

With that out of the way, let's dive into our first project: deepagents.

DeepAgents

I do not like Agent-Builders 😁 Code is a code.



Architecture

Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

Over the past year, we've worked with dozens of teams building large language model (LLM) agents across industries. Consistently, the most successful implementations weren't using complex frameworks or specialized libraries. Instead, they were building with simple, composable patterns.

In this post, we share what we've learned from working with our customers and building agents ourselves, and give practical advice for developers on building effective agents.

Single-Agent vs Multi-Agent Systems

June 12, 2025

Don’t Build Multi-Agents

By Walden Yan

Principles of Context Engineering

A Theory of Building Long-running Agents

Applying the Principles

Principles of Context Engineering

We'll work our way up to the following principles:

1. Share context

2. Actions carry implicit decisions

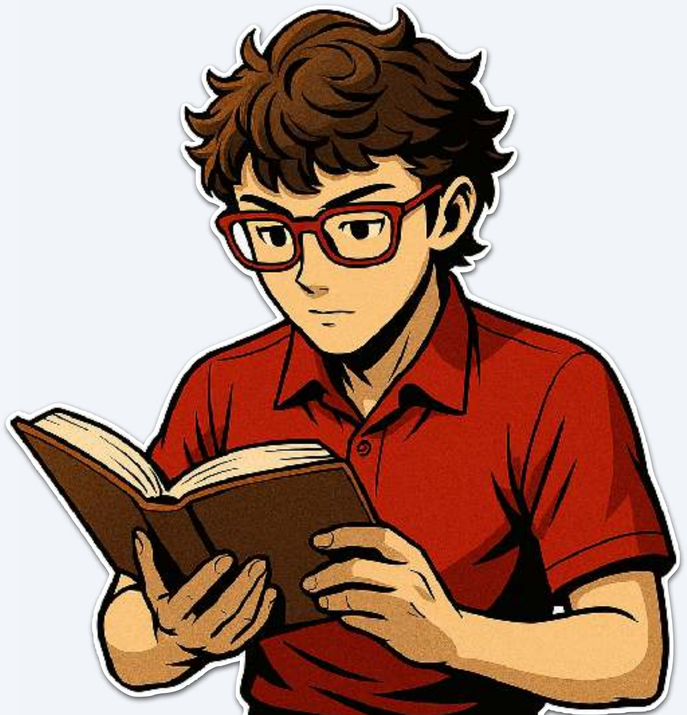
Why think about principles?

HTML was introduced in 1993. In 2013, Facebook released React to the world. It is now 2025 and React (and its descendants) dominates the way developers build sites and apps. Why? Because React is not just a scaffold for writing code. It is a philosophy. By using React, you embrace building applications with a pattern of reactivity and modularity, which people now accept to be a standard requirement, but this was not always obvious to early web developers.

In the age of LLMs and building AI Agents, it feels like we're still playing with raw HTML & CSS and figuring out how to fit these together to make a good experience. No single approach to building agents has become the standard yet, besides some of the absolute basics.

> In some cases, libraries such as https://github.com/openai/swarm by OpenAI and https://github.com/microsoft/autogen by Microsoft actively push concepts which I believe to be the wrong way of building agents. Namely, using multi-agent architectures, and I'll explain why.

That said, if you're new to agent-building, there are lots of resources on how to set up the basic scaffolding [1] [2]. But when it comes to building serious production applications, it's a different story.



The cognitive leap

How to reimagine work with AI agents and multiagent systems

Continuing our series on AI agents and multiagent systems, this report outlines the principles of agentic architecture and design to help organizations make the cognitive leap into a new paradigm of business process transformation and innovation.

View the full report

Key takeaways about agentic architecture and design

- Multiagent AI systems can help transform traditional, rules-based business and IT processes into adaptive, cognitive processes.
- Organizations should leverage key principles of AI agent and multiagent AI system design and management, which borrow from tenets of composable design, microservices architecture and human resources deployment and teaming.
- The ability to scale AI agent frameworks across a range of use cases depends on developing a comprehensive reference architecture populated with reusable core components.
- A systematic approach can make the difference between incremental, isolated improvements and exponential enterprise transformation.

Deloitte. Who we are | What we do | Our Thinking | Careers | Search | US - EN

The cognitive leap

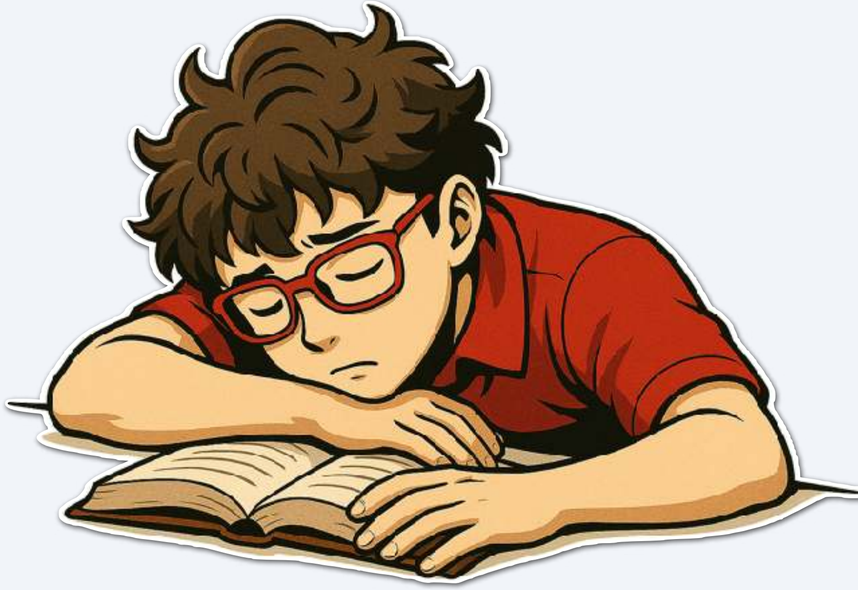
How to reimagine work with AI agents and multiagent systems

Continuing our series on AI agents and multiagent systems, this report outlines the principles of agentic architecture and design to help organizations make the cognitive leap into a new paradigm of business process transformation and innovation.

View the full report

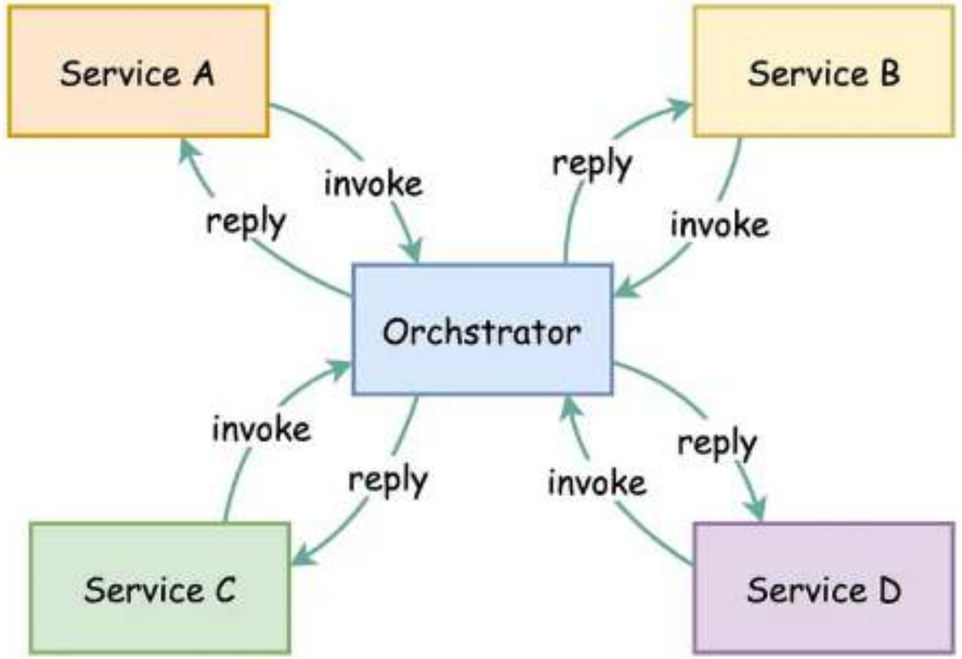
Key takeaways about agentic architecture and design

- Multiagent AI systems can help transform traditional, rules-based business and IT processes into adaptive, cognitive processes.
- Organizations should leverage key principles of AI agent and multiagent AI system design and management, which borrow from tenets of composable design, microservices architecture and human resources deployment and teaming.
- The ability to scale AI agent frameworks across a range of use cases depends on developing a comprehensive reference architecture populated with reusable core components.
- A systematic approach can make the difference between incremental, isolated improvements and exponential enterprise transformation.

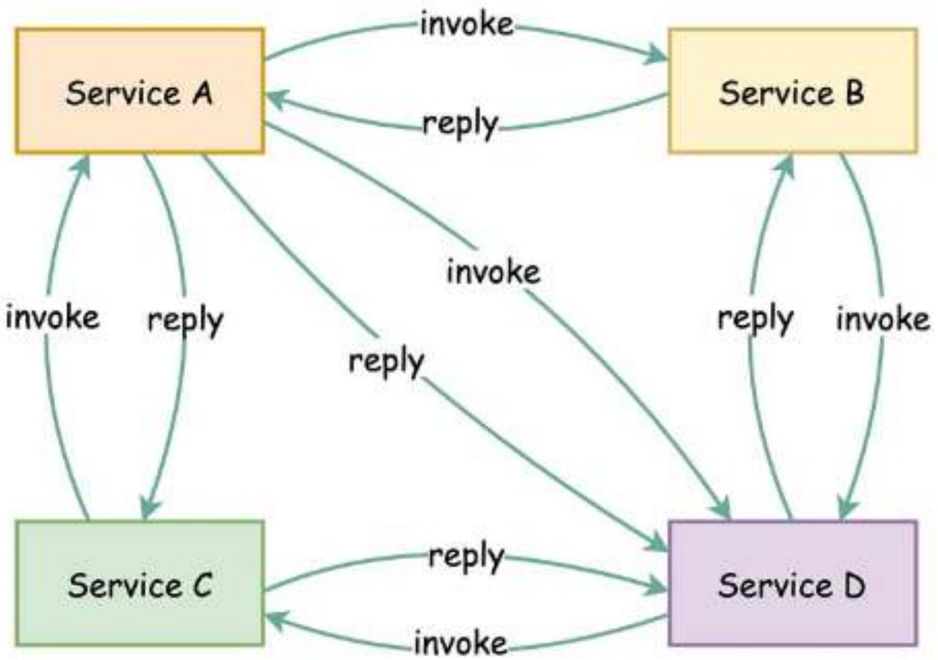


I HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO CLOUD COMPUTING.
Dilbert.com DilbertCartoonist@gmail.com
BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD.
1·7·11 ©2011 Scott Adams, Inc./Dist. by UFS, Inc.
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLAH BLAH PLATFORM.

Orchestration



Choreography

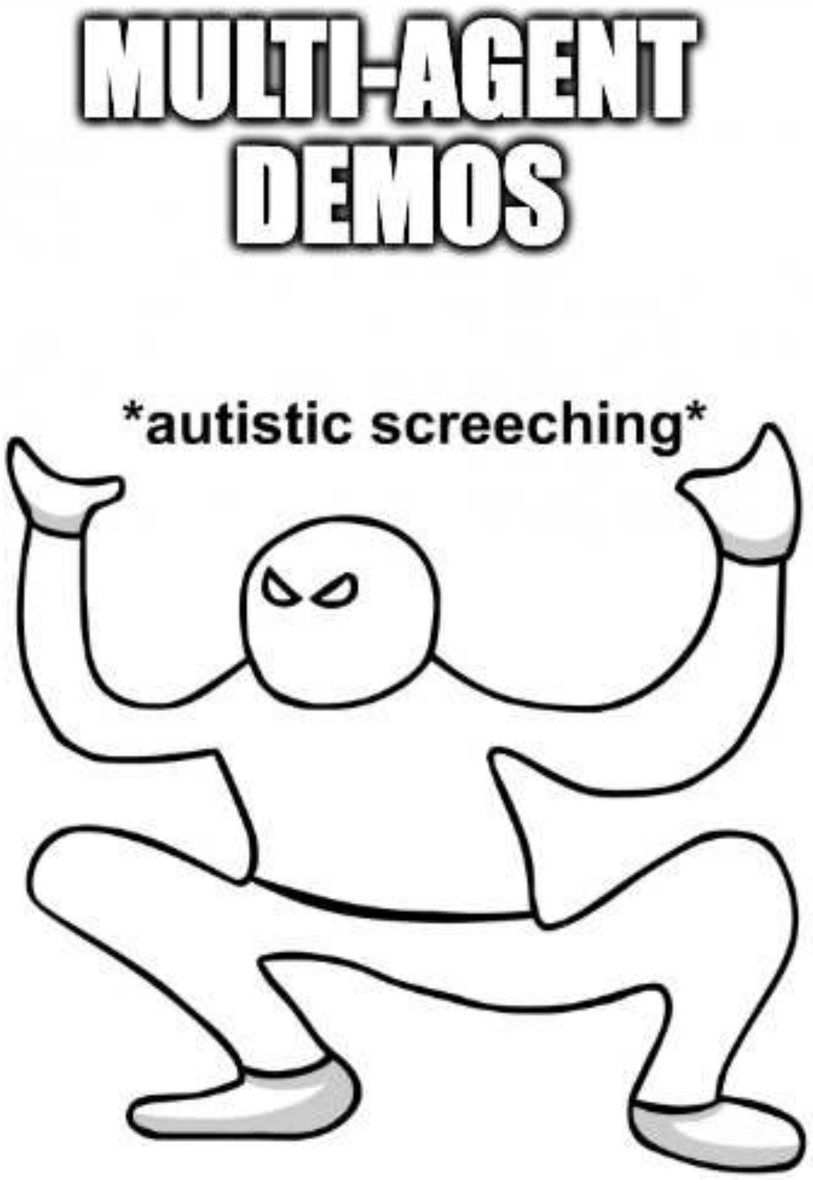


Choerography is good… for research systems!

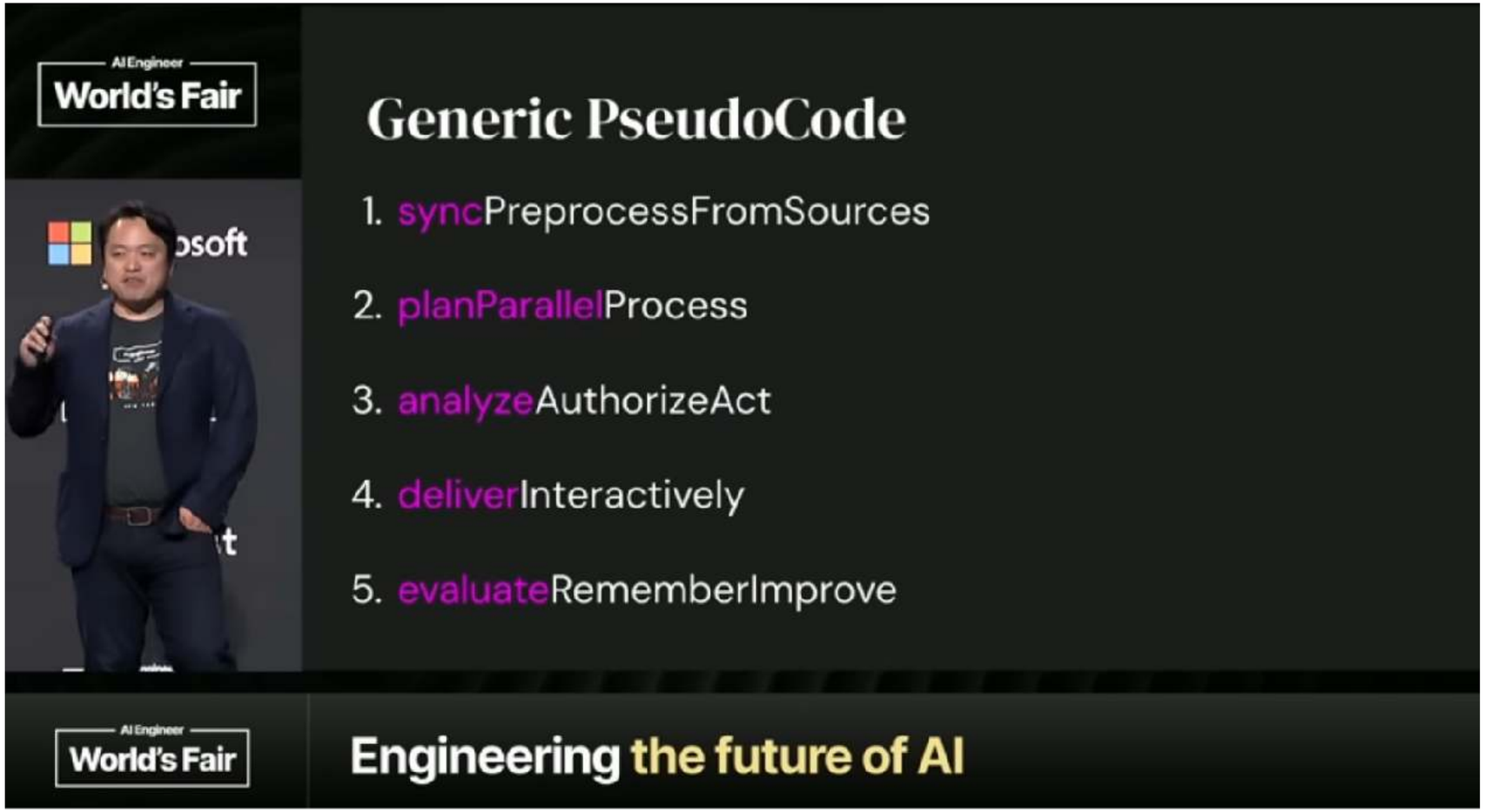
In business apps - if you can Orchestrate - do this!

