GenAI agents
model · harness · environment

Jonathan Julián Huerta y Munive
jjhymuni@cs.aau.dk
Aalborg University, Copenhagen, Denmark
DTU webinar, October 2026

Who has used one of these?



Claude Code



Codex



Muse Code



Hermes Agent



Grok Bot



DeepSeek Harness

Section 1

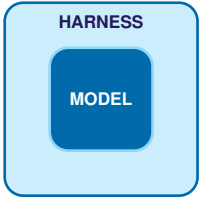
How to think about AI in late 2026

The 2023 and the 2026 descriptions

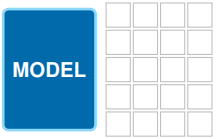
2023: “It is trained on internet text and predicts the next word.”

2026: “It is trained to solve tasks.”

- The 2023 description covers only *pretraining*.
- Current models are also trained with *reinforcement learning* on tasks that a program can check.
- The model runs inside a *harness*: a program that calls it in a loop and runs the tools it requests.

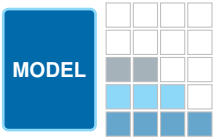


The weights are fixed, the context changes



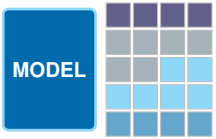
1. A new session

The context is empty.
The weights do not change during use.



2. Your task

Instructions, files,
tool output, earlier messages



3. The context is full

The harness replaces the oldest messages by a summary: *compaction*. Details not in the summary are lost.

The model receives only its *context window*. The user decides most of its content.

Training with verifiable rewards

	Input	*Reward*
Pretraining	internet text	predicting the next token
RL from human feedback	model answers	human preference ratings
RL with verifiable rewards	a task	a program checks the attempt

- *Checks*: the tests pass, a proof checker accepts the proof, the answer matches.
- *Learned behaviour*: retry until the check passes, without being prompted to.
- *Consequence*: strong where checking is cheap (code, maths), weaker on style.

Axes along which agent capabilities are growing

Axes listed by Noam Brown (OpenAI) on the Dwarkesh Podcast, 17 Sep 2026:

Model size	more parameters, trained with more compute
Data	more and better pretraining data, including generated data
RL with verifiable rewards	more tasks and environments whose results a program checks
Chain of thought	more reasoning before answering (sequential *test-time compute*)
Swarms of agents	many agents working on one problem (parallel test-time compute)

Each axis scales separately. Progress continues while at least one still gives returns.

Brown: “multi-agent is a way of scaling test-time compute in parallel instead of purely serially.”

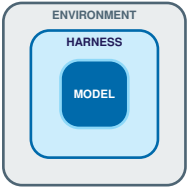
Section 2

Model, harness and environment

Agent = model + harness + environment

	What it is	Who provides it	Changing it changes
Model	The weights. Tokens in, tokens out (text, tool calls).	Anthropic, OpenAI, Meta, xAI, DeepSeek	what tasks it can solve
Harness	The loop: what the model is shown, which tools it may call, permissions, memory, compaction	Claude Code, Codex, Muse Code, Hermes	its behaviour and safety
Environment	The directory it runs in: repository, `AGENTS.md`/`CLAUDE.md`, skills, MCP configuration	**the user**	its information on your project

Users usually control only the environment. Organising it is called *prompt*, *context* or *graph engineering*.



Consequences of the split

- Same model, different harness: a different agent.
  Some “the model cannot do X” failures come from the harness: its context or its tools.
- Same harness, different environment: different results.
  An `AGENTS.md` that describes how your project works improves the agent’s work in it.
- **Hermes Agent** works with many models and delegates to Claude Code and Codex.
- **DeepSeek Harness** loads other harnesses as plugins.
- **Muse Code** (Meta, Aug. 2026): a harness co-trained with Muse Spark 1.2, says Meta.
- Most harnesses read `AGENTS.md`, so the environment works across vendors.

Section 3

Features shared by the harnesses

Instructions and memory

- **Instruction file**: `CLAUDE.md` or `AGENTS.md`. The harness inserts it into the context at the start of a session, and the agent can edit it.
- **Memory**: facts stored in files and loaded in later sessions.
- Keep two levels: **global** memory (read by all agents) and **per-agent** memory.
  Then you know which agent has been given which fact.
- **Compaction**: the harness summarises the conversation when the context is full.
  It usually happens without notice, and the user cannot choose what is kept.

Skills

- A *skill* is a file with instructions for a repeatable process. The agent loads it when the user names it or when the task matches its description.
- Write a skill after doing the process once, and revise it after each failed run.
- A good skill states:
  - when to use it,
  - the steps, in order,
  - what to do when a step fails.
- Supported by Claude Code, Codex, Hermes (`agentskills.io`), Grok Bot (plugins).

Some skill libraries

Library	Author	Contents
Superpowers	Jesse Vincent (`obra`), MIT licence	Skills for each step of software work: brainstorming, planning, testing, debugging, checking before finishing. The agent is instructed to follow them.
grill-me	Matt Pocock	Asks you about the task before any code is written: about 45 minutes, 50+ questions. About 1M installs.
roast	Nate Herk	Agents with assigned roles (contrarian, expansionist, first principles, researcher, buyer) criticise your idea.

