

## Learn How Monday.com Built Their AI Agent Workforce with LangGraph | LangChain Interrupt



LangChain  
117K subscribers

Subscribe



32



Download



1,067 views Jun 9, 2025 #langchain #AIAgents #LangGraph

Assaf Elovic, Head of AI at Monday.com, reveals how they achieved 100% month-over-month growth in AI usage while processing 1 billion work tasks annually. Learn why trust, not technology, is the biggest barrier to AI adoption and discover the four critical components that dramatically increased user adoption: autonomy control, seamless entry points, preview functionality, and explainability. Built on LangGraph and LangSmith, Monday's digital workforce demonstrates how to balance multi-agent systems while avoiding compound hallucination and implementing effective human-in-the-loop workflows. Assaf also shares their vision for dynamic agent orchestration—where a finite set of specialized agents can handle infinite tasks through intelligent workflow automation.

Watch all of our recorded sessions from Interrupt here: <https://interrupt.langchain.com/video...>

# BREAKTHROUGH AGENTS: BUILDING OUR DIGITAL WORKFORCE WITH LANGGRAPH

by **monday.com**



- ✦ Head of AI at monday.com
- ✦ Scout at Sequoia Capital
- ✦ Live in Tel Aviv, Israel
- ✦ Building AI products for the past 10+ years
- ✦ Co-Founded GPT Researcher and Tavily
- ✦ Previously Head of R&D at Wix.com

## Introducing monday.com

300K+ businesses  
worldwide

Over \$1B in ARR

7M AI actions  
per month  
(and growing)