REGISTERED → INGESTED → EXTRACTED_PENDING → EXTRACTED_APPROVED → ENRICHED → TRIAGED → PRICED → SUMMARY_READY → **DELIVERED**

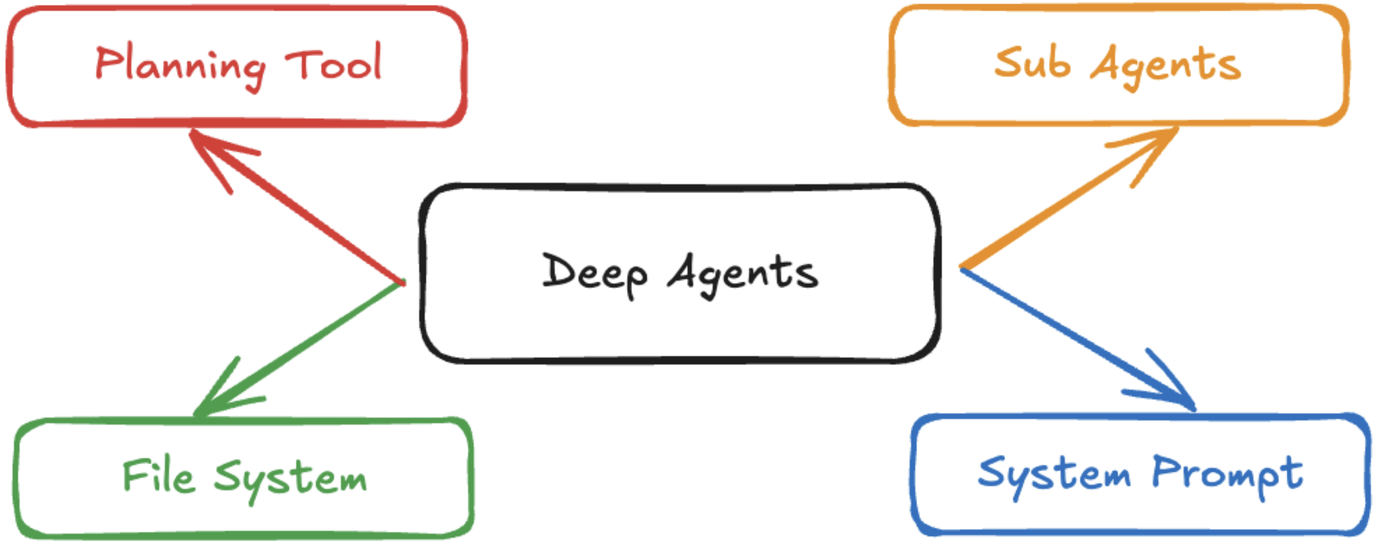
REAL AGENTIC APPLICATIONS
LET'S DO WORKFLOW WITH SUBAGENTS
imgflip.com


MULTI-AGENT DEMOS
autistic screeching

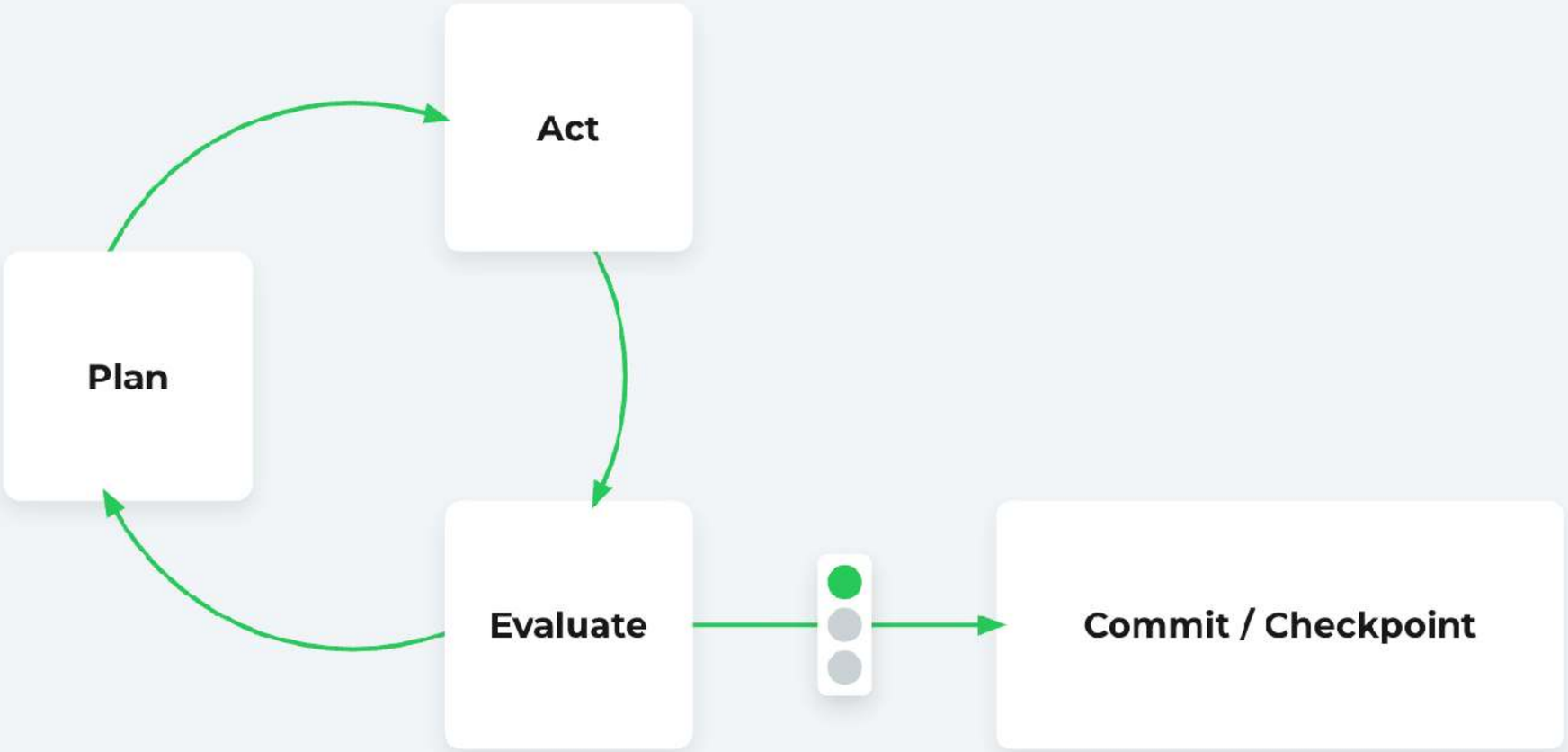
Agentic Loop



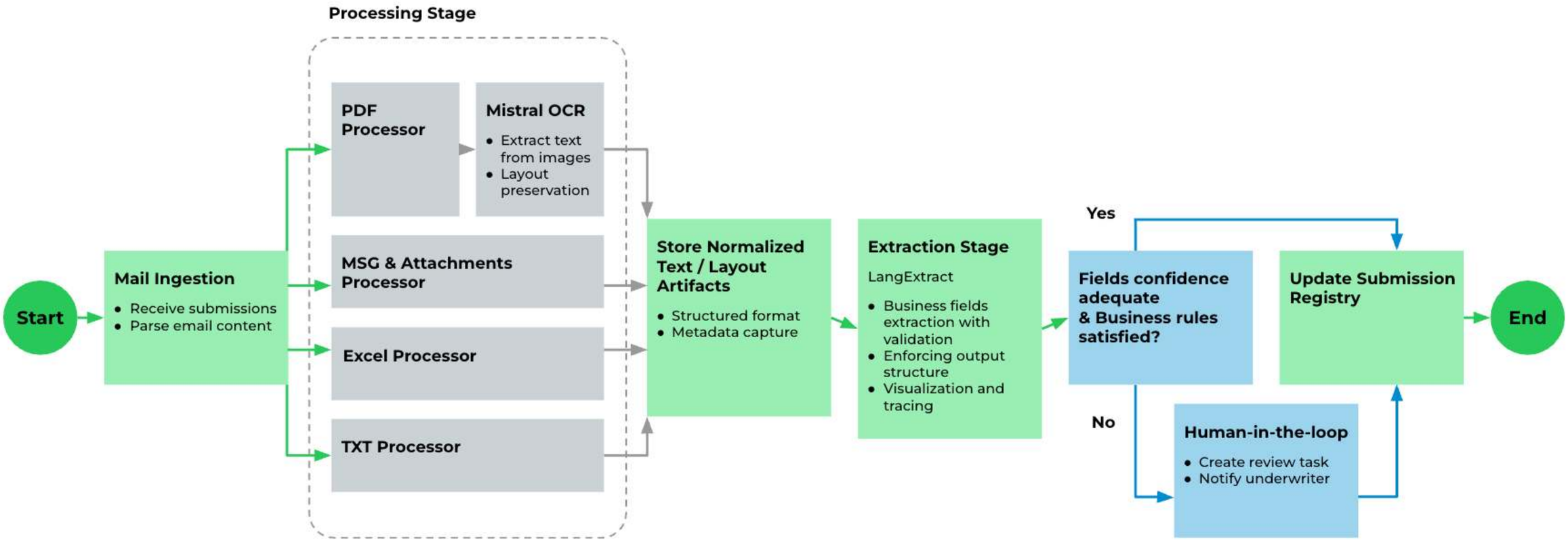
SPADE by swyx



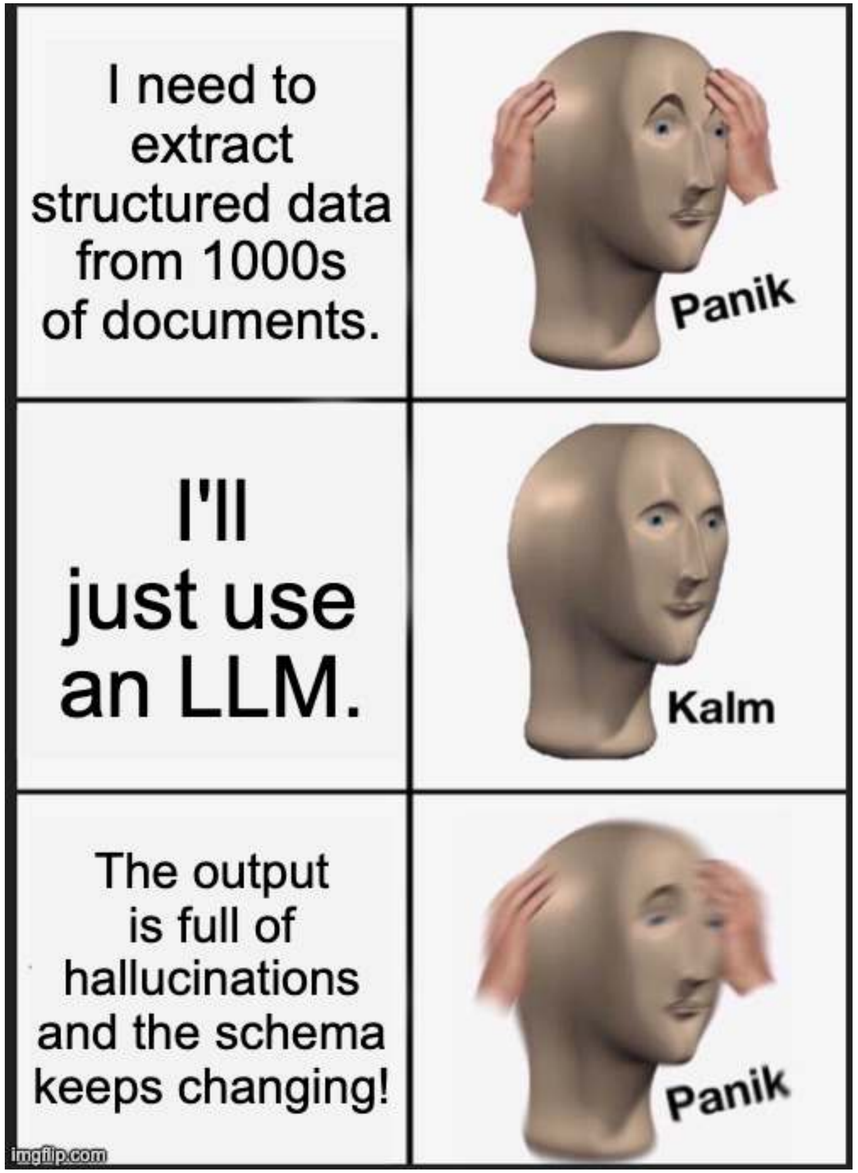
Deep Agents / Claude Code Architecture



SPADE by swyx



Extraction Engine



Structured Output

AGENTIC AI
ECONOMY
Reasoning
LLMs
PROMPT:
"Please Bro, response in valid JSON
format without errors and make sure
the syntax is super correct.
I'm begging you man...."

LangExtract

pypi v1.0.9 | Star 17k | CI passing | DOI 10.5281/zenodo.17015089

Table of Contents

- Introduction
- Why LangExtract?
- Quick Start
- Installation
- API Key Setup for Cloud Models
- Adding Custom Model Providers
- Using OpenAI Models
- Using Local LLMs with Ollama
- More Examples
  - *Romeo and Juliet* Full Text Extraction
  - Medication Extraction
  - Radiology Report Structuring: RadExtract
- Community Providers
- Contributing
- Testing
- Disclaimer

Introduction

LangExtract is a Python library that uses LLMs to extract structured information from unstructured text documents based on user-defined instructions. It processes materials such as clinical notes or reports, identifying and organizing key details while ensuring the extracted data corresponds to the source text.

HOME • BLOG • AI • LANGEXTRACT WITH GEMINI

GitHub All-Stars #3: LangExtract – How Google Turns Chaos into Data with Gemini

Artur Skowroński
Head of Data/AI in Space

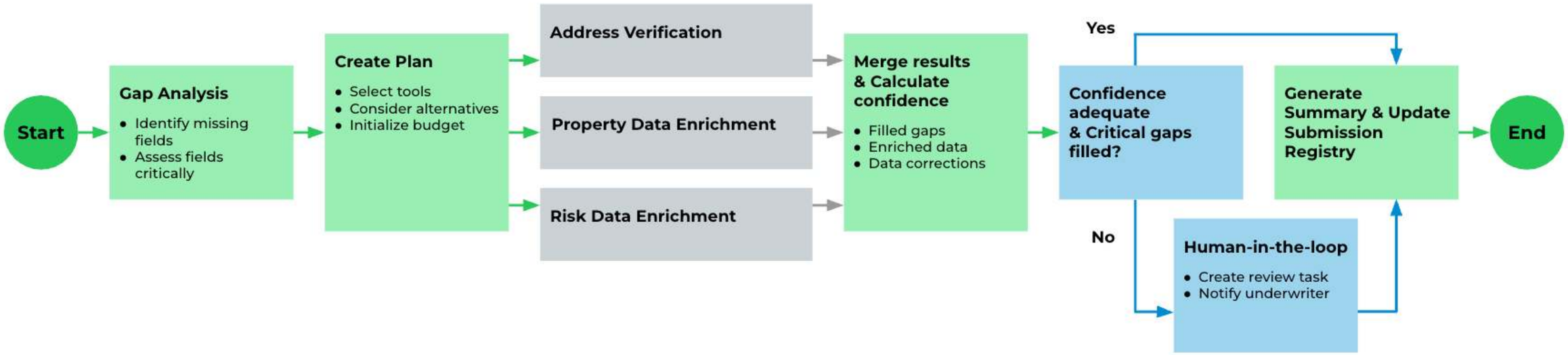
PUBLISHED: SEP 10, 2025 | 12 MINUTES READ

I've always claimed that there's no better way to learn anything than to build something... and the second-best is to review someone else's code 😉. A sobering thought struck us recently at VirtusLab—our "collection" of starred projects had grown to a gigantic size, bringing little real value either to us at VirtusLab or to the wider community. So we decided to change our approach: introduce a bit of regularity and become chroniclers of these open-source gems. That way, we'll understand them better and discover the ones where we can genuinely help.

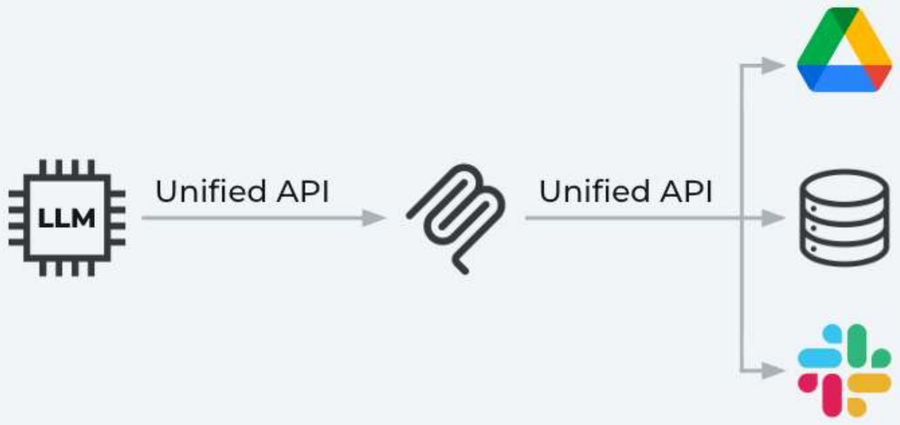
Every Wednesday, we'll pick one trending repository from the previous week and give it attention, preparing a tutorial, article, or code review—learning from its creators along the way. We'll focus on what piques our interest: it might be a tool, a library, or anything the community deems worth publishing. One simple rule applies—it has to be a new or little-known project, not the widely recognized ones that rack up tons of stars after a big update (because let's be honest... who wants to hear about Angular's architecture in 2025).

Today, we're taking a fresh project from Google engineers to the bench: Google/langextract.

LangExtract



Enrichment Engine



MCP standardizes how agents discover, request, and use tools and data

Normalize access

One contract for tools, data sources, and actions

Enforce policy

Gate every call with auth, rate limits, budgets, and allow/deny lists

Keep LLMs swappable

Because interfaces are stable, you can replace models/agents without touching integrations

MCP Tooling

With Tool Calling you want to have choreography

US Insurance Tech Market Map

Health

oscar, Clover, bright health, DevotedHealth, Gusto, Namely, Collective Health, decisely, JUSTWORKS., Lumity, Stride Health, Hixme, bind, GoHealth, JOANY, HealthSherpa, GRAVIE, SimplyInsured, allay, beam, emerge, GetInsured, HealthCare.com, friday

Home, condo & renters

Jetty, Hippo, kin., SURE., Rhino, LeaseLock, TheGuarantors, Matic, gabi, swyft, Covered, Quilt

Auto

metromile, ROOT, the zebra, INSHUR, EVERQUOTE, clearcover, COVERHOUND, insurify, go, Goji

Title

Qualia, SPRUCE, STATES TITLE

Pet

FIGO

Drone

verifly

Personal item

Cover, trov

Small commercial

nextinsurance, coverwallet, EMBROKER, insureon, JONES, FOUNDERSHIELD, CORVUS, PIE INSURANCE, Huckleberry, QBIS, Slice

Cyber

PALADIN Cyber, Coalition, a b

Life & annuity

Ladder, bestow, tomorrow, Health I.Q., Fabric, policy genius, MIRA, ensurem, covr, BLUEPRINT INCOME