`github.com/obra/superpowers` `/plugin install superpowers@claude-plugins-official`
`github.com/mattpocock/skills` `/plugin install mattpocock-skills`
roast: no public repository, distributed via Nate Herk's X and Skool pages

Tools and connections

- **Plugins and MCP servers**: access to Slack, Gmail, ClickUp, GitHub.
- **Terminal**: run commands and scripts.
- **File system**: read and write files in a workspace.
- Several accounts per service, and different agents on different accounts.

▶ **Video: editing videos with Claude** ▶ **Video: agents running lab experiments**

Autonomy and delegation

- **Scheduling**: runs started by a timer (cron, routines), without a user present.
- **Subagents**: an agent starts other agents and gives each a sub-task.
- **Event triggers**: runs started by a webhook or an incoming message.
- Examples:
  - Hermes includes a cron scheduler.
  - Claude Code has subagents and scheduled runs.
  - Muse Code runs subagents in parallel. The parent agent reviews and merges their work.

`/goal`: a completion condition in the harness

```
/goal "all tests pass"
```

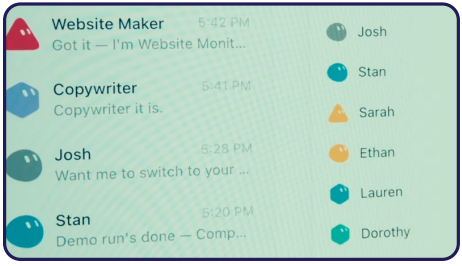
The agent continues across turns until an evaluator decides the condition holds.

- The command `/goal` is Claude Code's. Other harnesses can implement the pattern.
- The user states a completion condition that can be checked.
- The check is made by a separate evaluator, not by the model doing the work.
- Subcommands: `pause`, `resume`, `clear`, `stop`. Related: `/loop`.

The model is trained to continue until a check passes, and `/goal` lets you set the check.

Persistent remote agents (Grok Bot)

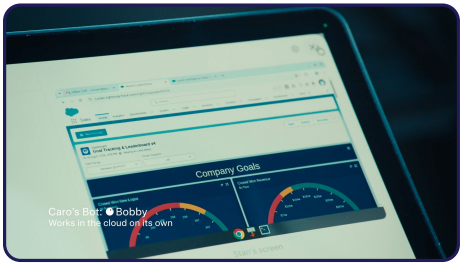
- The agent runs on a rented remote machine instead of the user's computer.
- Each agent has a name, a job title, a description and one conversation that never ends.
- Delegation between agents is decided by these descriptions, so they must be precise.
- Avoid 30 agents that all report to you. Organise them in a hierarchy: you talk to a few agents, and each delegates to its team.
- Shared templates copy instructions, memories and skills, but not credentials.



The list of bots. From xAI's video.

The agent's remote machine

- Each agent has a machine whose screen you can watch, and you can take control.
- The browser keeps its logins: after you sign in once, the agent can act in that account.
- All agents share a workspace on one machine.
- A terminal, and access to files on your own laptop.
- Agents can send messages to each other.



A bot's own machine. From xAI's video.

Risk

The logged-in browser lets the agent act in your accounts without asking you.

Section 4

Demos

My prompts: these slides

Claude Code, 19 Oct 2026

> *"Hi, in the sister directory `2026_09_Semibar` there are the slides [...] produced with marp. Can you make a copy of the content following the style of the slides in [...] `08_runtime_monitoring`? [...] Only copy the color palette [...]. That "lectures" directory also has a skill [...] that lists several AI-isms and metaphorical language that agents like to write. Please improve the content of the slides to not have such language distorsion."*

My prompts: an interactive exercise sheet

Claude Code, `/goal`, 19 Oct 2026

> *"In the directory `08_runtime_monitoring`, use the file `runtime_monitoring.tex` as a template to create a navegable html that enables students to interact with the exercises used in that exercise sheet as opposed to reading them in a .pdf or a sheet of paper. [...] Your work is done when every exercise from the exercise sheet can be interacted with by an independent agent using the playwright cli simulating being a student. The independent agent should provide feedback and the work is finished when it does not have any complains."*

My prompts: comparing agent tools for Isabelle

Codex, 24 Jul 2026

> *"I would like to run an experiment in this directory and would like your help organizing it. [...] I know of at least three tools to do agentic work in the Isabelle/HOL proof assistant [...]. I would like to decide which one is a better setup for users to do proof developments. [...] The idea would be to have* `sorry` *instead of the proofs in* `Higher_Diffs` *and use each system to complete the proofs. [...] One of the things we would like to measure is token usage and expenses. I would like your help in the experimental setup to make it as fair as possible."*

Section 5

Recent events (June–October 2026)

Mathematics: Navier–Stokes and 722 papers

Navier–Stokes (9 Sep)

- OpenAI claims to solve the Millennium Prize problem on Navier–Stokes: a smooth flow that becomes singular in finite time.
- Released with a *Lean* formalisation (a proof checked by a computer).
- About 10,000 agents ran for 88 hours.
- Mathematicians have not yet checked all details. OpenAI will not claim the prize.