Growing 33% y/y

Processes over 1B  
tasks per year

Over 60K active  
accounts using AI  
and over 100K  
monthly active users

# Introducing monday.com

1B

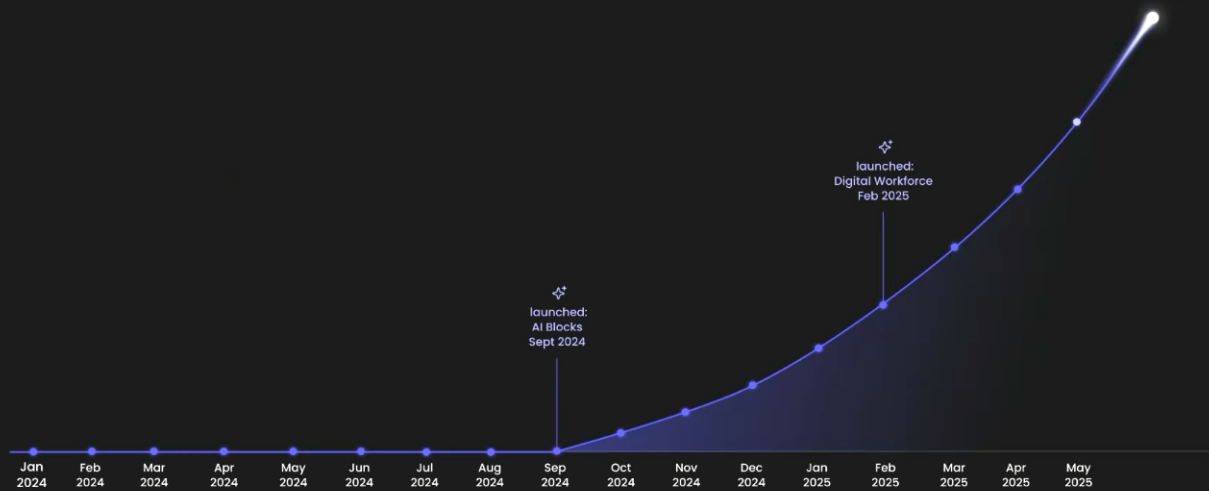
tasks are created on  
monday per year

Done

Working on it

Stuck

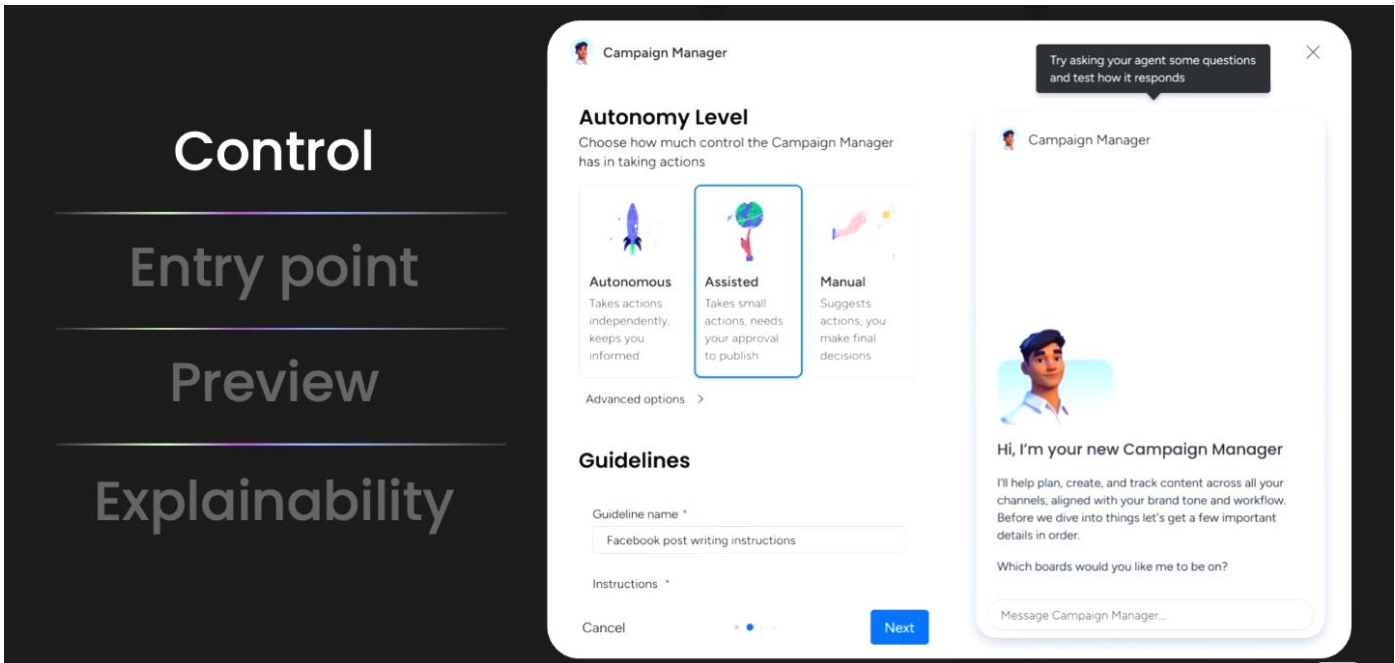
## Gaining momentum with AI



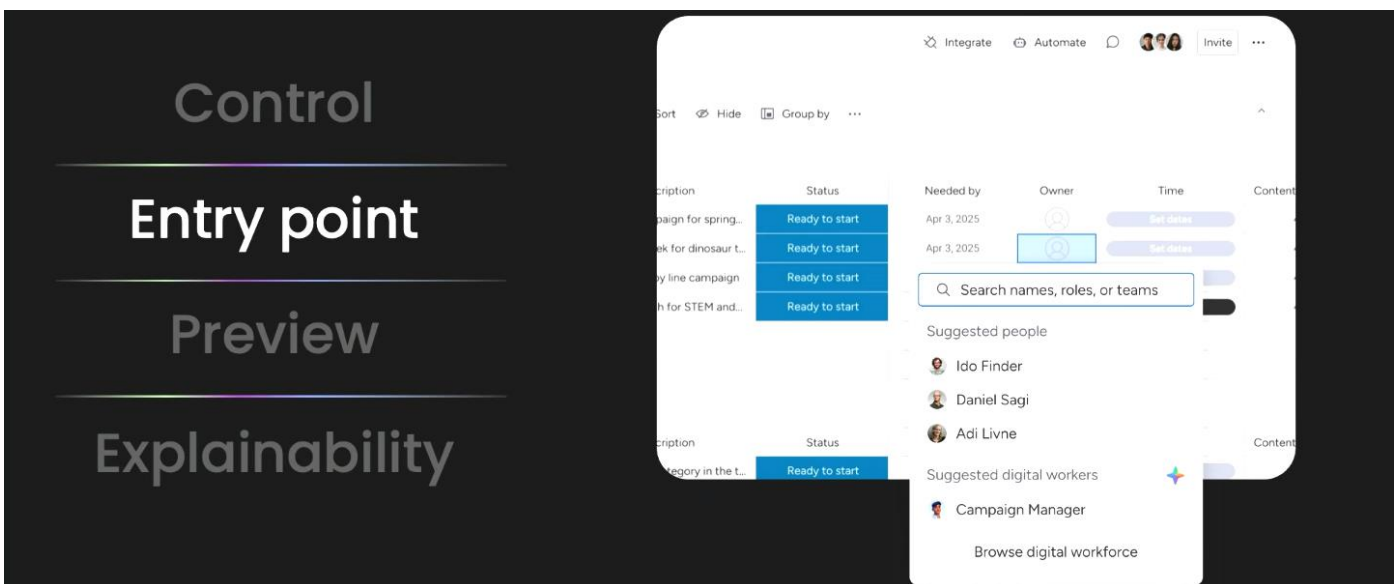
## Your own AI team of specialists

Digital workers that handle real work for you — constantly  
learning, adapting, and anticipating your needs.





Every customer has a different control/trust level, give them options to gain trust.



Think about how the user can assign task to agents easily

Control

---

Entry point

---

Preview

---

Explainability

All the user to be able to preview the work being done by the agents, like a **dashboard** that displays tasks and the test results with **links** for the user to check on the artifacts. Allow the user to make **reassignments** or **stoppage** if needed.

Control

---

Entry point

---

Preview

---

Explainability

Think about how the user can understand what the agent actually did or plans to do.