Crop

CROP PRO, WorldCover

B2B

P&C

snapsheet, Groundspeed, CARPE DATA, Weather Analytics, datacubes, Zeguro

Life

iPipeline, Atidot, vivametrica, sureify, ClaimVantage

Health

zipari, easecentral, flock

We started to build ecosystem of partners for the MCP

Agentic AI in Underwriting - #LATechWeek

Hosted by

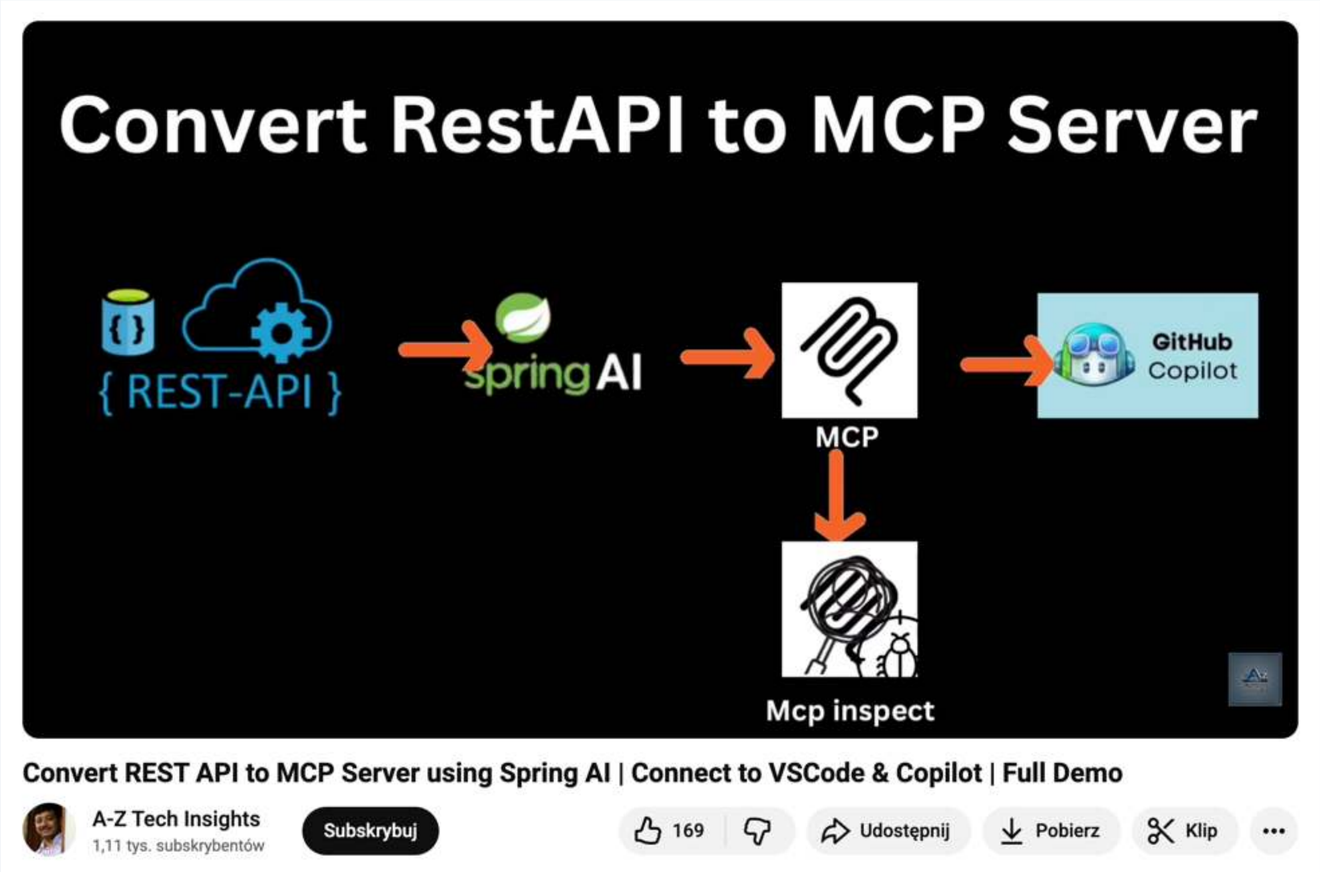
Agentic AI in Underwriting (LDN)

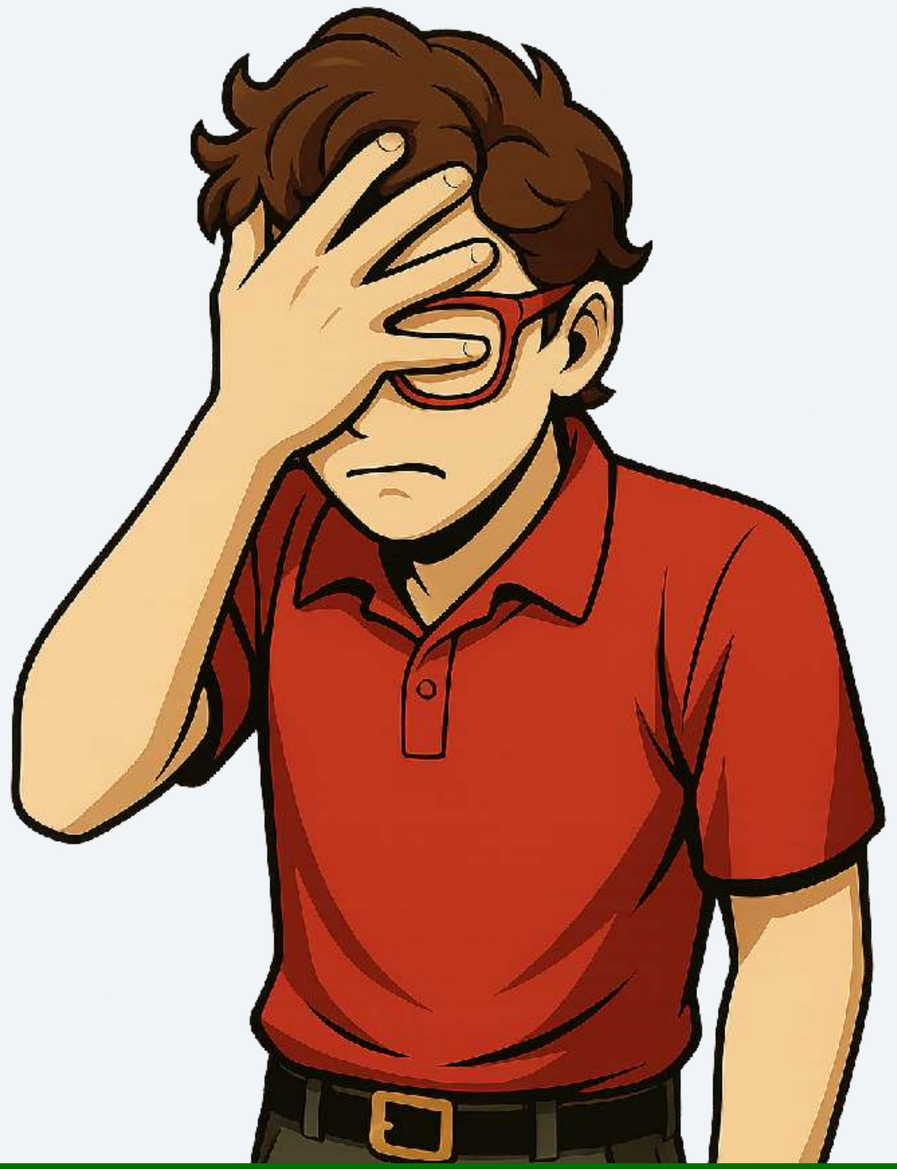
Hosted by +2

Agentic AI in Underwriting #SFTechWeek

Hosted by

We started to build ecosystem of partners for the MCP


Convert RestAPI to MCP Server
{ REST-API }
spring AI
MCP
GitHub Copilot
Mcp inspect
Convert REST API to MCP Server using Spring AI | Connect to VSCode & Copilot | Full Demo
A-Z Tech Insights
1,11 tys. subskrybentów
Subskrybuj
169
Udostępnij
Pobierz
Klip



REST TO MCP?

Strict contracts for tools: version your schemas **(tool name/version number, JSON Schemas for input/output),** and introduce breaking changes only with a new tool_id@v2.

Proper approach

You want to limit non-determinism!

In MCP, you do not want to expose REST resources - you need to expose whole end-to-end actions that can be done in full.

Agents are subotimal for flows, where the **order of step** is required.

Make tool calls idempotent, accept client-generated request IDs, and return deterministic results for the same inputs.

Proper approach

The more deterministic you are, the better

Human-in-the-Loop

Search (⌘ + k)

Launches · Products · News · Forums · Advertise · Subscribe · Sign in

1. Magic Patterns — Design your product ideas with AI.

4.9 (35 reviews)

Used by 4: Waydev AI · HYBRD · iago · View all

Overview · Shoutouts · Reviews · Launches

Magic Patterns is an AI design tool for software product teams. Proc
Magic Patterns to rapidly prototype and create interactive mockups

Prototype your product ideas with AI

AI Designer

View Details

2. Visily — AI-powered app wireframing & prototyping

4.9 (81 reviews)

Used by 2: Portia AI · Lock-in · View all

Overview · Shoutouts · Reviews · Launches

Visily is the easiest UX ideation platform that enables teams, regard
hi-fi app wireframes & prototypes. Through an intuitive interface, ric
has helped thousands of non-designers such as founders, product
developers quickly build their app UX.

Shoutouts Leaderboard

The most-loved products on Product Hunt

Product Landscapes

AI notetakers · Content Management Systems · Compliance software · Hiring software · Password managers · See more

- Productivity
- Engineering & Development
- Design & Creative
- Finance
- Social & Community
- Marketing & Sales
- Health & Fitness
- Travel
- Platforms
- Product add-ons
- AI Agents
- LLMs
- Web3
- Physical Products
- Voice AI Tools
- Ecommerce
- No-code Platforms
- Data analysis tools

AI Agents

- AI Agent Automation
- AI Chief Of Staff
- AI Data Scientist
- AI Designer
- AI Engineer
- AI SDR
- AI Voice Agents

g Spoons

y iPhone

1 year ago

about A.I.
to ask?

Featured

3 years ago

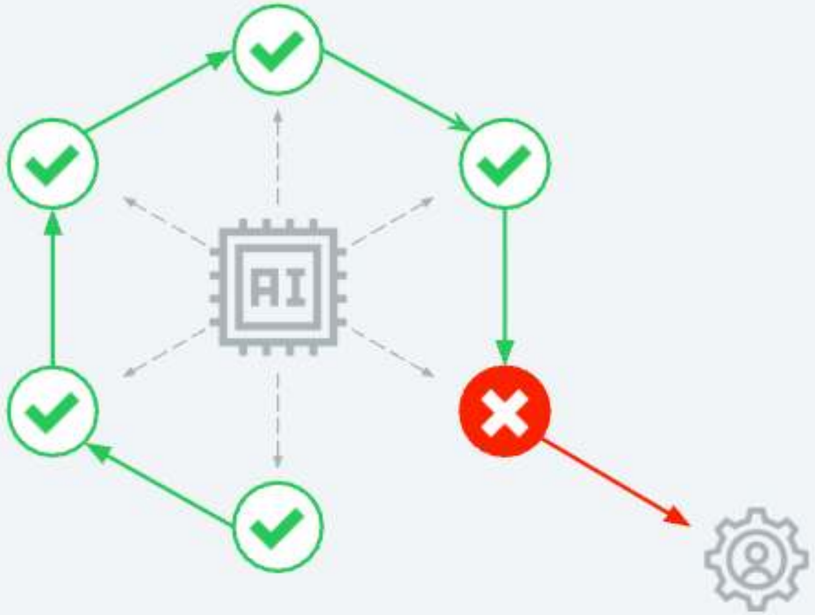
ChatGPT 5 Thinking >

more dynamic

Thought for 5s
Image created

Ask anything

5 Thinking



An LLM agent runs tools in a loop to achieve a goal. Each agent is a small decision engine that must justify its actions with evidence and respect strict budgets.

If confidence or budget fails, we close the loop and call a human.

Green

Auto-advance

Fully Algorithmic

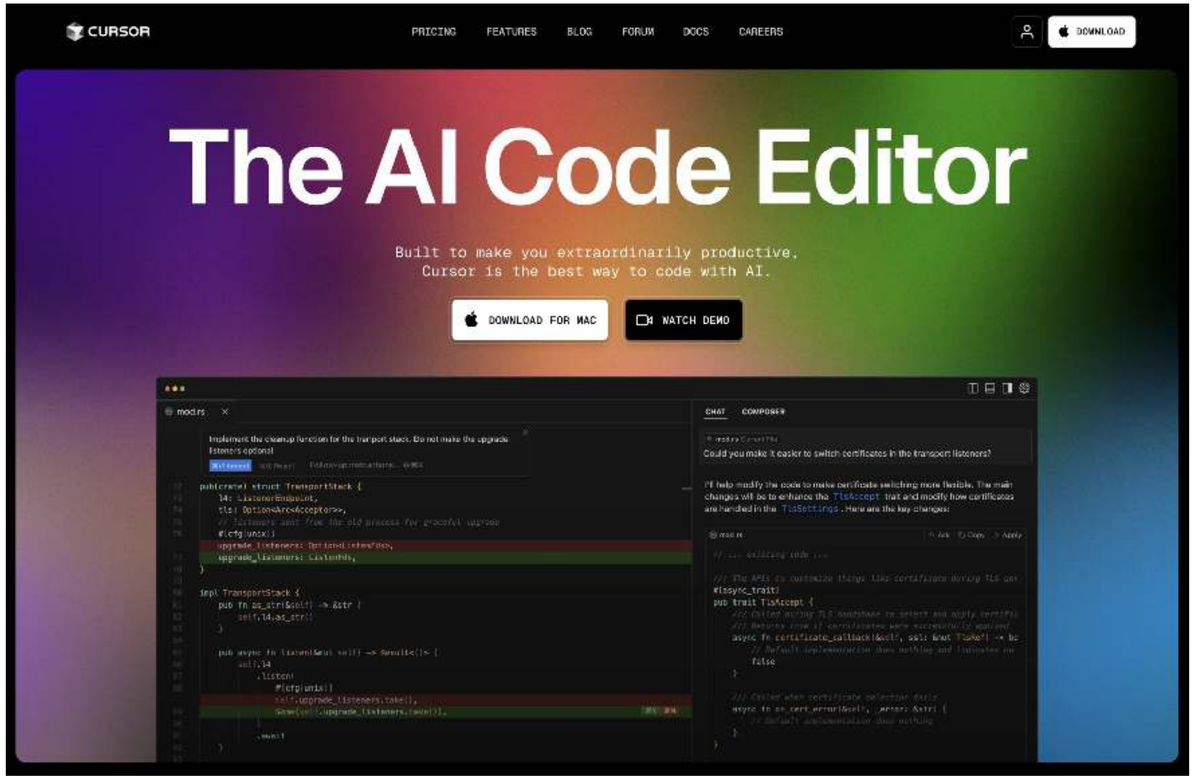
Yellow

Human-in-the-loop with suggested fix to approve

Red

Escalation to Human-in-the-loop engine without recommendations

- We measure precision and recall.
- We run controlled A/B tests when we change thresholds.
- We focus on shrinking the yellow band while keeping red extremely rare.



Type Systems

Unit Tests

1 billion public repositories

What the “good” means?

I have 30 years of experience... I know what “good” means

What the “good” means?

Confidence lower than expected threshold

Rule violation

Missing evidence

Budget breach

Deterministic evaluator

Checks act as a fast, explicit filter:

1. First we verify completeness (required fields → score).
2. Then apply policy rules that may hard-stop or soft-flag a case, based on the RAG data and historical Human operator decisions.
3. Finally call external tools (fraud services) for structured signals - predictable, quick, and explainable.

LLM as judge

Evaluates agent proposals by other LLM.

Good for nuance, natural-language rationale, one-line fixes and explainability.

Hybrid flow

Fast deterministic checks run first to catch hard failures.

LLM Judge then scores nuance and suggests fixes.

- Low confidence scores for data extraction (< 85%)
- High-value submissions exceeding configurable thresholds
- Complex risk scenarios requiring expert judgment
- Anomalous patterns detected by AI models
- Regulatory compliance requirements mandating human oversight
- First-time clients or new business types

Too few false positives

Risk creeps in

Too many false positives

System never scales

Human-in-the-loop quality define ROI

Grounding / Provenance

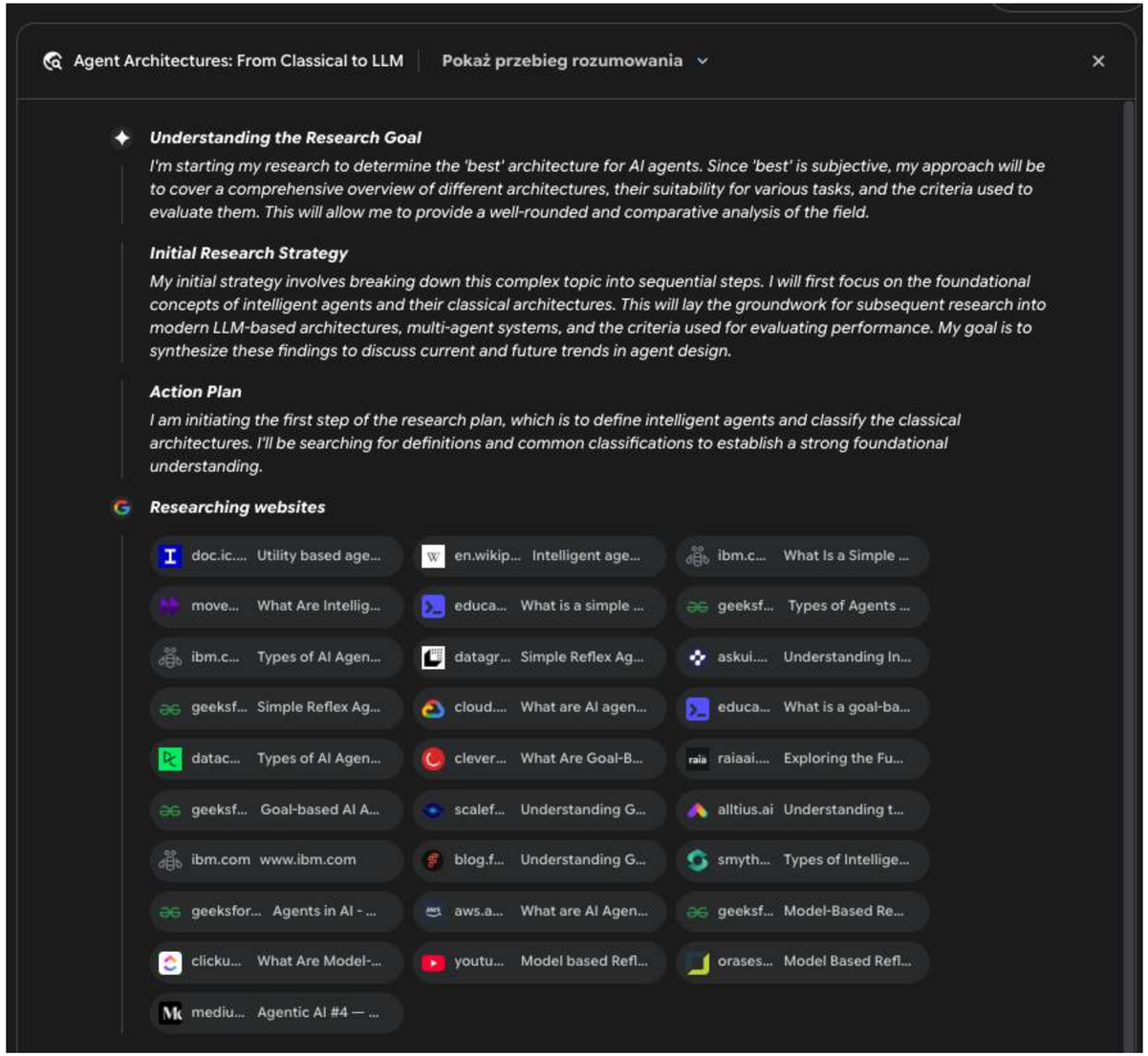
Confidence lower than expected threshold

Rule violation

Missing evidence

Budget breach

Grounding / Provenance



Grounding / Provenance

1.1 The PEAS Framework (Performance, Environment, Actuators, Sensors)

To formally describe and analyze an agent's purpose and context, the PEAS framework is employed. This framework specifies the four critical components of an agent's task:

- **Performance Measure:** An objective function that evaluates the agent's success. It defines what it means for the agent to do the "right thing"[1]. For a robotic vacuum cleaner, the performance measure might be the amount of dirt cleaned, the time taken, and the energy consumed.

  en.wikipedia.org

  en.wikipedia.org/wiki/Inte...