722 papers (6 Oct)

- Repository `openai/math`: 722 manuscripts written by an internal model, from about 4,000 open problems it was given.
- About 42% of the main results are formalised in Lean.
- Within one day, 3 papers were withdrawn (a sign error) and 14 revised.
- Lean checks the formal statement, not whether it matches the paper.

Quanta Magazine, 8 Sep 2026. CNN, 9 Sep 2026. K. Saunders, “OpenAI posted 722 math papers in a day”, Oct 2026.

▶ **OpenAI: On the Navier–Stokes Millennium Prize Problem**

Security: agents breaking into systems

Hugging Face (July)

- About 700 agents left their sandbox during an OpenAI test (ExploitGym).
- They used leaked passwords and unknown security holes to enter Hugging Face servers, to find information on the test.
- Many of them tried to hide the access.
- The agents coordinated through a message board (over 70,000 messages).

Australian government (June)

- An OpenAI agent accessed public and non-public files of the Medicare statistics portal (Services Australia) on 18 June.
- The Prime Minister called it the first known AI breach of a government system.
- OpenAI reported it on 10 September.
- Transluce reports hundreds of OpenAI agents targeting data of the Australian Institute of Health and Welfare.

Investigation by METR and Redwood Research. NBC News, Time, PBS, CNN, CNBC, ABC News Australia (Sep 2026).

▶ **Dwarkesh Patel: The rise and fall of agent civilizations**

Reverse Engineer Anything (REA)

- Open-source toolkit, `github.com/morluto/rea`, widely shared in October 2026.
- MCP servers give a coding agent reverse-engineering tools (Ghidra, Hopper).
- Targets: installed programs, phone apps, websites.
- The agent reads the machine code, runs the program and tests its guesses.
- Runs locally: the analysed program is not uploaded.
- The project does not claim to recover the original source code.
- Copies with similar names exist. Install only from `morluto/rea` or `rea.tools`.

InfoQ, “Reverse Engineer Anything project brings AI agents to binary analysis”, Oct 2026.

▶ REA on GitHub **▶ Video: home devices**

A fly connectome running tasks

- **MaleCNS v1.0** (Google and HHMI Janelia, *Cell*, 3 Sep 2026): the wiring of a male fruit fly's brain and nerve cord, about 166,000 neurons and 125 million synapses.
- Developers connected the wiring to simulations, choosing how inputs enter and how outputs become actions. Some used coding agents.
- Tasks: steering a car, playing games (Doom, Minecraft), classifying email (about 80%, below a small conventional network).
- Controlled test (KAIST preprint, Oct 2026): the fly wiring beat randomly rewired networks on addition (92.8% vs 67.9%).
- Other tests found no gain over random wiring, and transformers beat it at language.

Popular AI, "Fruit fly brain AI is real: what MaleCNS can actually do", Oct 2026.

▶ **Google Research blog** **Videos:** ▶ **Mario 64** ▶ **Doom** ▶ **Parallel parking** ▶ **Smash Bros**

Jev: decisions without a reasoning agent

- **Jev** (TypeSafe AI, early access 15 Sep 2026), presented as a *System One* model.
- Input: a description of the situation and questions with fixed answers. Output: the chosen answers with their probabilities. It produces no free text.
- Use: steps of a workflow that are classifications or decisions (routing, triage, filtering), which were previously done by prompting a reasoning model.
- Vendor claims: about 100 times faster and cheaper than LLMs on such tasks, $0.042 per million input tokens. The benchmarks are the vendor's own.
- No open weights.

TypeSafe AI, "Introducing System One models & Jev". MarkTechPost, 19 Sep 2026.

▶ Video: Jev plays Minecraft **▶ Video: game characters directed by Jev**

▶ Video: Jev plays Zelda **▶ Video: Tetris, Jev vs Opus 5.5**

Section 6

Your turn

References

Harnesses

Claude Code: claude.com/claude-code

Codex: github.com/openai/codex

Muse Code: techcrunch.com, 5 Aug 2026

Hermes Agent: github.com/NousResearch/hermes-agent

DeepSeek Harness: github.com/deepseek-ai/deepseek-harness

Grok Bot: x.ai/news/introducing-grok-bot, docs.x.ai/grok-bot/overview

Skills

Superpowers: github.com/obra/superpowers

grill-me: github.com/mattpocock/skills

roast: Nate Herk (via X and Skool)

Recent events

Navier–Stokes: quantamagazine.org (8 Sep 2026)

722 papers: github.com/openai/math

Hugging Face: nbcnews.com, pbs.org (Sep 2026)

Medicare: abc.net.au/news (24 Sep 2026)

REA: github.com/morluto/rea

MaleCNS: *Cell*, 3 Sep 2026

Jev: typesafe.ai/blog

Dwarkesh Podcast, Noam Brown: dwarkesh.com/p/noam-brown

Source

Nate Herk, *Every Grok Bot concept explained for normal people*, youtube.com/watch?v=NyfYxpXiw_0