## Why LangGraph?

Flexible graph-based flow

Not opinionated with maximum customization

Native LangChain integration

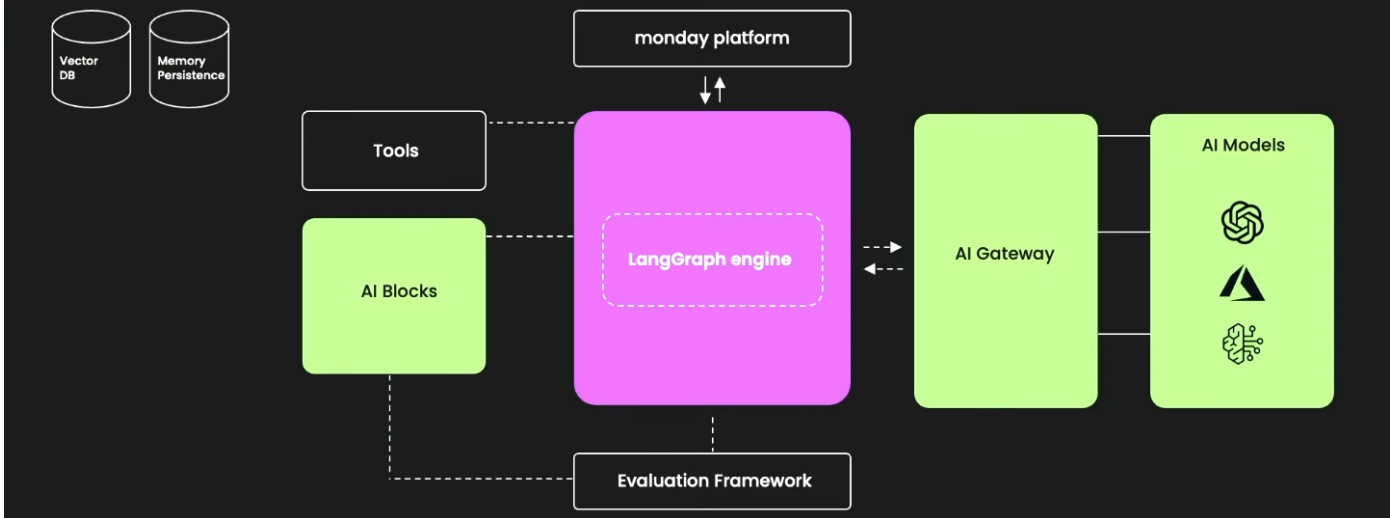
Proven production reliability

Exceptional observability

Active community and support

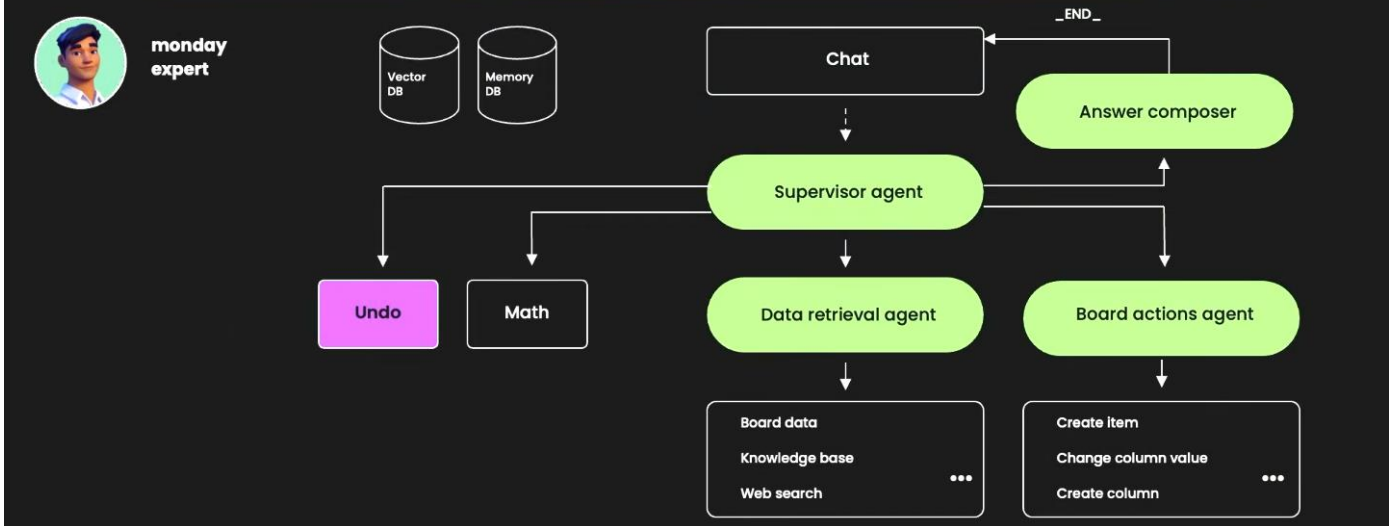
LangGraph and LangSmith (for monitoring) agent ecosystem framework is non-opinionated but still provides features like interrupts, checkpoints, human-in-the-loop, persistence memory, and customizations with native integrations.

# Architecture Overview



AI Blocks are the internal AI actions that we have developed at Monday, we also built our own evaluation framework for our agents. AI Gateway is our way of preserving inputs and outputs in our engine.

# Digital Worker Example



This is our conversational agent-based system based on the supervisor methodology, it holds 4 main agents that do the work. We also give the supervisor the ability to **undo** any actions within the system.

## Key Takeaways

Design for the  
**99%** of requests  
you won't know

Fallback first,  
features second



Build your eval  
pipeline **before**  
your model

Evaluation is your real IP



Deploy **humans**  
at decision  
points

Human-in-the-Loop is non-  
negotiable



When building conversational agents, always assume there will be user-cases that you will not know how to handle. Use fallbacks like searching the knowledgebase and tell the user how they can do it.