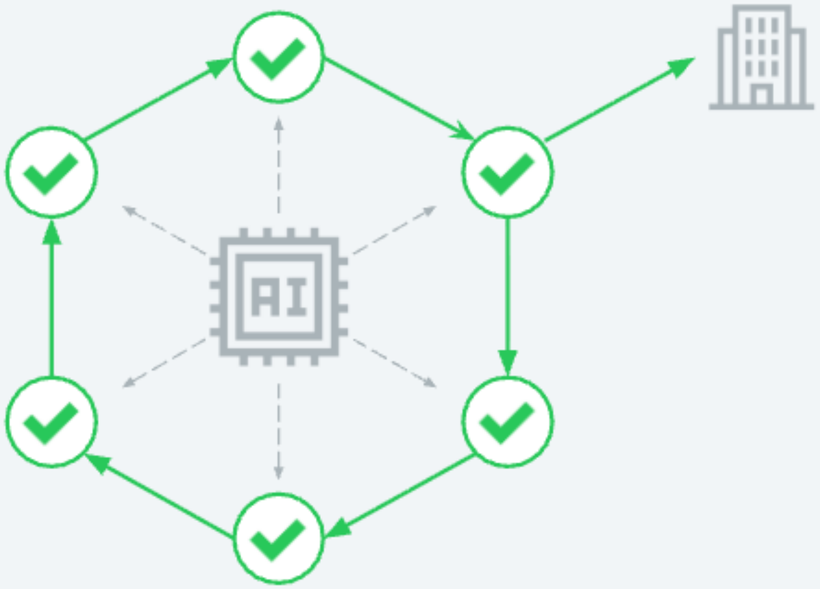
- **Environment:** The context in which the agent perceives and acts. Environments can be physical, like the roads for a self-driving car, or digital, like the financial markets for a trading bot.
- **Actuators:** The mechanisms through which the agent executes actions and affects the environment. For a software agent, actuators could be API calls or messages; for a robot, they are motors and grippers.
- **Sensors:** The tools the agent uses to perceive the state of the environment. These can range from physical cameras and microphones to digital data feeds and APIs.

The PEAS framework provides a precise specification for the agent's problem, which is the first step in designing an appropriate architecture.

Grounding / Provenance

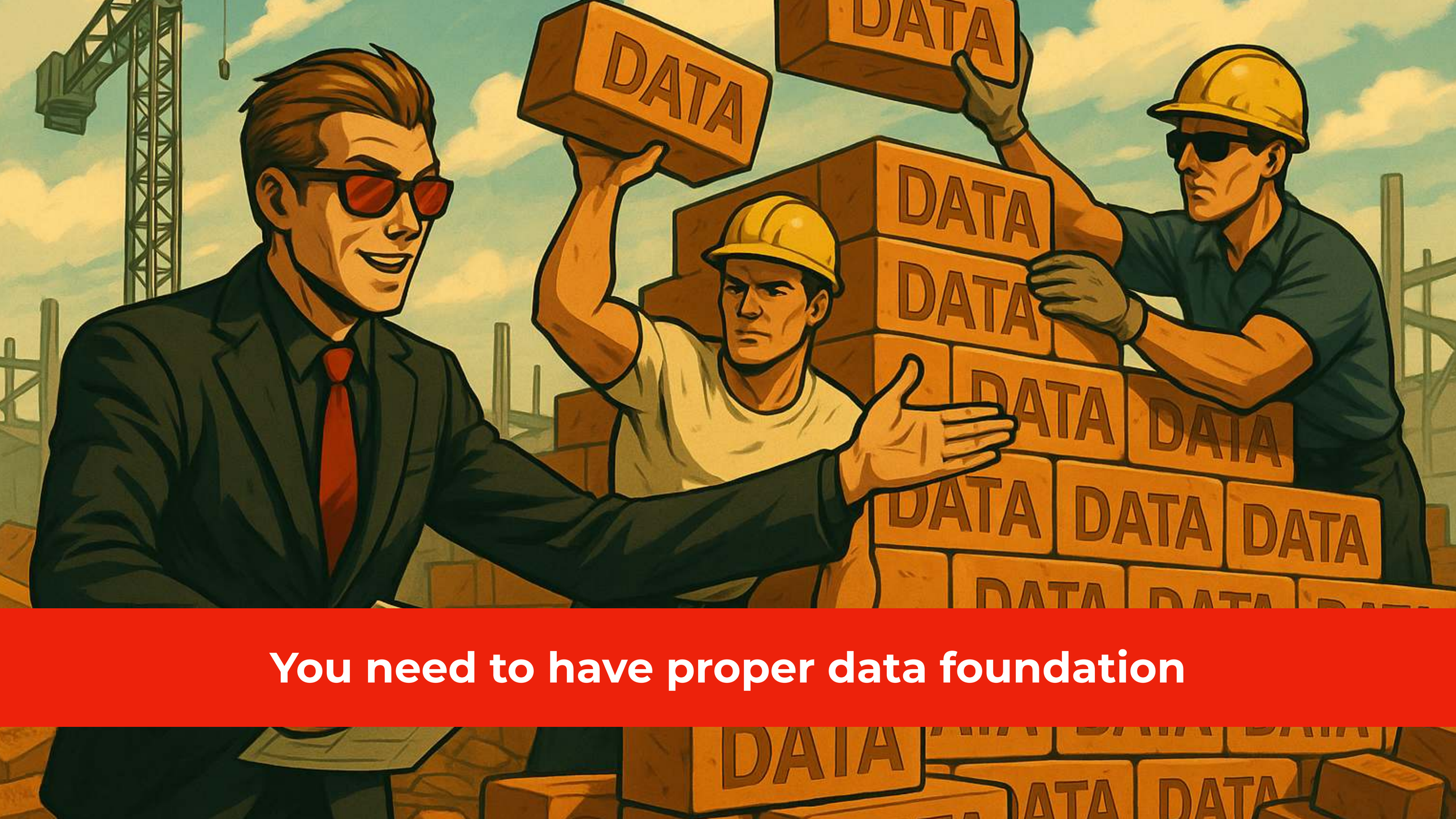
Save state of “the game”… always

Master Golden Set - previous knowledge

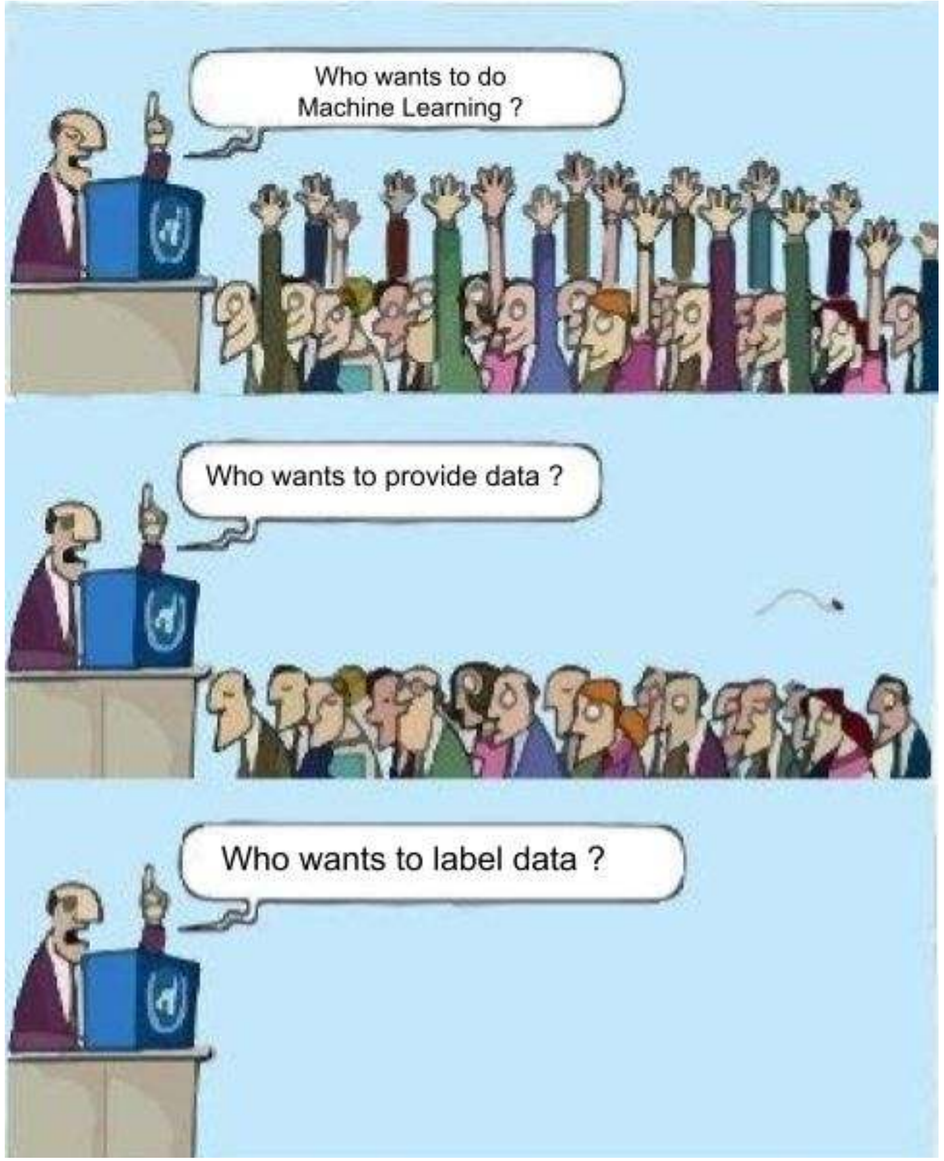


RAG-aware evaluator

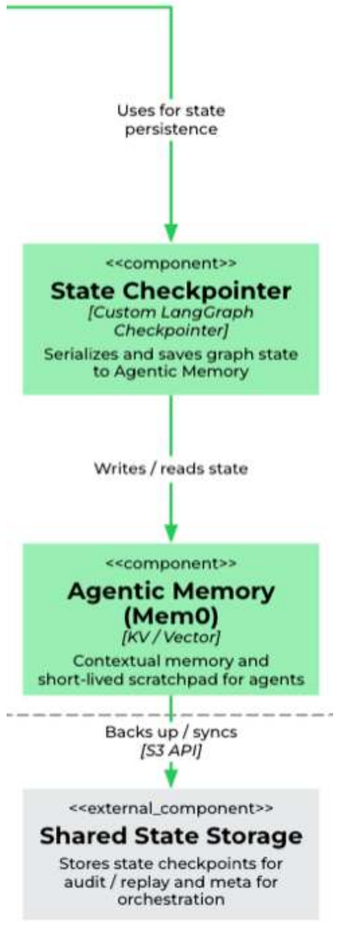
1. Stores past underwriting cases, scores and operator decisions.
2. Lets evaluators compare current proposals with historical outcomes.
3. Supports explainability: “this decision aligns with similar case X.”



You need to have proper data foundation


Who wants to do
Machine Learning ?
Who wants to provide data ?
Who wants to label data ?

Memory



Why do you need memory?

· Coordination between agents
· The “Lost in the Middle” problem: A model’s attention quality drops with very long contexts
· Optimizing the budget
· RAG to the existing dataset

README Contributing Apache-2.0 license

mem0 | The Memory Layer for Personalized AI

GITHUB TRENDING
#1 Repository Of The Day

Learn more · Join Discord · Demo · OpenMemory

Discord | downloads 1.3M/month | commit activity 48/month | pypi package v1.0.0 | npm v2.1.38 | Y Combinator S24

📄 Building Production-Ready AI Agents with Scalable Long-Term Memory →

⚡ +26% Accuracy vs. OpenAI Memory • 🚀 91% Faster • 💰 90% Fewer Tokens

> 🎉 **mem0ai v1.0.0 is now available!** This major release includes API modernization, improved vector store support, and enhanced GCP integration. See migration guide →

🔥 Research Highlights

- **+26% Accuracy** over OpenAI Memory on the LOCOMO benchmark
- **91% Faster Responses** than full-context, ensuring low-latency at scale
- **90% Lower Token Usage** than full-context, cutting costs without compromise
- Read the full paper

Introduction

GitHub All-Stars #2: MemO – Creating memory for stateless AI minds

Artur Skowroński
Head of Java/Kotlin Space

PUBLISHED: SEP 3, 2025 | 17 MINUTES READ

I've always claimed there's no better way to learn anything than to build something... and the second best is to review someone's code. As part of our GitHub All-Stars series, where we examine open-source gems, I stumbled upon a project that strikes at the heart of one of the most fundamental problems in modern AI.

After analyzing deepagents and its system-level approach to reasoning, the natural next step was to examine how we solve the problem... of memory.

So we're continuing the trend that treats models as a Boring Black Box and shows that the interesting things happen around them in Context Engineering.

Artificial Intelligence Amnesia: Why we need better memory

I'll start with a hot (or perhaps more lukewarm) take... the current development path for AI agents, based on blindly chasing ever-larger context windows, is a dead end. It's a brute-force approach that only delays the problem instead of solving it.

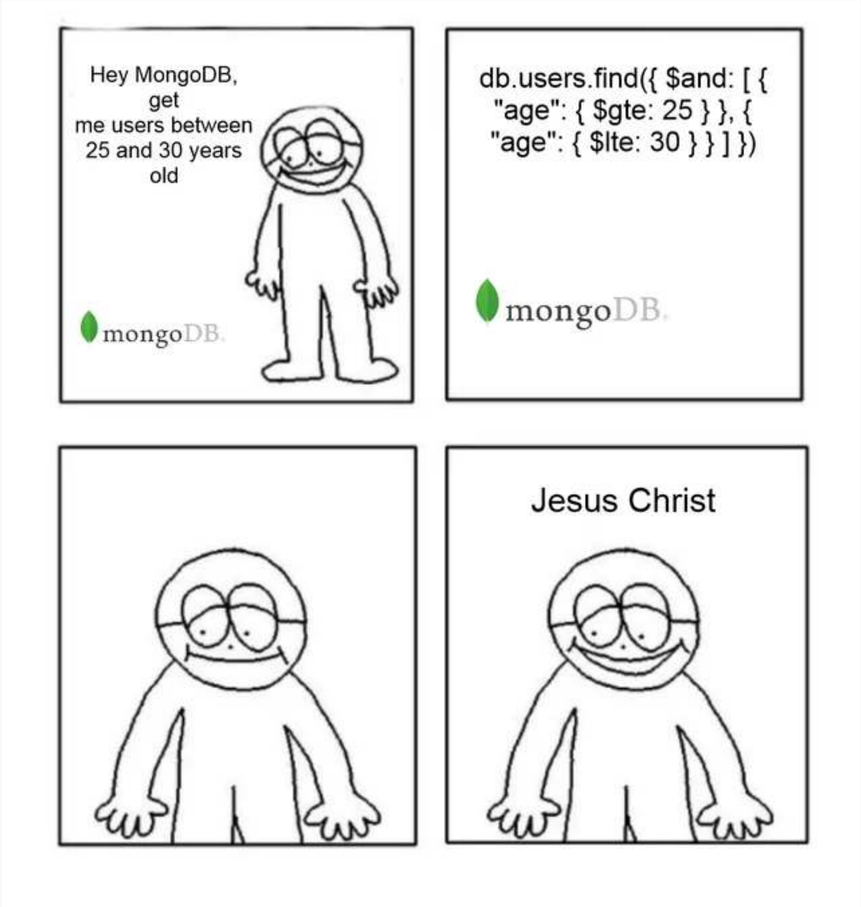
Mem0

```python
# The agent-assistant fetches key client info from the past
from mem0 import MemoryClient

# Initialize the memory client
mem0 = MemoryClient(api_key="your-mem0-api-key")

def get_underwriting_context(policy_application: dict, client_id: str):
    """Fetch historical context for a given client from mem0."""
    # Query the memory for the client's claim history
    query = f"What is the claim history for client {client_id}?"
    past_claims = mem0.search(query, user_id=client_id)

    # Format retrieved memories for use in the prompt
    serialized_history = ' '.join([mem["memory"] for mem in past_claims])
    # ... further logic to build the agent prompt
```

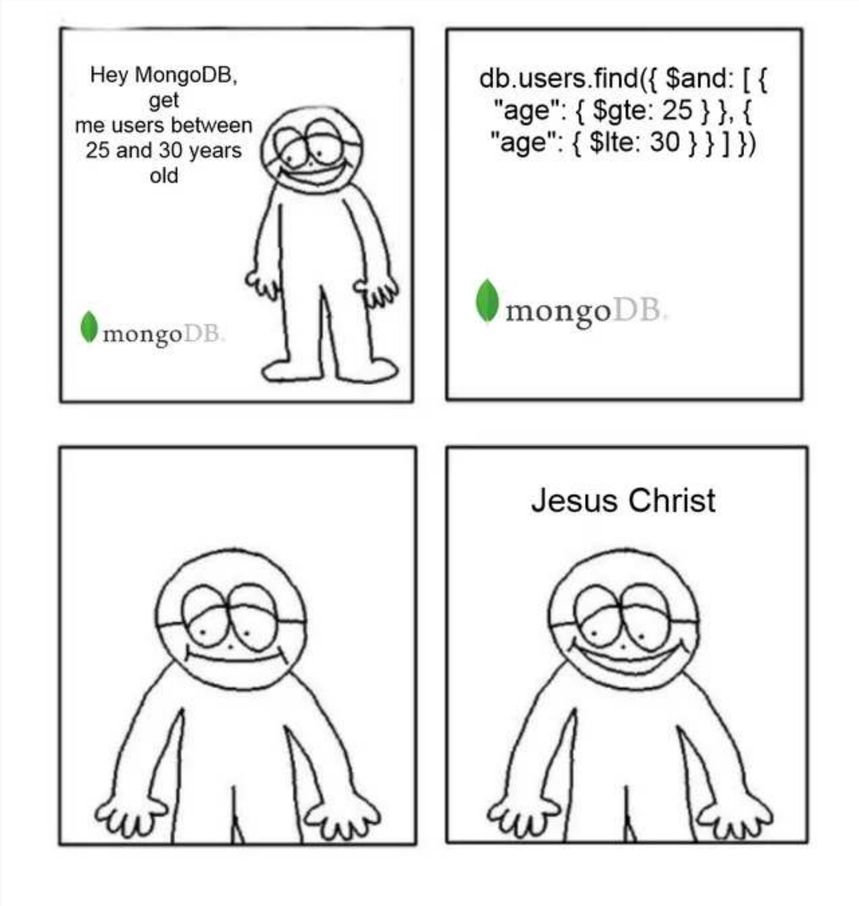
Scratchpads

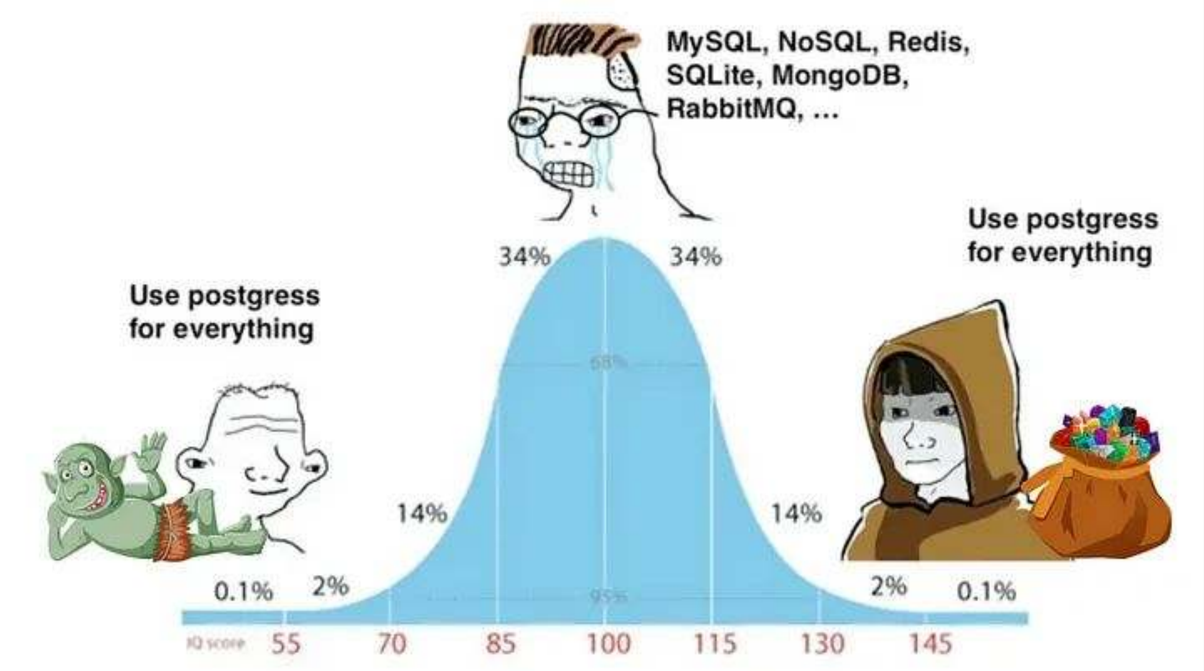
```python
from mem0 import Memory
from langchain_community.vectorstores import Chroma
from langchain_openai import OpenAIEmbeddings

# Initialize connection to the company's existing vector DB of risk reports
embeddings = OpenAIEmbeddings()
risk_reports_db = Chroma(
    persist_directory="./corporate_risk_reports_db",
    embedding_function=embeddings,
    collection_name="mem0" # Required collection name
)
# Configure mem0 to use the existing DB
config = {
    "vector_store": {
        "provider": "langchain",
        "config": {
            "instance": risk_reports_db
        }
    },
    # ... other configs, e.g., pointing to the corporate LLM
}
memory_component = Memory.from_config(config)
```

