

The ATS Agent

What it is, how it works, and how your team talks to it.

Built 2026-07-14 · hardened 2026-07-15 · for Anthony

TL;DR

- The ATS Agent is a standing recruiting agent (same framework as Pam) that both advises you on hiring decisions and helps the team execute screening, interview prep, and candidate comms. It reports to Toby.
- Jose and Homer only ever talk to Toby, who relays to the ATS Agent through an identity-gated tool. This keeps the confidential Vegas AM search fenced off without Toby losing his normal access to everyone else.
- You and Jose reach it directly on the Vegas Hiring tab: teach it rules, approve/reject proposals, thumbs up/down answers. Homer gets thumbs and a bug button, not the teach box.
- Nothing it's taught becomes policy until it passes 2-3 held-out real cases graded independently. That proof (`learned 3/3`) is what separates "I taught it" from "it generalized."
- A two-model hardening pass fixed 8 real problems yesterday (access control, race conditions, a broken spend cap, a compliance-checker gap). All re-verified live. One thing is still outstanding: nobody has sent Toby a real recruiting question yet.

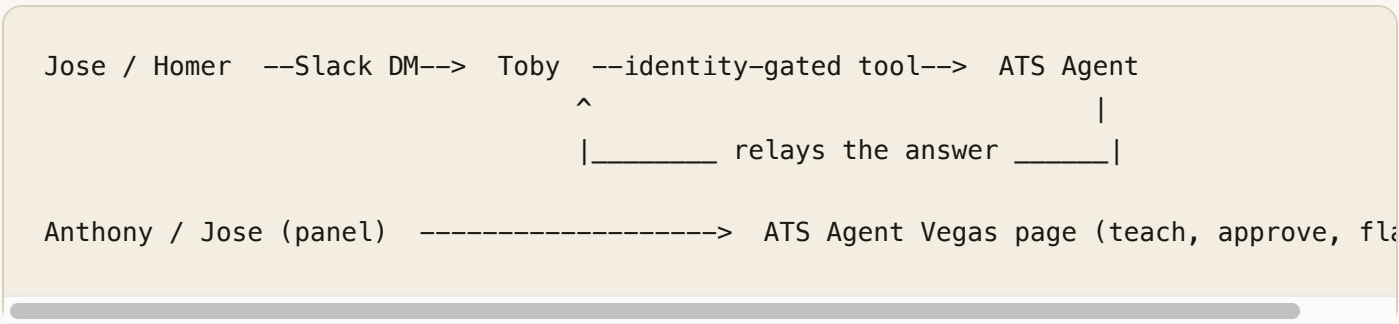
What it is, in one paragraph

The ATS Agent is a standing recruiting agent, built on the same framework as Pam (your CS agent). It works two jobs at once: an ATS director that advises you on hiring decisions (funnel metrics, req priorities, offers, compliance), and a recruiting manager/interviewer that helps the team execute (screening, interview prep, candidate comms). It reports to Toby, the same way a recruiting coordinator would report to your HR manager. It has its own memory, learns from feedback your team gives it, and every rule it learns is proven before it ships, not just accepted on faith.

How your team talks to it

The only thing Jose and Homer need to know: talk to Toby. They never need to know the ATS Agent exists by name, or that Toby is relaying to it behind the scenes.

When Jose or Homer ask Toby a recruiting question in Slack, Toby does not answer from his own memory. He is required to hand the question to the ATS Agent, get its answer, and relay it back like a manager would. This was a deliberate build choice: Toby already talks to every employee with no restrictions, and the Vegas AM search running through the ATS Agent right now is confidential. Piping the question through a locked door (see below) means Toby can serve everyone else normally while this one topic stays fenced.



Where you and Jose reach it directly

The shared ATS page (Vegas Hiring tab) is where you and the panel work with it hands-on:

FEATURE	WHO	WHAT IT DOES
Teach box	You + Jose only	Free-form notes that become candidate rules for the agent's memory. Locked to the hiring panel as of today; Homer gets a friendly redirect to the bug button instead.
Thumbs up/down + flag answers	Everyone with page access	Lightweight signal, stays open to Homer too, feeds the same learning loop without letting him author rules directly.
Approve / Reject proposals	You + Jose only	A drafted rule only becomes real once one of you approves it, and only after it proves itself (below).
Bug button	Everyone with page access	Homer's path for anything wrong. Goes to a separate report queue, not the KB.

How it actually learns (not just "we told it something")

A note in the teach box does not become policy just because someone approved it. Before a taught rule ships to the agent's live memory, it has to answer 2-3 real past cases it was never shown, and an independent grader checks its answers against what actually happened. Pass, and the rule ships tagged with proof (like `learned 3/3`). Fail, and it comes back to the Today page with the case it missed, so you can see exactly where the lesson didn't stick. This is the difference between "I taught it" and "it generalized," and it is the part of the build most likely to quietly break if anyone ever touches the code without re-running the proof.

What changed in yesterday's hardening pass

A full two-model code review (two different AI model families, so blind spots do not overlap) found 8 real problems in the build. All are fixed and re-verified live:

- **FIXED** Homer could previously approve rules straight into the agent's authoritative memory, more access than your own team had through Toby. Now locked to you and Jose.
- **FIXED** A channel-thread Slack conversation could let someone else's reply ride an authorized person's identity into the confidential search. Delegation now only works in a direct message.
- **FIXED** An approved rule's text was never actually run through the compliance checker before shipping. It is now, and it fails closed (holds the rule) if the checker can't run.
- **FIXED** The spend safety cap (meant to halt the agent if it runs too often in a day) was silently broken, watching a log file that never existed. It now counts real runs correctly.
- **FIXED** Two people clicking Approve around the same time could silently lose one click. Both write paths now share a lock.
- **FIXED TODAY** The compliance checker's age-question detector only caught one exact phrase. Widened to catch birthdate, graduation year, retirement-year questions, and a few other protected-class categories that were similarly narrow.
- **FIXED TODAY** The ATS page's buttons are now part of your nightly automated build-verify check, so a broken button gets caught overnight instead of by someone clicking it.
- **FIXED TODAY** When Toby relays a candidate-facing draft (an email to a candidate, for example), it now also passes a stricter candidate-communications check, not just the internal one.

Where things live (for whoever touches this next)

<code>~/cron-launchers/ats-bot/</code>	Agent home: brain, memory, tools, logs
<code>AGENT.md / SOUL.md / RULES.md</code>	Your instructions to the agent (not its memory)
<code>data/kb/ats-learned.md</code>	The agent's own memory, written by the feedback loop
<code>vegas-hiring/index.html</code>	The shared ATS page (ATS Agent tab)
<code>tools/ats_connectors.py</code>	The locked door Toby uses to reach the agent
<code>data/logs/build-ledger.jsonl</code>	Proof log: every build/hardening step and what verified it

Still outstanding

One thing only you can close: send Toby a real recruiting question in Slack. Every other part of the loop has been tested end to end; this is the one link no human has actually exercised yet.

Smaller, optional items: the Chargeback Agent (a separate, unrelated build) has a cron that never got loaded, unrelated to anything here. And two open questions from the original build review are still unanswered: whether flags should also ping Slack, and whether the agent should triage the Vegas hiring inbox daily.