## Key Takeaways

Budget 3x more  
time for the **last**  
**30%** accuracy

Demo  $\neq$  Production



Add safety  
checks **outside**  
your LLM, not  
inside

Safety mechanisms are  
your insurance



Split your **super**  
**agent** into  
specialist teams

One agent can't do  
everything, but there's a  
balance

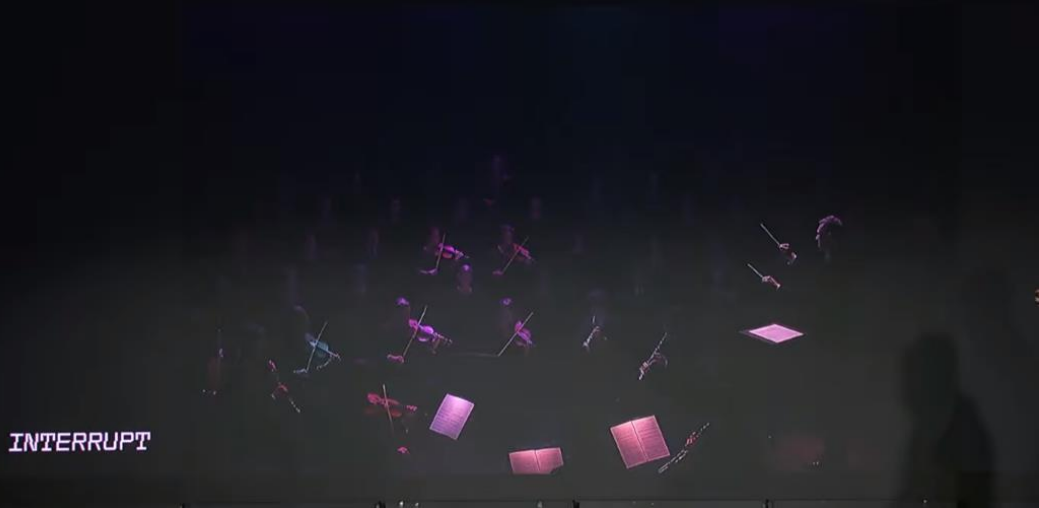


Build all **guardrails** outside of your LLM using things like LLM as a judge, cursor stopping after 25 runs even if running successful. Always try to build **optimal agent count** to prevent compound hallucinations.