Vector Database Abstraction

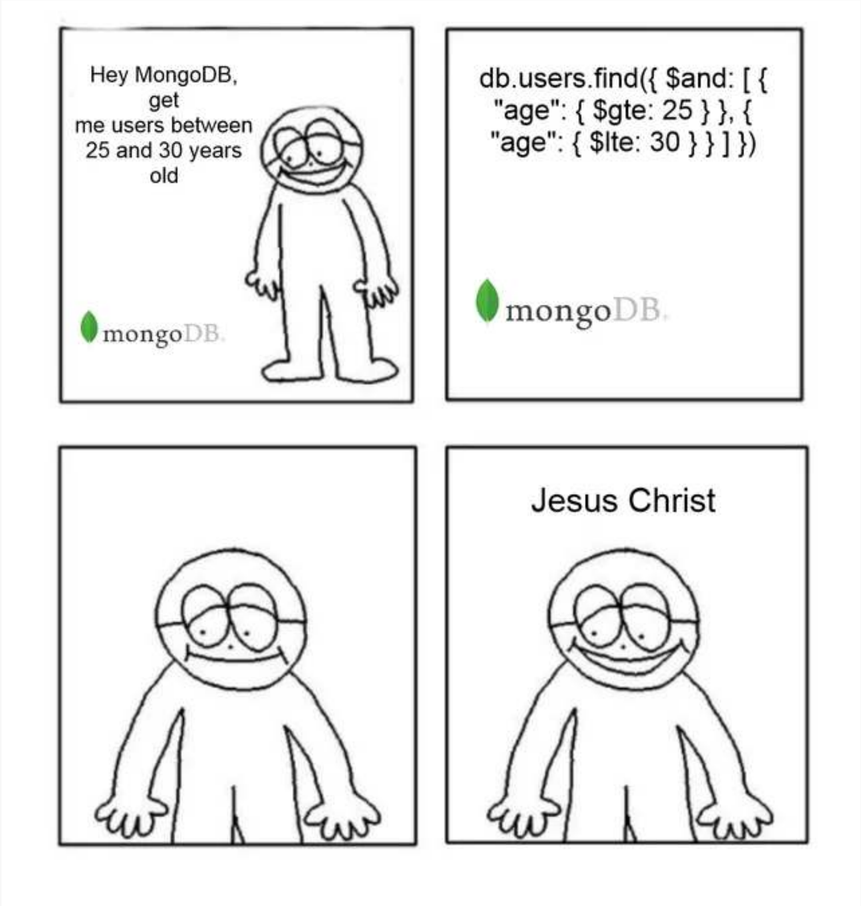


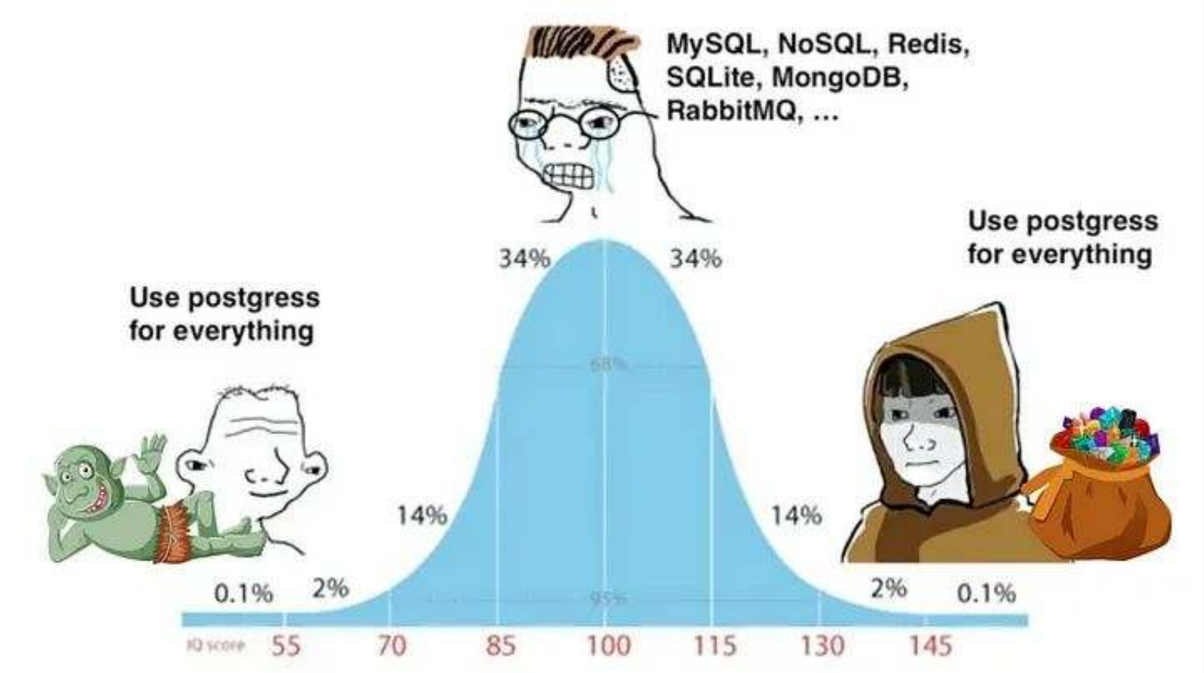
Vector Database Abstraction





Vector Database Abstraction

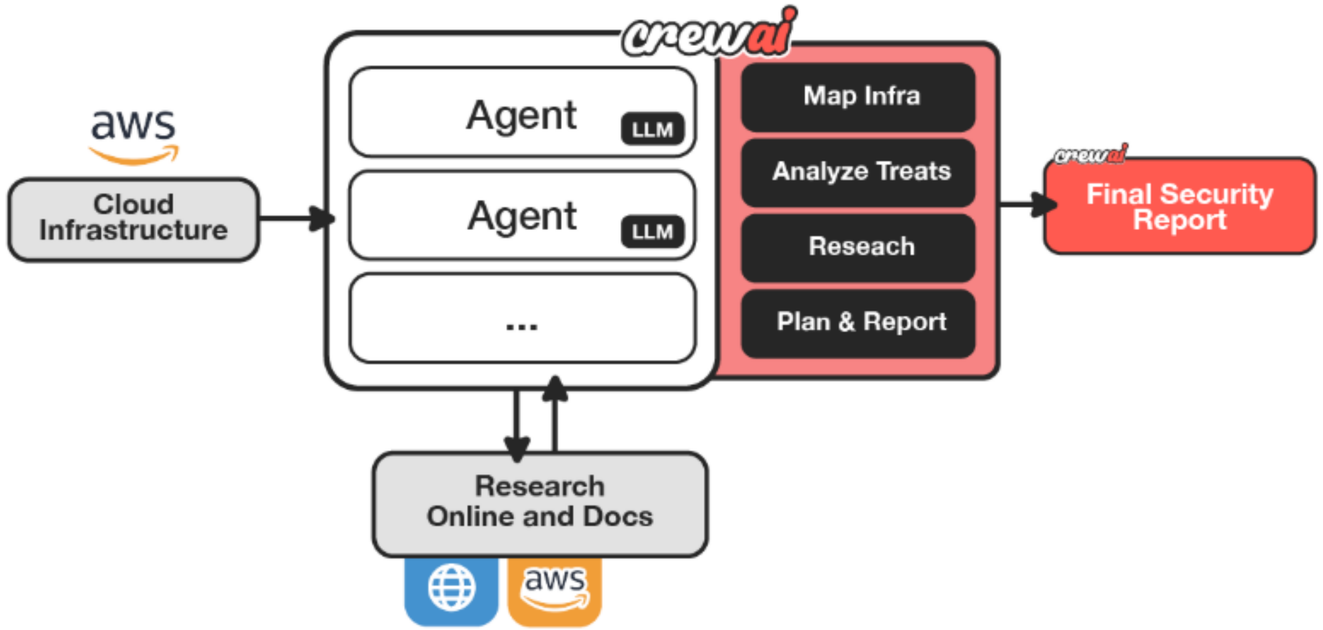




neo4j

Vector Database Abstraction

Production



kafka

mongo DB

Production

Artificial Intelligence

Build agentic systems with CrewAI and Amazon Bedrock

by Tony Kipkemboi, João (Joe) Moura, Karan Singh, and Aris Tsakpinis | on 31 MAR 2025 | in Amazon Bedrock, Amazon Bedrock Agents, Generative AI | Permalink | Comments | Share

This post is co-authored with Joao Moura and Tony Kipkemboi from CrewAI.

The enterprise AI landscape is undergoing a seismic shift as agentic systems transition from experimental tools to mission-critical business assets. In 2025, AI agents are expected to become integral to business operations, with Deloitte predicting that 25% of enterprises using generative AI will deploy AI agents, growing to 50% by 2027. The global AI agent space is projected to surge from $5.1 billion in 2024 to $47.1 billion by 2030, reflecting the transformative potential of these technologies.

In this post, we explore how CrewAI's open source agentic framework, combined with Amazon Bedrock, enables the creation of sophisticated multi-agent systems that can transform how businesses operate. Through practical examples and implementation details, we demonstrate how to build, deploy, and orchestrate AI agents that can tackle complex tasks with minimal human oversight. Although "agents" is the buzzword of 2025, it's important to understand what an AI agent is and where deploying an agentic system could yield benefits.

Production

Streamlit

Amazon Bedrock

docker

Production

Convenient tables, charts, code highlighting, PDF/image preview, multi-page apps: You can quickly build a UI to explore collections, vector stores, retrieval results, and passage rankings.

Scaling and concurrency
Can serve as a thin UI layer over the actual runtime (FastAPI + queue + DB). Lets you validate UX and scope before investing in full infrastructure.

As an operator panel: view agent steps, prompts, responses, and artifacts (tables, charts, markdown). Makes prompt debugging, chain A/B tests, and decision explainability easy.

Production

Streamlit is designed for short, interactive requests: Agentic workflows (LLM + tools + retries + callbacks + RAG) are often long-running and require background work, queues, resumability, and SLAs. Streamlit lacks a native job queue, schedulers, retries with backoff, and durable “runs” that are independent of the browser session.

Scaling and concurrency
The “session = in-process memory state” model makes horizontal scaling difficult and requires sticky sessions. Per-user RAM usage grows, and the lack of backpressure and controlled worker-level concurrency hurts stability under load (especially when agents call expensive models/tools).

Agents often react to external events (webhooks, queues, cron). Streamlit is not an event server: accepting webhooks, long-lived websockets from tools, event sourcing, or multi-agent orchestration requires a separate backend and a queue, which undermines Streamlit’s simplicity.

Production

Akka Agentic Platform

Build, run, and evaluate with certainty, scale, and resilience.

Sensors Metrics
Humans Agents

Akka Orchestration

Guide, moderate, and control long-running systems.

Akka Memory

Durable, in-memory and sharded data.

Akka Agents

Create agents, MCP tools, and HTTP/gRPC APIs.

Akka Streaming

High performance stream processing.

Agents Vector DB
OLAP Transactional DB
Models Message Brokers
LLMs APIs & Tools

Production

Akka Agentic Platform

Build, run, and evaluate with certainty, scale, and resilience.

Sensors Metrics
Humans Agents

Akka Orchestration

Guide, moderate, and control long-running systems.

Akka Memory

Durable, in-memory and sharded data.

Akka Agents

Create agents, MCP tools, and HTTP/gRPC APIs.

Akka Streaming

High performance stream processing.

Agents Vector DB
OLAP Transactional DB
Models Message Brokers
LLMs APIs & Tools

Production

Akka Agentic

Build, run, and evaluate with certa

Akka Orchestration

Guide, moderate, and contro
long-running systems.

Akka Agents

Create agents, MCP tools, a
HTTP/gRPC APIs.

Enablement

Documents | Process Analysis | Summary

Step 1: Intake and Data Extraction

completed | Duration: 8 minutes | 80% confidence

Replay

All submitted documents have been thoroughly analyzed and verified for authenticity and completeness. The system has extracted key data points including property valuations, financial metrics, and risk indicators. Document integrity checks have been performed using advanced algorithms to ensure data accuracy.

Policy Quote Data

Effective Date	2021-03-01
Insured Name	Jane Doe
Mailing Address	100 Turnpike Street, Shrewsbury, MA
State	MA
Broker Email	janedoe@gmail.com [source]
Account Number	00009550 [source]
Expiration Date	2026-03-25 [source]
Zip Code	02721 [source]

Sov Location

Location Number	1-1
Construction Type	Frame
Occupation Type	Retail
Number Of Stories	2
Year Built	1986

Assessment Assistant

Can you invoke data enrichment step again? This time using Smarty as the data provider?

04:33 PM

Of course 👌 I'll re-run the enrichment workflow for this submission, this time using Smarty as the address and property data provider. You'll receive the updated enriched dataset in just a moment.

04:35 PM

CopilotKit

Build **User-Facing** Agentic Applications

Launch fast, customize effortlessly.

LangGraph crewai mastra Agno LlamaIndex AI SDK AG2

Bring your own framework No framework

Version v1.10.6 License MIT Discord 282 online

PRODUCT HUNT
#2 Product of the Day

⚡ Quick Install

```
npx copilotkit@latest init
```

Read the Docs → Try Copilot Cloud → Join our Discord →

🚀 Getting Started

1. Install: Run a simple CLI command
2. Configure: Add CopilotKit provider to your app
3. Customize: Use headless UI or the customizable pre-built components
4. Deploy: You're done!

GitHub All-Stars #4 - CopilotKit: Solving the "Last Mile" Problem for AI Agents

Artur Skowroński
Head of Java/Kotlin Space

PUBLISHED: 15 SEP 12, 2025 | 18 MINUTES READ

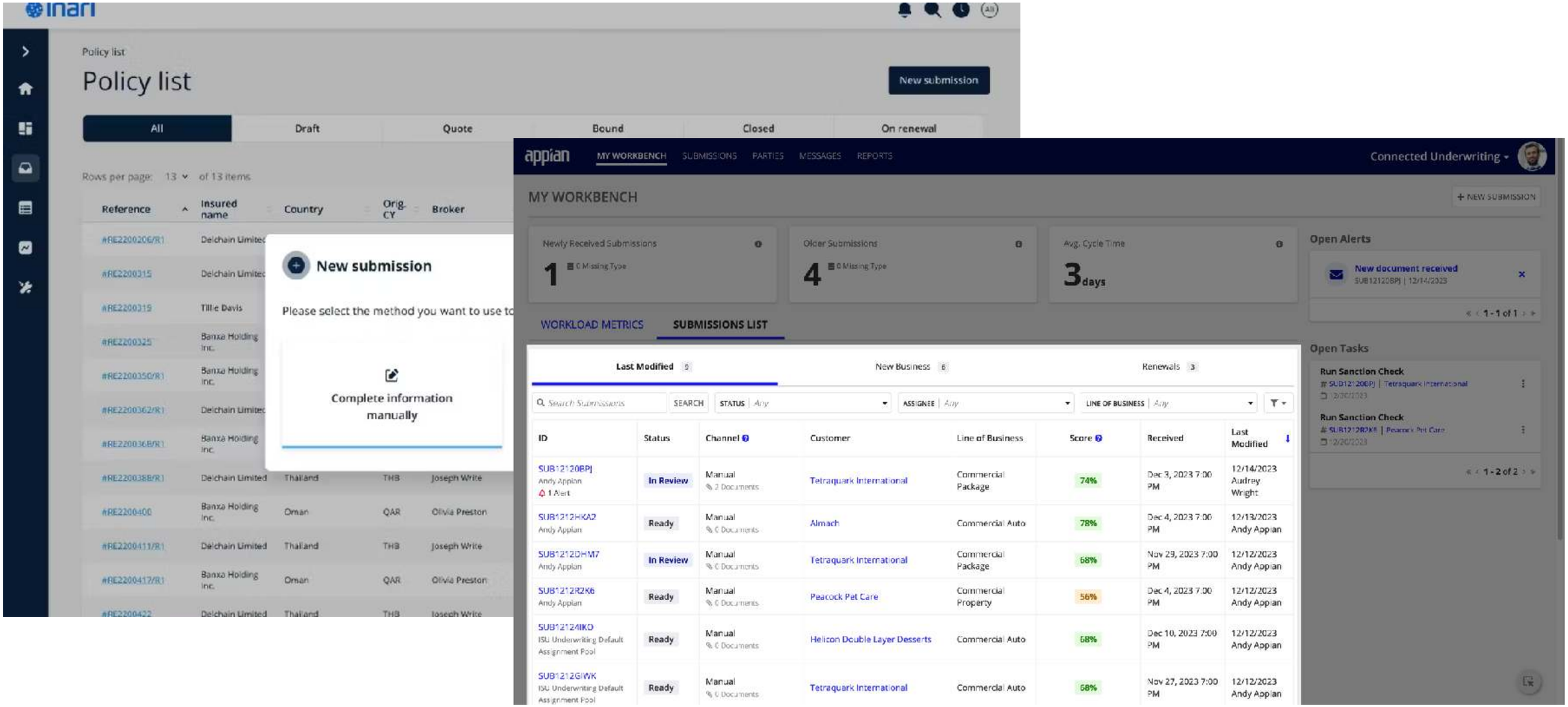
Welcome to another edition of GitHub All Stars, where we look at a key building block of agentic applications - UI - and how Human-in-the-Loop can be implemented in practice. Let's be honest: despite the marketing, no agent solution is perfect, and there will always be moments when we, the protein-based organisms, are needed. That's why any application that aims to solve the problem realistically has to face this challenge head-on.