## The future of work

INTERRUPT

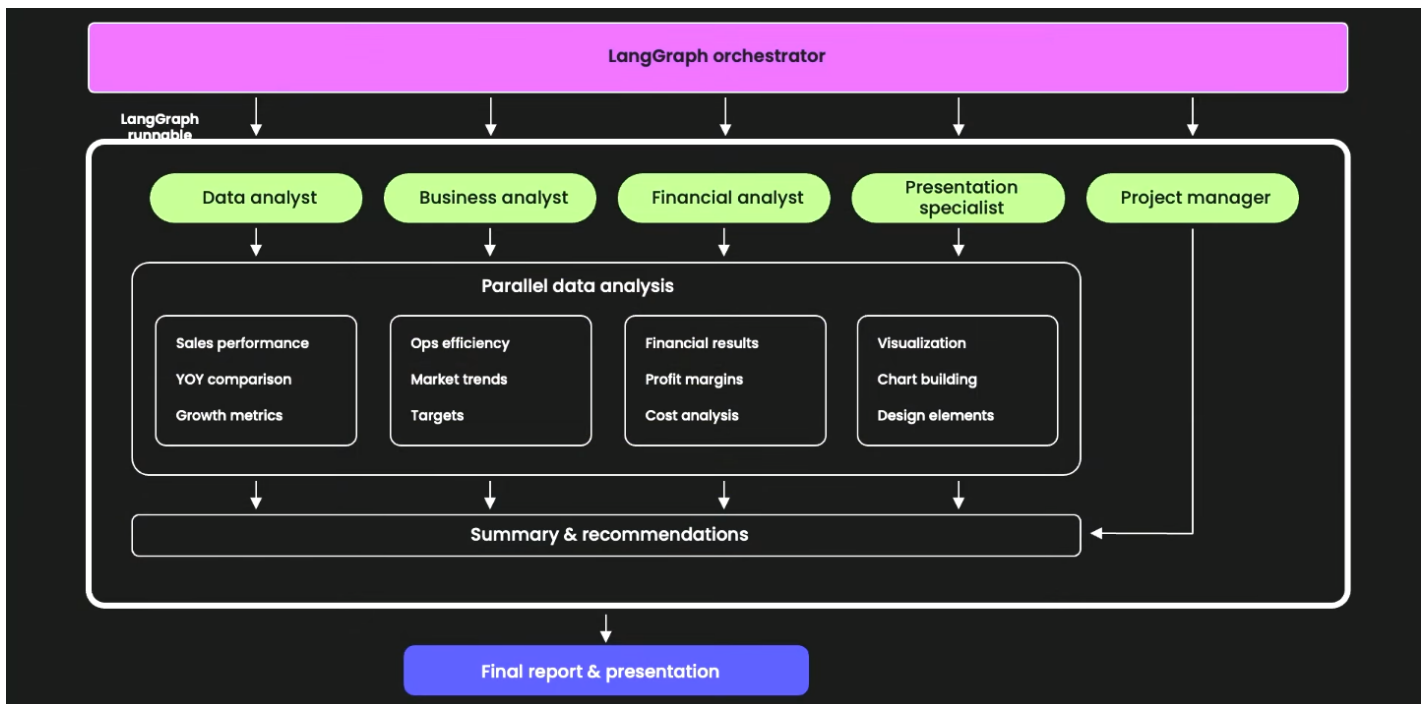


## The future of work: example

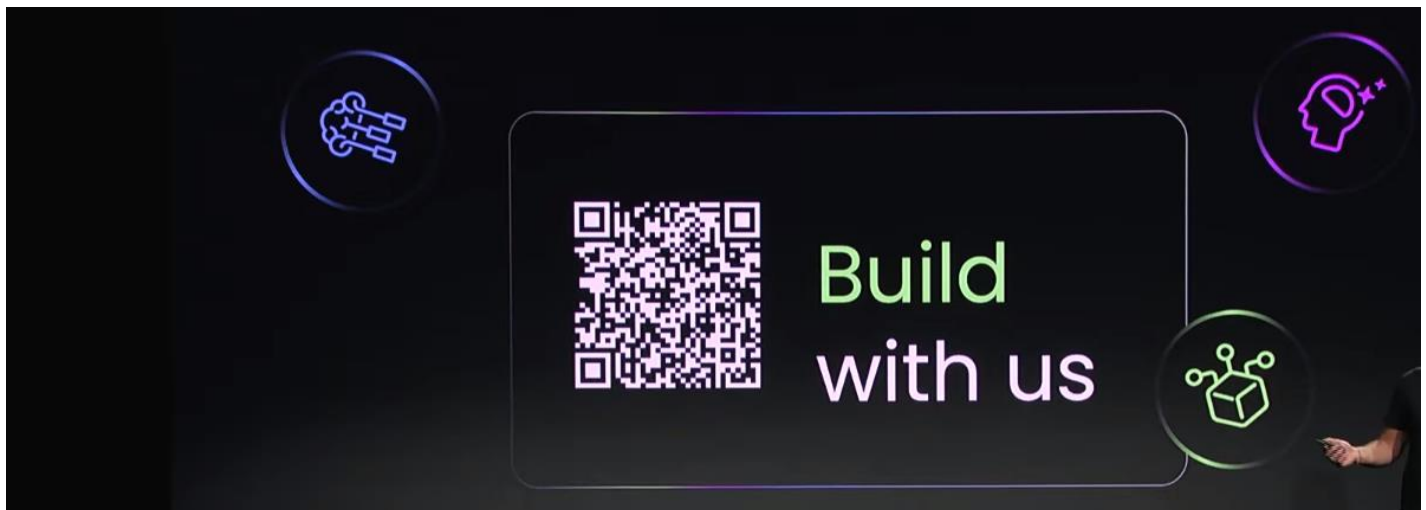
"task": "Q2 Business performance report for board meeting"

## The future of work

A finite set of specialized agents can solve an infinite variety of business problems



Imagine for every work to be done, we have a dynamic way to orchestrate and build out a dynamic workflow with dynamic edges and dynamic rules and choosing a dynamic set of agents for the tasks. Run the task and then dynamically dissolve the team.



We are opening up our agent marketplace to all of you via our waitlist.