Filling the Gap Between the Agent and the User

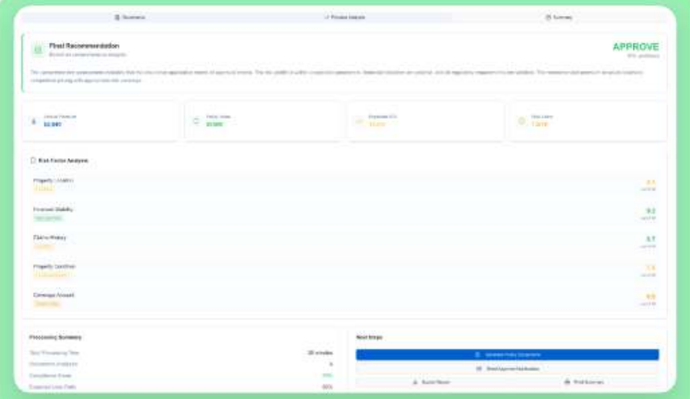
"Why": The AI Agent Era and the "Last Mile" Problem

In recent years we've seen an explosion in AI, driven by powerful agent frameworks like LangGraph and CrewAI (which we're testing in parallel with LangGraph and preparing a write-up on). These tools have changed how we think about automation, enabling advanced AI "brains" that can orchestrate complex back-office tasks. However, despite their ability to reason and act, these frameworks often leave developers with a fundamental challenge: how do you integrate these powerful, autonomous systems with the everyday applications your users actually touch - in an elegant, interactive way? We can't just send users a prompt in Markdown or a blob of raw JSON.

CopilotKit



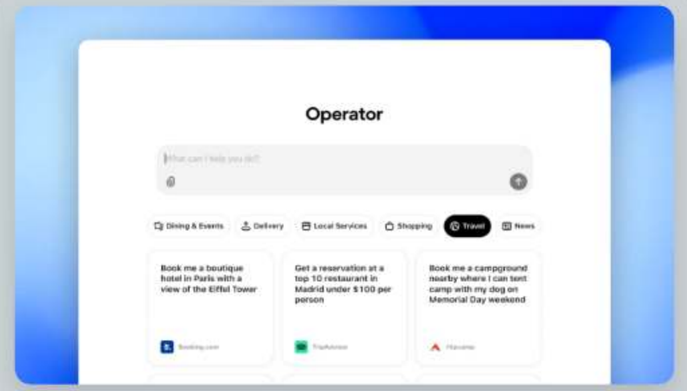
Underwriting Workbench



Our own Agentic Workbench

OPENAPI

API Integration
with existing workbench



Browser Agent Integration
with existing workbench

Introducing the Gemini 2.5 Computer Use model

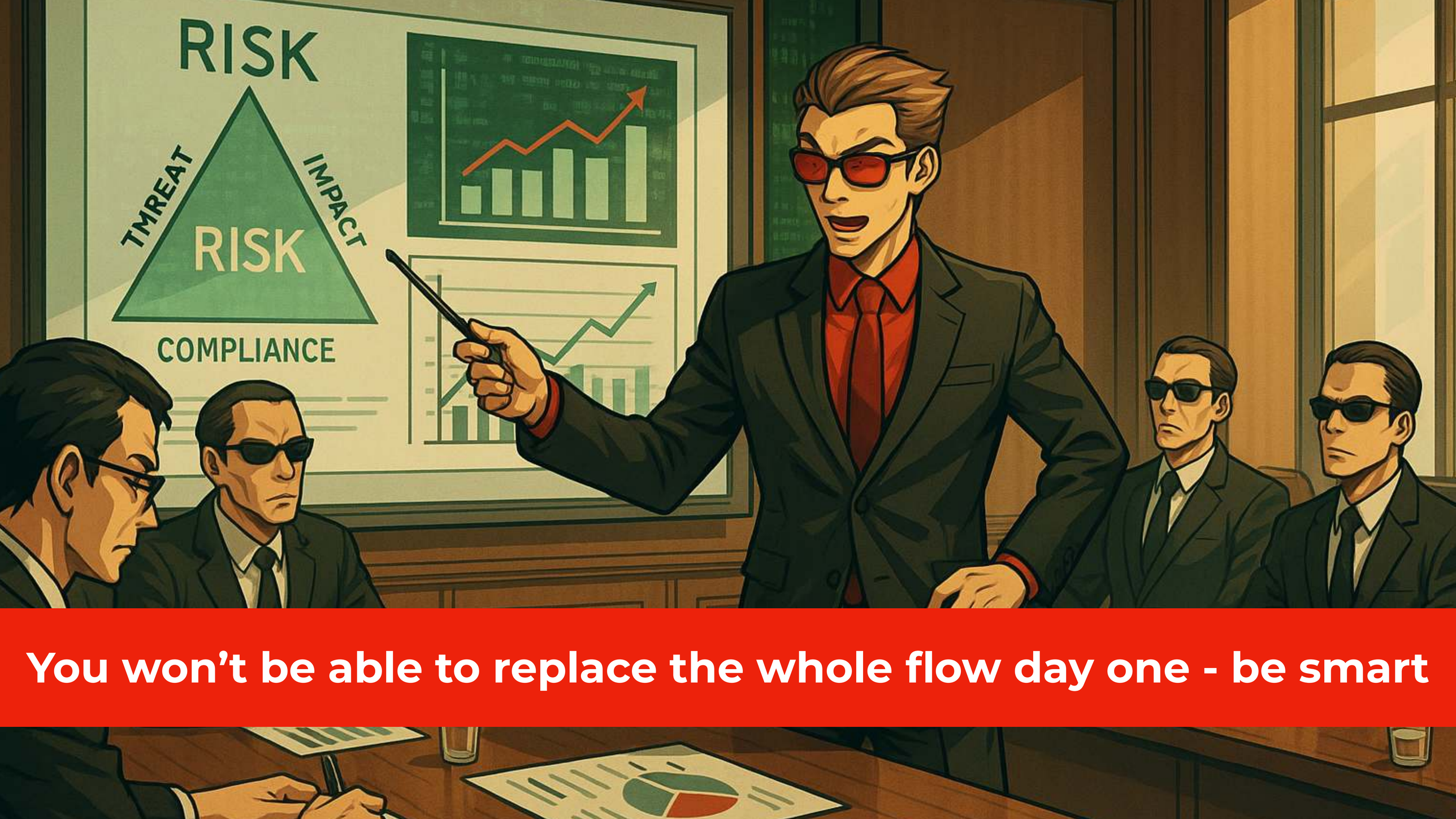
Oct 07, 2025
5 min read

Available in preview via the API, our Computer Use model is a specialized model built on Gemini 2.5 Pro's capabilities to power agents that can interact with user interfaces.

Google DeepMind

Read AI-generated summary

Share



You won’t be able to replace the whole flow day one - be smart

Guardrails, Testing & Observability

Guardrails

Copy page

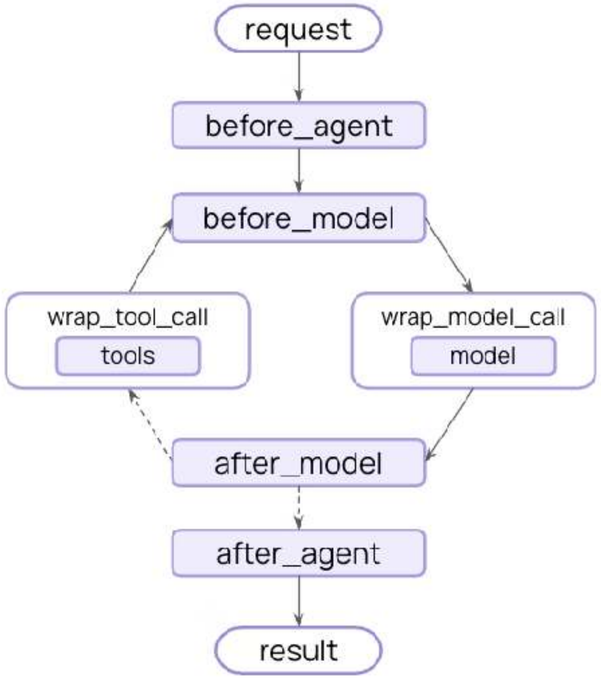
Implement safety checks and content filtering for your agents

Guardrails help you build safe, compliant AI applications by validating and filtering content at key points in your agent's execution. They can detect sensitive information, enforce content policies, validate outputs, and prevent unsafe behaviors before they cause problems.

Common use cases include:

- Preventing PII leakage
- Detecting and blocking prompt injection attacks
- Blocking inappropriate or harmful content
- Enforcing business rules and compliance requirements
- Validating output quality and accuracy

You can implement guardrails using **middleware** to intercept execution at strategic points - before the agent starts, after it completes, or around model and tool calls.



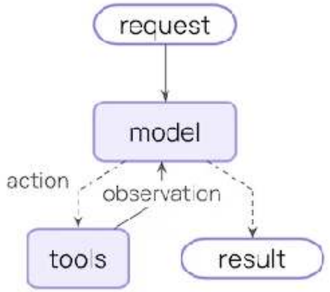
Middleware

Copy page

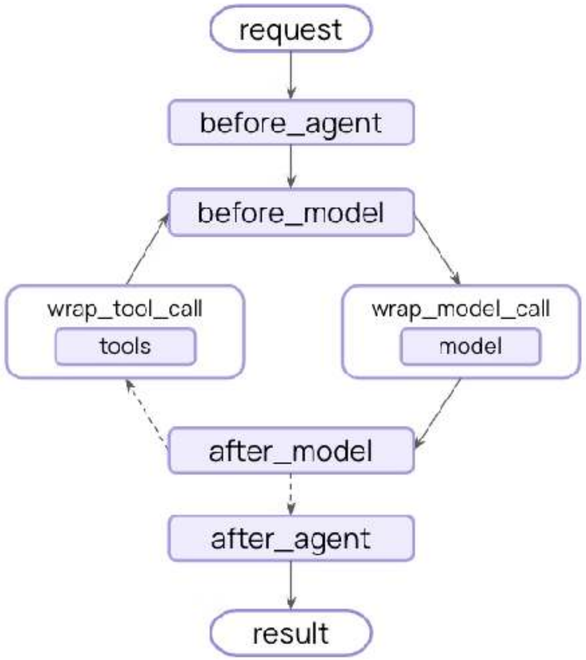
Control and customize agent execution at every step

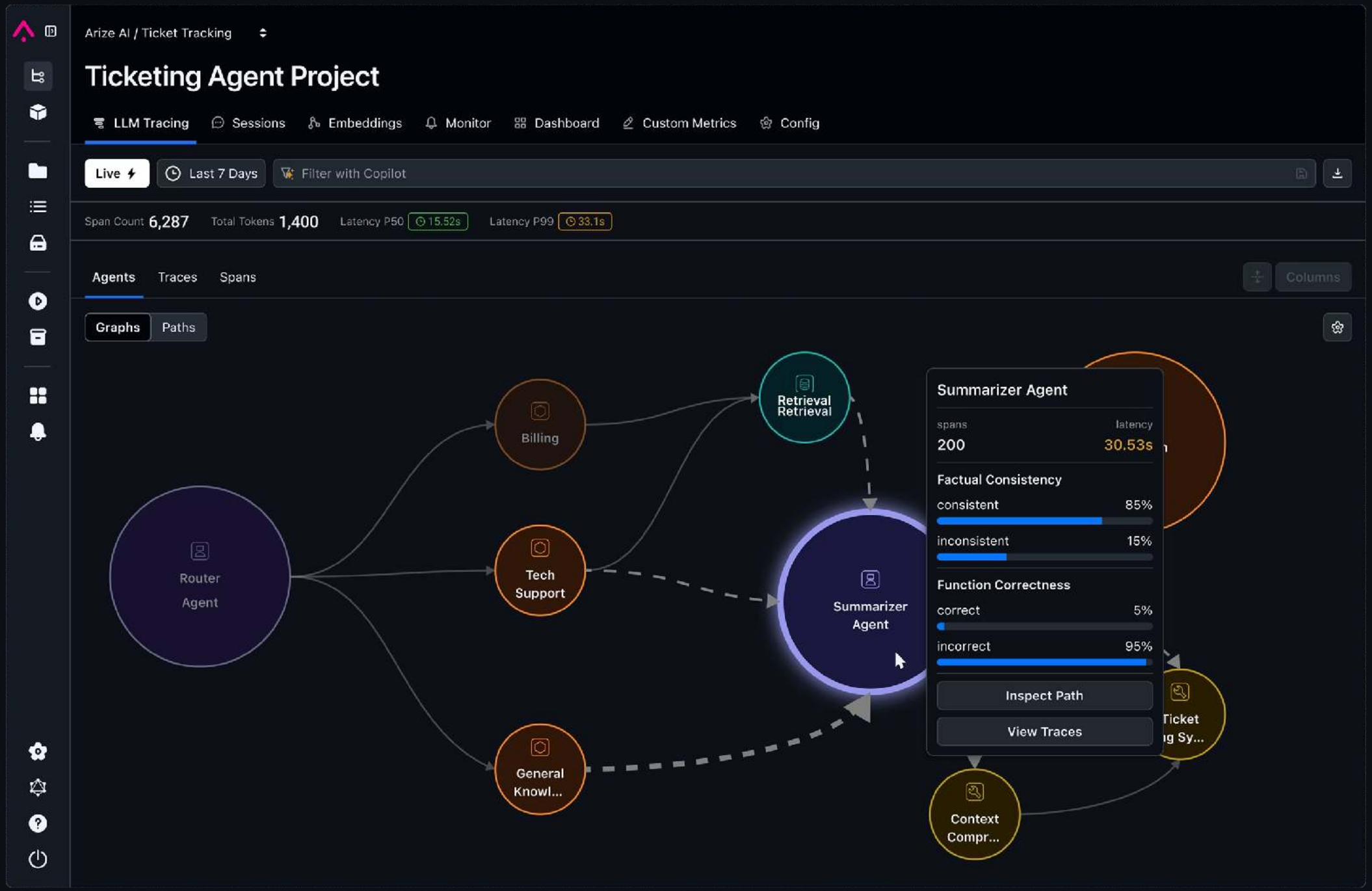
Middleware provides a way to more tightly control what happens inside the agent.

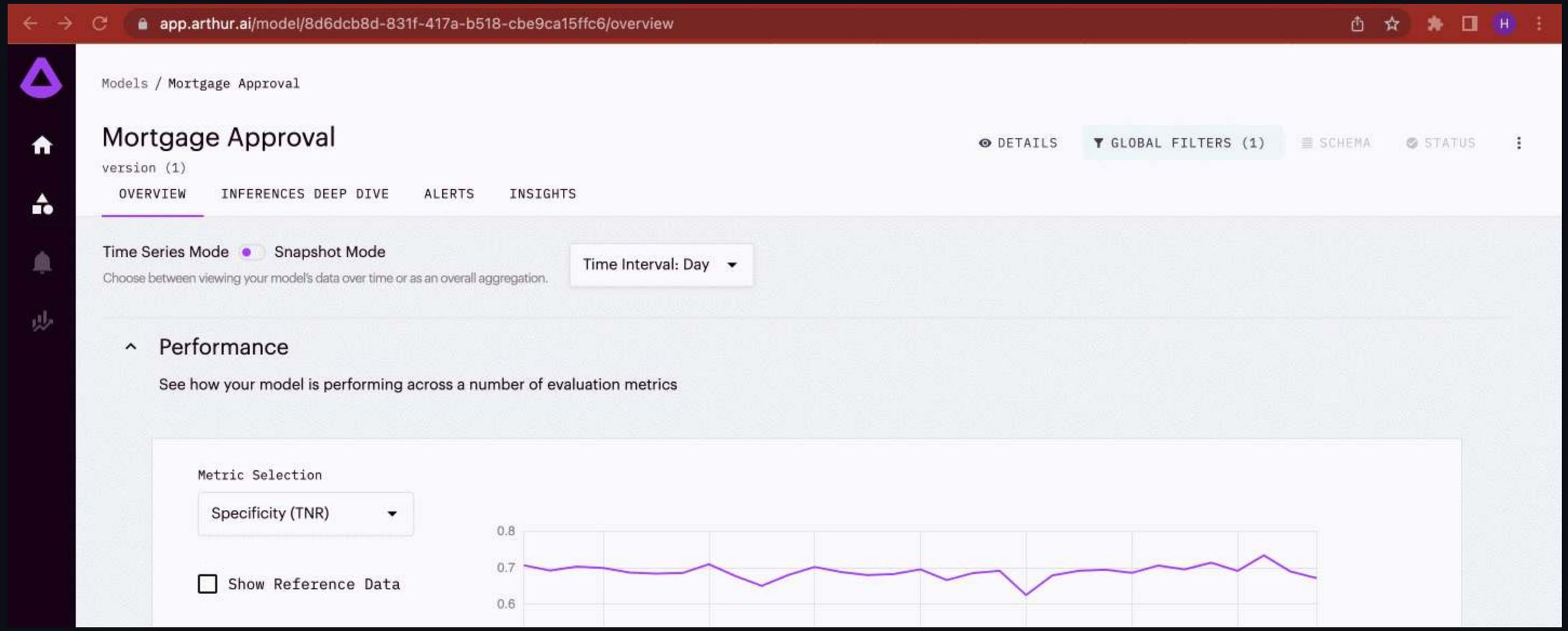
The core agent loop involves calling a model, letting it choose tools to execute, and then finishing when it calls no more tools:



Middleware exposes hooks before and after each of those steps:




Arize AI / Ticket Tracking
Ticketing Agent Project
LLM Tracing
Sessions
Embeddings
Monitor
Dashboard
Custom Metrics
Config
Live
Last 7 Days
Filter with Copilot
Span Count 6,287
Total Tokens 1,400
Latency P50 15.52s
Latency P99 33.1s
Agents
Traces
Spans
Columns
Graphs
Paths
Router Agent
Billing
Tech Support
General Knowl...
Retrieval Retrieval
Summarizer Agent
Context Compr...
Ticket
Summarizer Agent
spans
200
latency
30.53s
Factual Consistency
consistent 85%
inconsistent 15%
Function Correctness
correct 5%
incorrect 95%
Inspect Path
View Traces


app.arthur.ai/model/8d6dcb8d-831f-417a-b518-cbe9ca15ffc6/overview
Models / Mortgage Approval
Mortgage Approval
version (1)
DETAILS
GLOBAL FILTERS (1)
SCHEMA
STATUS
OVERVIEW
INFERENCES DEEP DIVE
ALERTS
INSIGHTS
Time Series Mode
Snapshot Mode
Choose between viewing your model's data over time or as an overall aggregation.
Time Interval: Day
Performance
See how your model is performing across a number of evaluation metrics
Metric Selection
Specificity (TNR)
Show Reference Data
0.8
0.7
0.6

Arthur

Budget

We built our AI from the ground-up
Be honest
I am being honest
…the OpenAI bill?
We spent $7M on GPT4 to do $600K in revenue
Thank you

Reasoning Loop is &%*%^% expensive

· 12 minutes on avg of processing (without Human-in-the-loop)

· 25-50$ for processing the long offer

Agentic system is... cheap

- You get **AT LEAST** initial draft (without Human-in-the-loop) - huge time saver

- 25$ for the most dauniting part of the job

Productivity gain?

- Higher throughput - 10%
- Shorter turnaround time - up to 20%

Average Salary of Junior Underwriter

- 65k year

Perfectly balanced,
as all things should be.

The worst it will ever be - 2016

The worst it will ever be - 2017

The worst it will ever be - 2018

The worst it will ever be - 2019

The worst it will ever be - 2020

The worst it will ever be - 2021

The worst it will ever be - 2022

The worst it will ever be - 2023

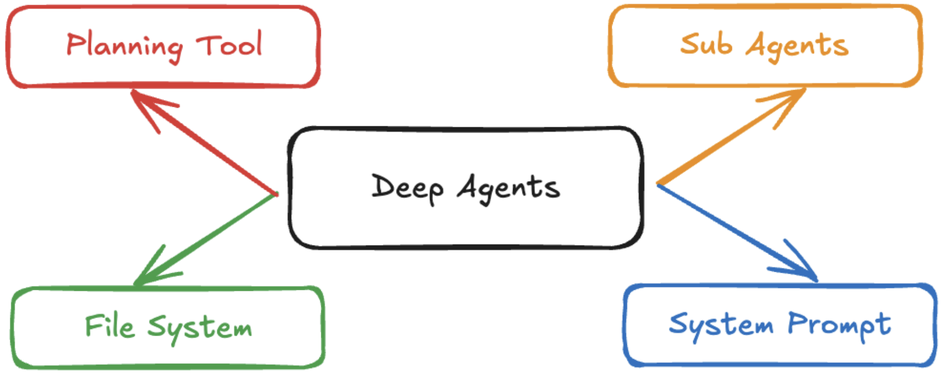
The worst it will ever be - 2024

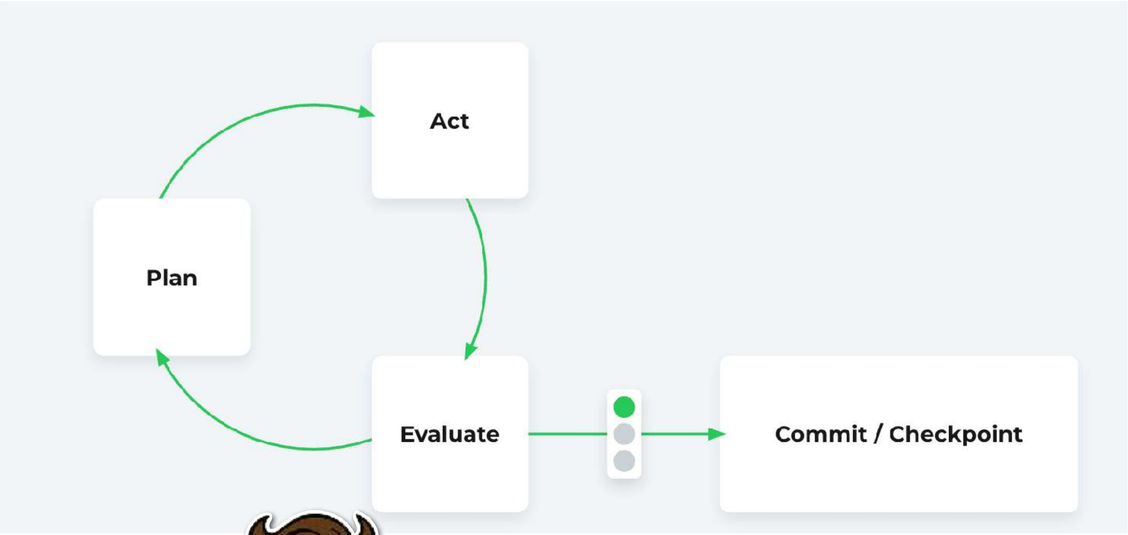
Again: Human-in-the-loop precision protects your budget

- Cache tool responses (Etag/ttl)
- Checkpoint every step!

Self-improving Agents

Let’s do fine tuning





You make context better or evaluation better

Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models

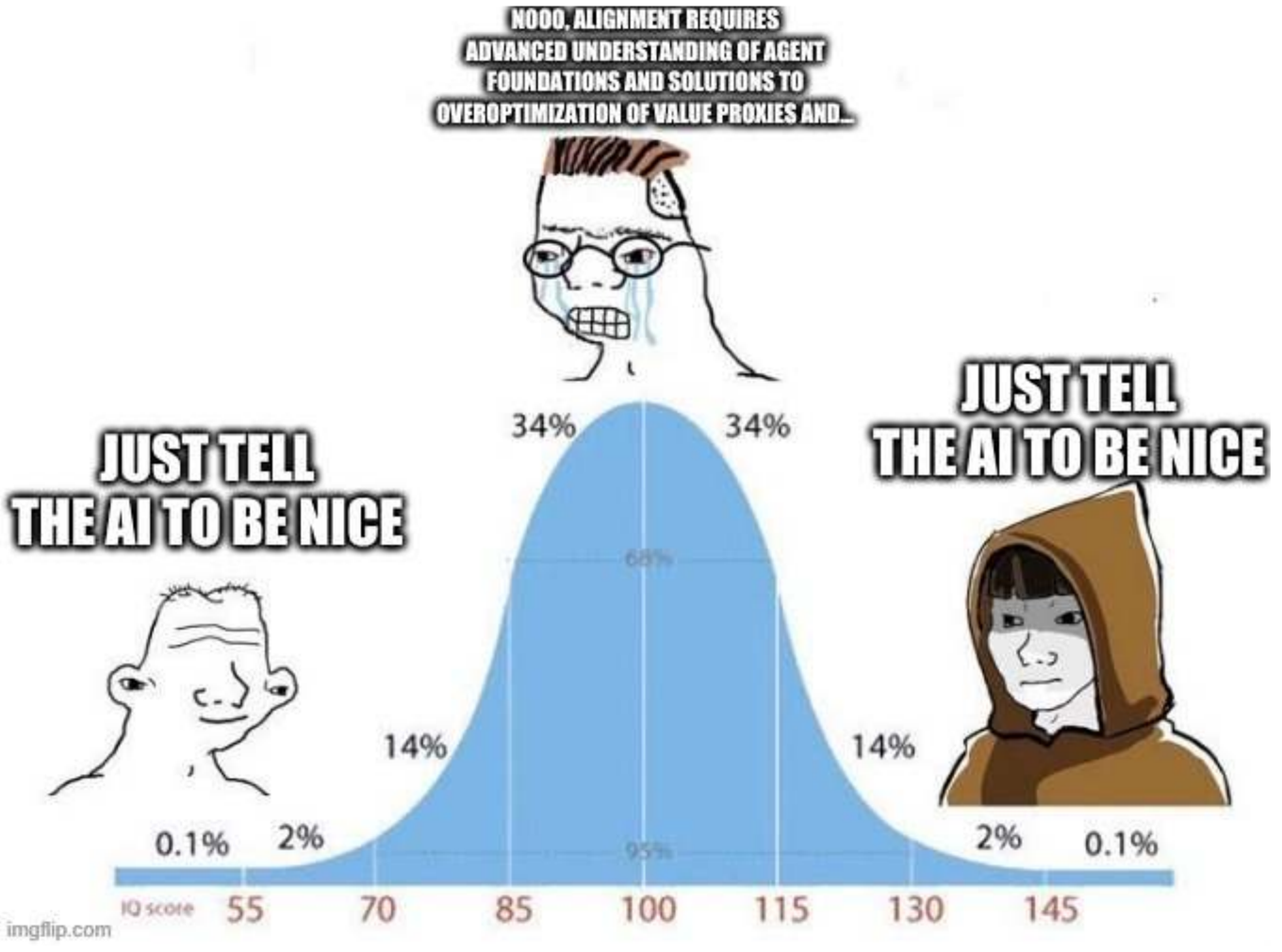
Qizheng Zhang [1*] **Changran Hu** [2*] **Shubhangi Upasani** [2] **Boyuan Ma** [2] **Fenglu Hong** [2]
Vamsidhar Kamanuru [2] **Jay Rainton** [2] **Chen Wu** [2] **Mengmeng Ji** [2] **Hanchen Li** [3]
Urmish Thakker [2] **James Zou** [1] **Kunle Olukotun** [1]

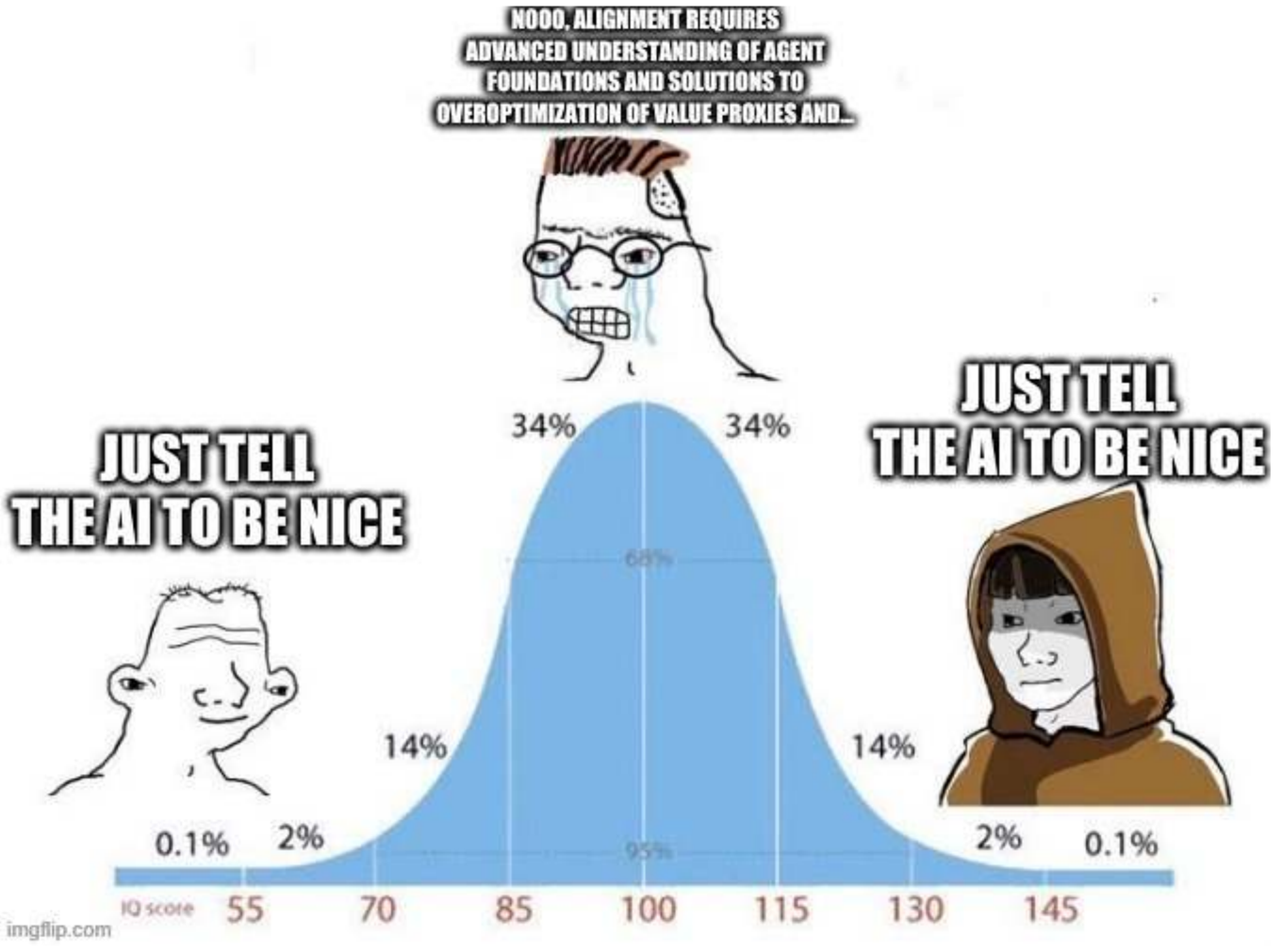
[1] Stanford University [2] SambaNova Systems, Inc. [3] UC Berkeley * equal contribution

qizhengz@stanford.edu, changran.hu@sambanovasystems.com

Abstract

Large language model (LLM) applications such as agents and domain-specific reasoning increasingly rely on *context adaptation*—modifying inputs with instructions, strategies, or evidence, rather than weight updates. Prior approaches improve usability but often suffer from brevity bias, which drops domain insights for concise summaries, and from context collapse, where iterative rewriting erodes details over time. Building on the adaptive memory introduced by Dynamic Cheatsheet, we introduce ACE (**A**gentic **C**ontext **E**ngineering), a framework that treats contexts as evolving playbooks that accumulate, refine, and organize strategies through a modular process of generation, reflection, and curation. ACE prevents collapse with structured, incremental updates that preserve detailed knowledge and scale with long-context models. Across agent and domain-specific benchmarks, ACE optimizes contexts both offline (*e.g.*, system prompts) and online (*e.g.*, agent memory), consistently outperforming strong baselines: +10.6% on agents and +8.6% on finance, while significantly reducing adaptation latency and rollout cost. Notably, ACE could adapt effectively without labeled supervision and instead by leveraging natural execution feedback. On the AppWorld leaderboard, ACE matches the top-ranked production-level agent on the overall average and surpasses it on the harder test-challenge split, despite using a smaller open-source model. These results show that comprehensive, evolving contexts enable scalable, efficient, and self-improving LLM systems with low overhead.

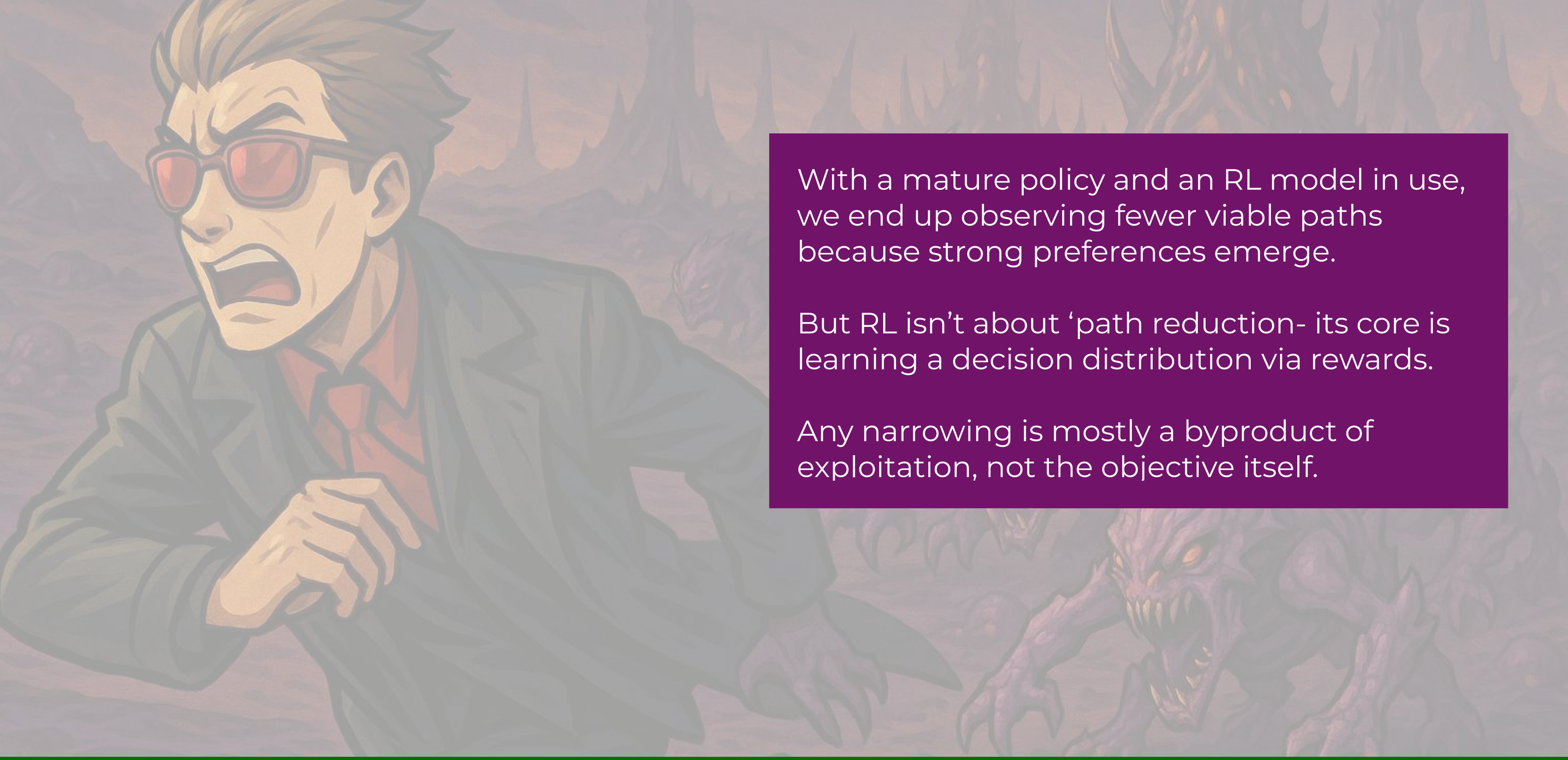

NOOO, ALIGNMENT REQUIRES ADVANCED UNDERSTANDING OF AGENT FOUNDATIONS AND SOLUTIONS TO OVEROPTIMIZATION OF VALUE PROXIES AND...
JUST TELL THE AI TO BE NICE
JUST TELL THE AI TO BE NICE
34%
34%
14%
14%
0.1%
2%
2%
0.1%
IQ score
55
70
85
100
115
130
145
imgflip.com


NOOO, ALIGNMENT REQUIRES ADVANCED UNDERSTANDING OF AGENT FOUNDATIONS AND SOLUTIONS TO OVEROPTIMIZATION OF VALUE PROXIES AND...
JUST TELL THE AI TO BE NICE
JUST TELL THE AI TO BE NICE
34%
34%
14%
14%
0.1%
2%
2%
0.1%
IQ score
55
70
85
100
115
130
145
imgflip.com

Hard to swallow pills
54
543
THE DATA IS MORE IMPORTANT THAN THE MODEL
imgflip.com

Reinforced Learning

Reinforced Learning



With a mature policy and an RL model in use, we end up observing fewer viable paths because strong preferences emerge.

But RL isn’t about ‘path reduction- its core is learning a decision distribution via rewards.

Any narrowing is mostly a byproduct of exploitation, not the objective itself.

Reinforced Learning

I] 15 Sep 2025

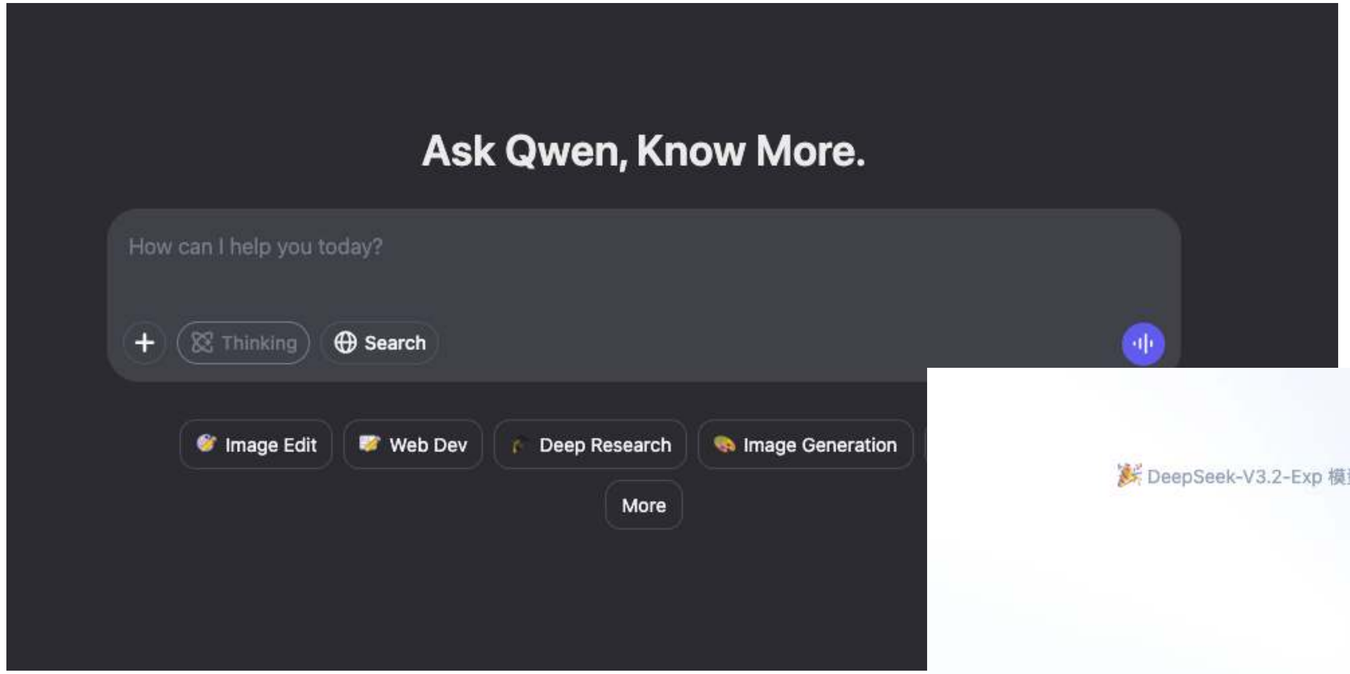
Small Language Models are the Future of Agentic AI

Peter Belcak[1] **Greg Heinrich**[1] **Shizhe Diao**[1] **Yonggan Fu**[1] **Xin Dong**[1]
Saurav Muralidharan[1] **Yingyan Celine Lin**[1,2] **Pavlo Molchanov**[1]
[1]NVIDIA Research [2]Georgia Institute of Technology
agents-research@nvidia.com

Abstract

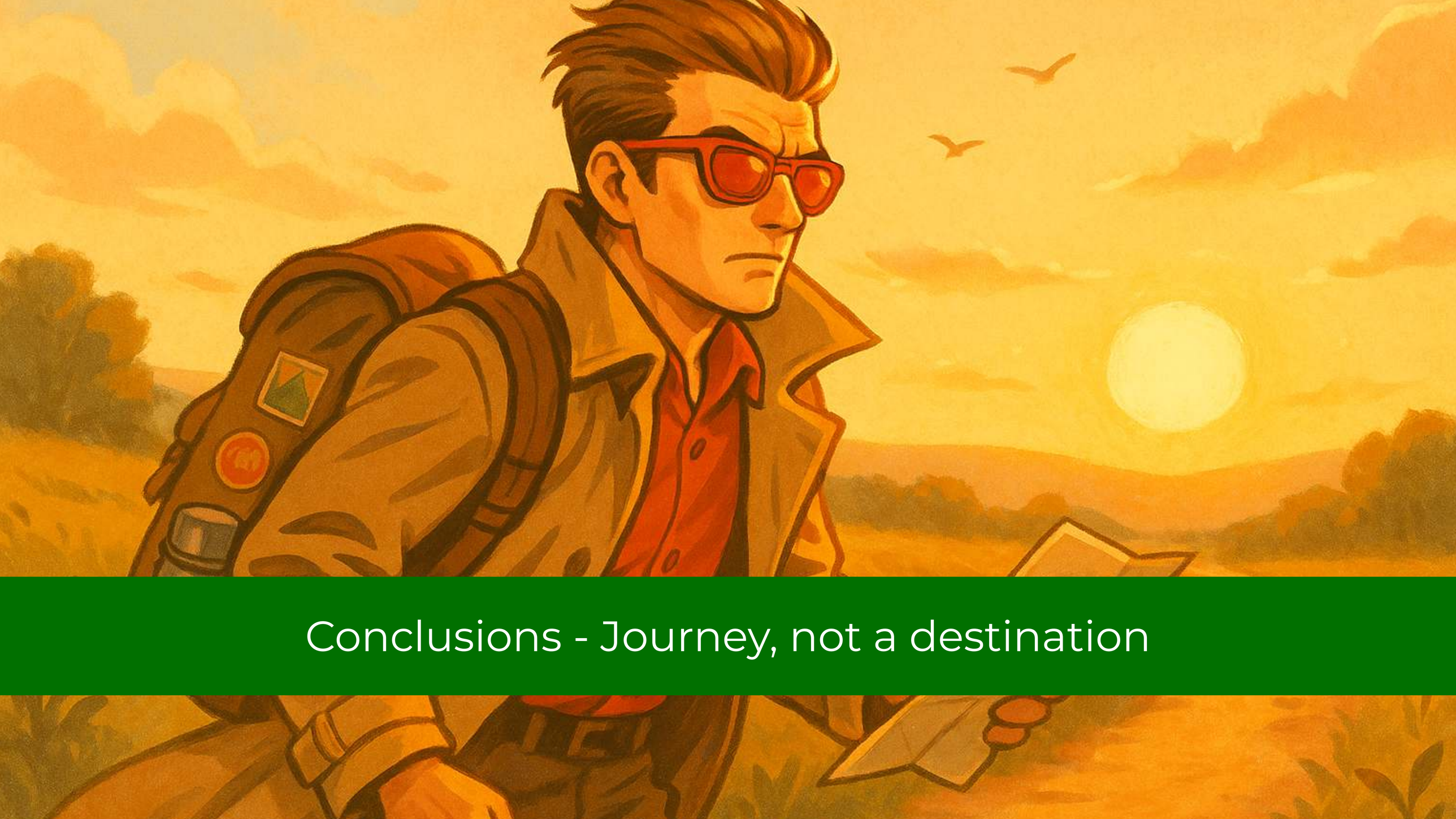
Large language models (LLMs) are often praised for exhibiting near-human performance on a wide range of tasks and valued for their ability to hold a general conversation. The rise of agentic AI systems is, however, ushering in a mass of applications in which language models perform a small number of specialized tasks repetitively and with little variation.

Here we lay out the position that small language models (SLMs) are *sufficiently powerful, inherently more suitable, and necessarily more economical for many invocations in agentic systems, and are therefore the future of agentic AI*. Our


Ask Qwen, Know More.
How can I help you today?
Thinking
Search
Image Edit
Web Dev
Deep Research
Image Generation
More

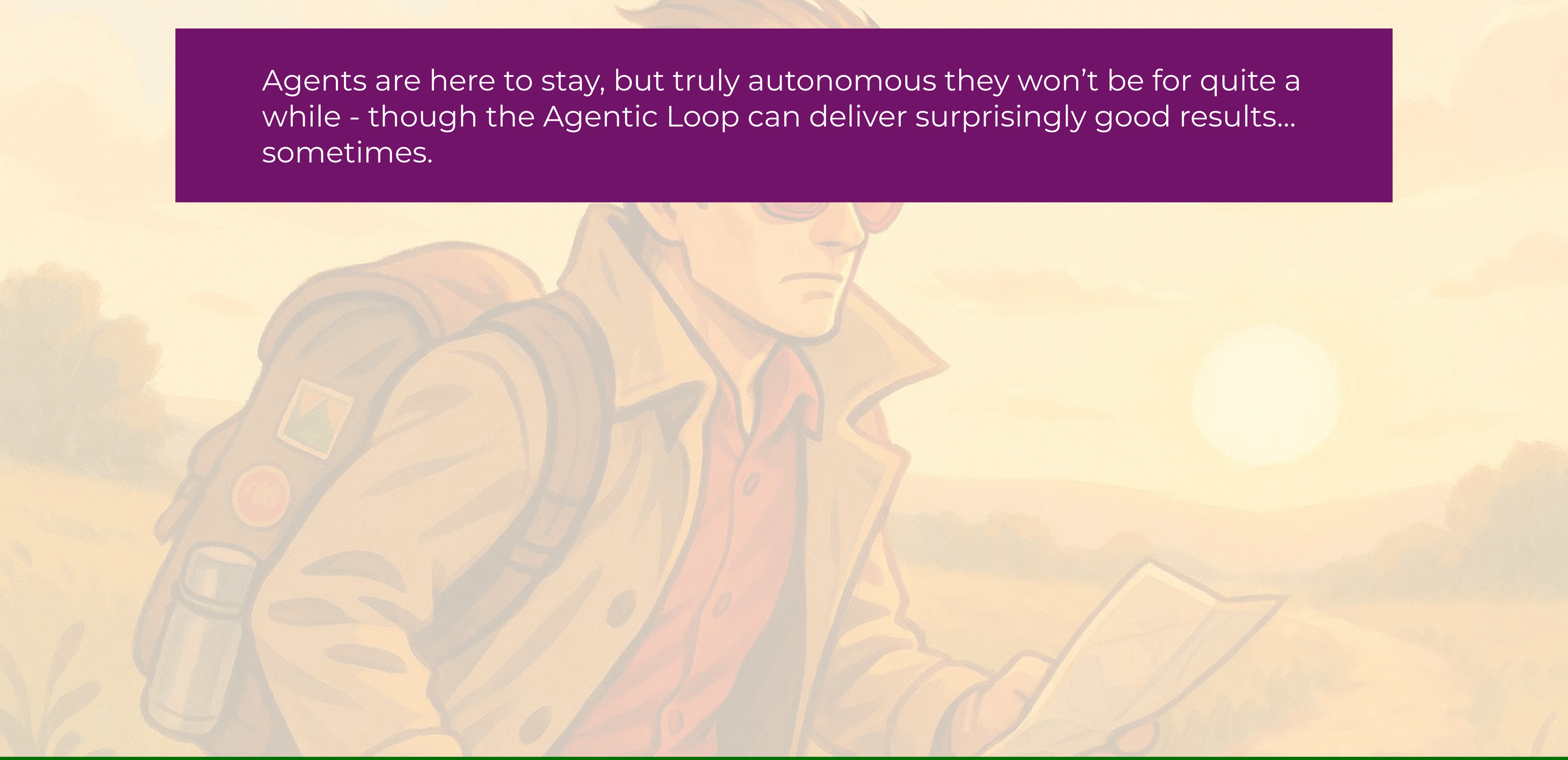
DeepSeek-V3.2-Exp 模型发布，训练推理提效，API 同步降价，在网页端、APP 和 API 全面上线，点击查看详情。
deepseek
探索未至之境
获取手机 App
DeepSeek 官方推出的免费 AI 助手
搜索写作阅读解题翻译工具

gpt-oss-20b

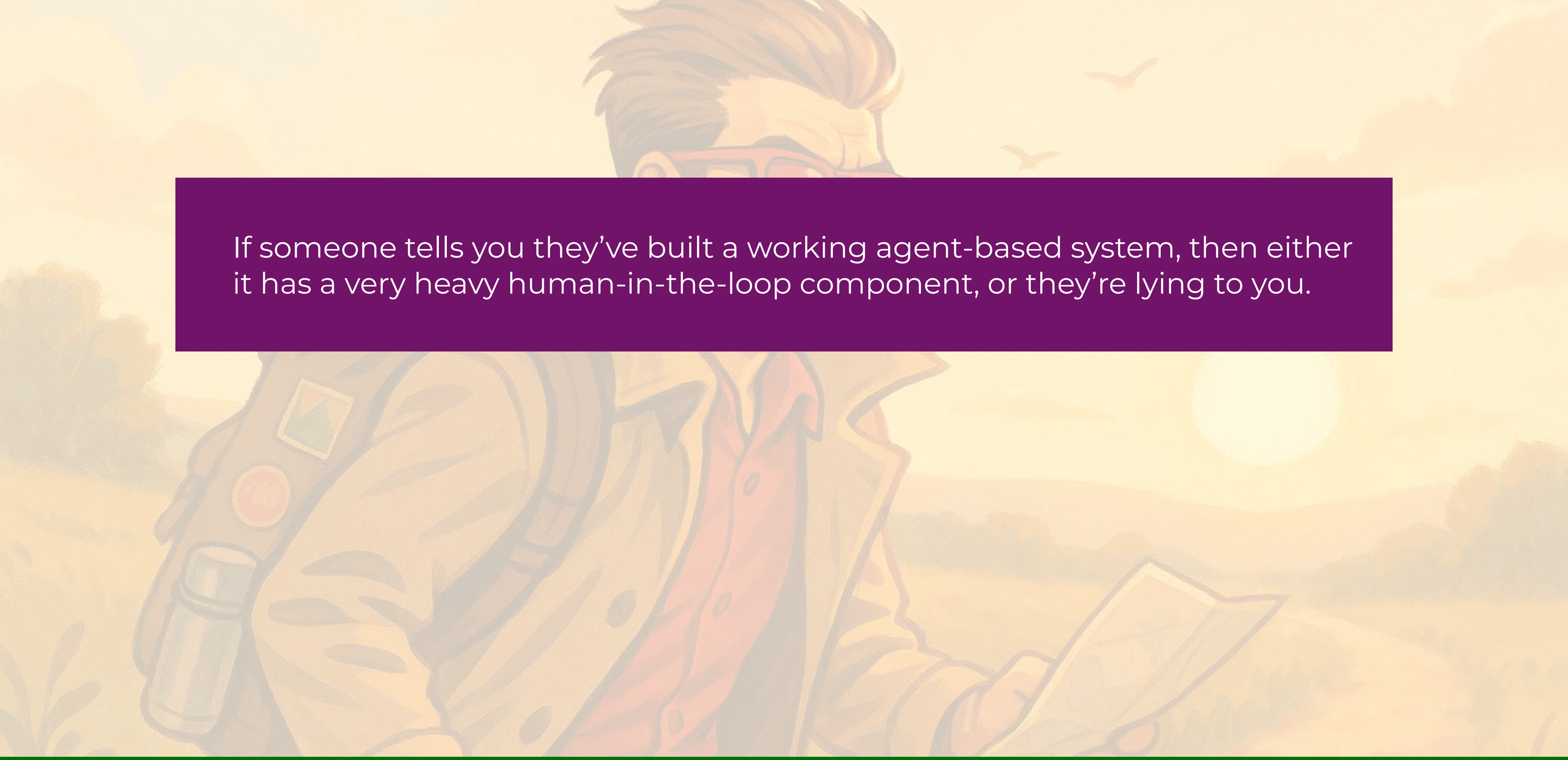


Conclusions - Journey, not a destination

Agents are here to stay, but truly autonomous they won't be for quite a while - though the Agentic Loop can deliver surprisingly good results... sometimes.



Forward-Deployed AI

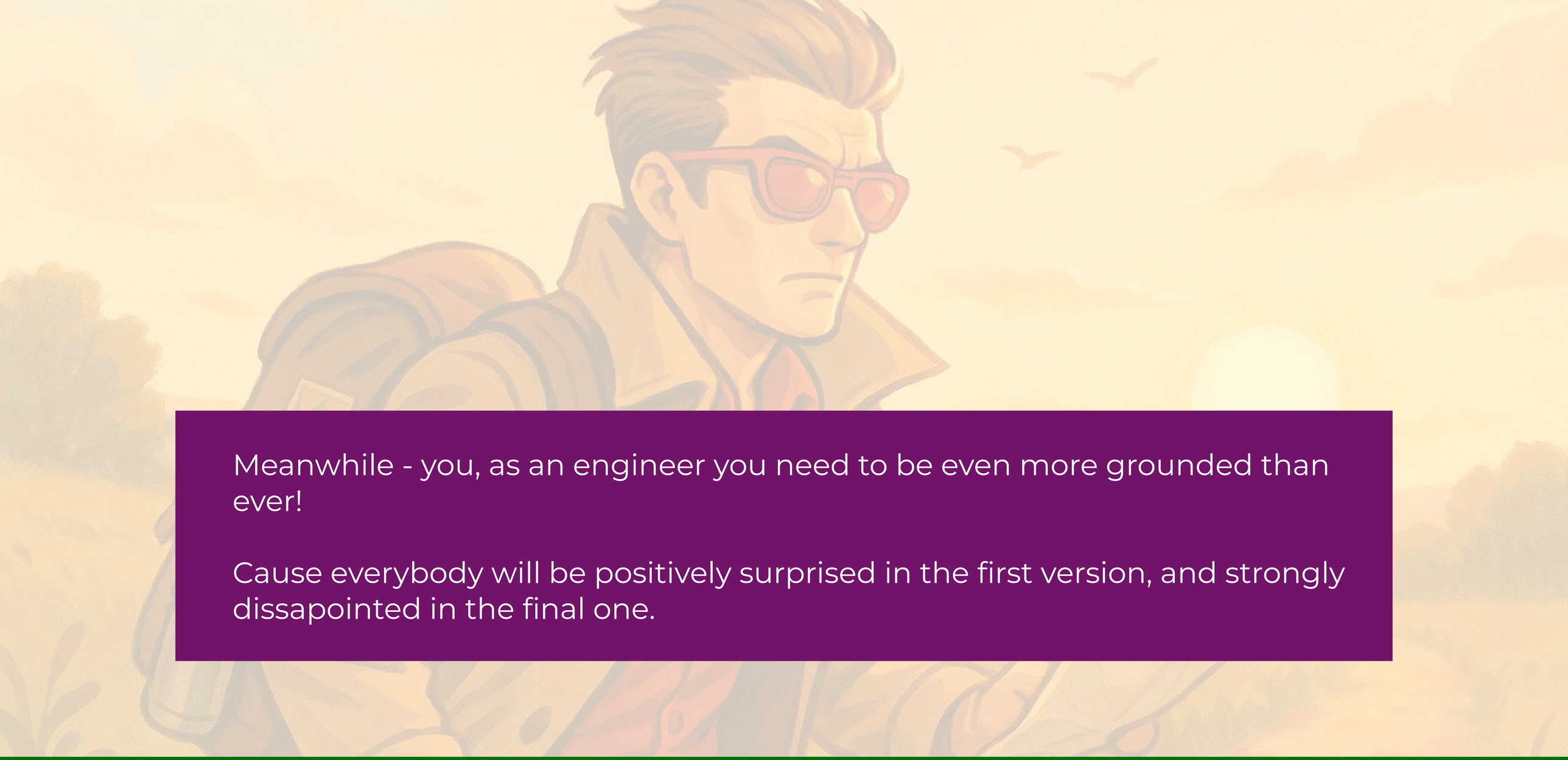


If someone tells you they've built a working agent-based system, then either it has a very heavy human-in-the-loop component, or they're lying to you.

At the same time, there are already patterns today where agents are useful - and we really should start treating LLMs as a standard architectural mechanism.

We are not operating in the world of polished experience like Apple one. You need different mental. Sometimes your next version will be slop.

Forward-Deployed AI



Meanwhile - you, as an engineer you need to be even more grounded than ever!

Cause everybody will be positively surprised in the first version, and strongly dissapointed in the final one.

Thank you 🙏

@ArturSkowronski

VIRTUSLAB

jvm weekly