Agent factory

Hand work to cloud coding agents and keep every change reviewable. Four planes, five tools and a team harness repo, for a team of six running about sixty agent sessions.

Team harness	Defines what the agents run	Define
hangar	Runs the agents	Run
flightdeck	Judges the runs	Judge
Tower	Governs risky actions	Govern
ai-dlc	Sets up machines and the workflow plugin	Setup
Spaceport	Serves data to agents	Capability

— The problem

Sixty agents at once breaks the way a team works

Live sessions

~60

Six engineers, about ten streams each. Nobody can watch them all.

Review

1 gate

Every change still needs a human. Reviewers become the bottleneck.

Harness changes

0 evidence

A new skill or prompt is judged by impressions, not paired runs.

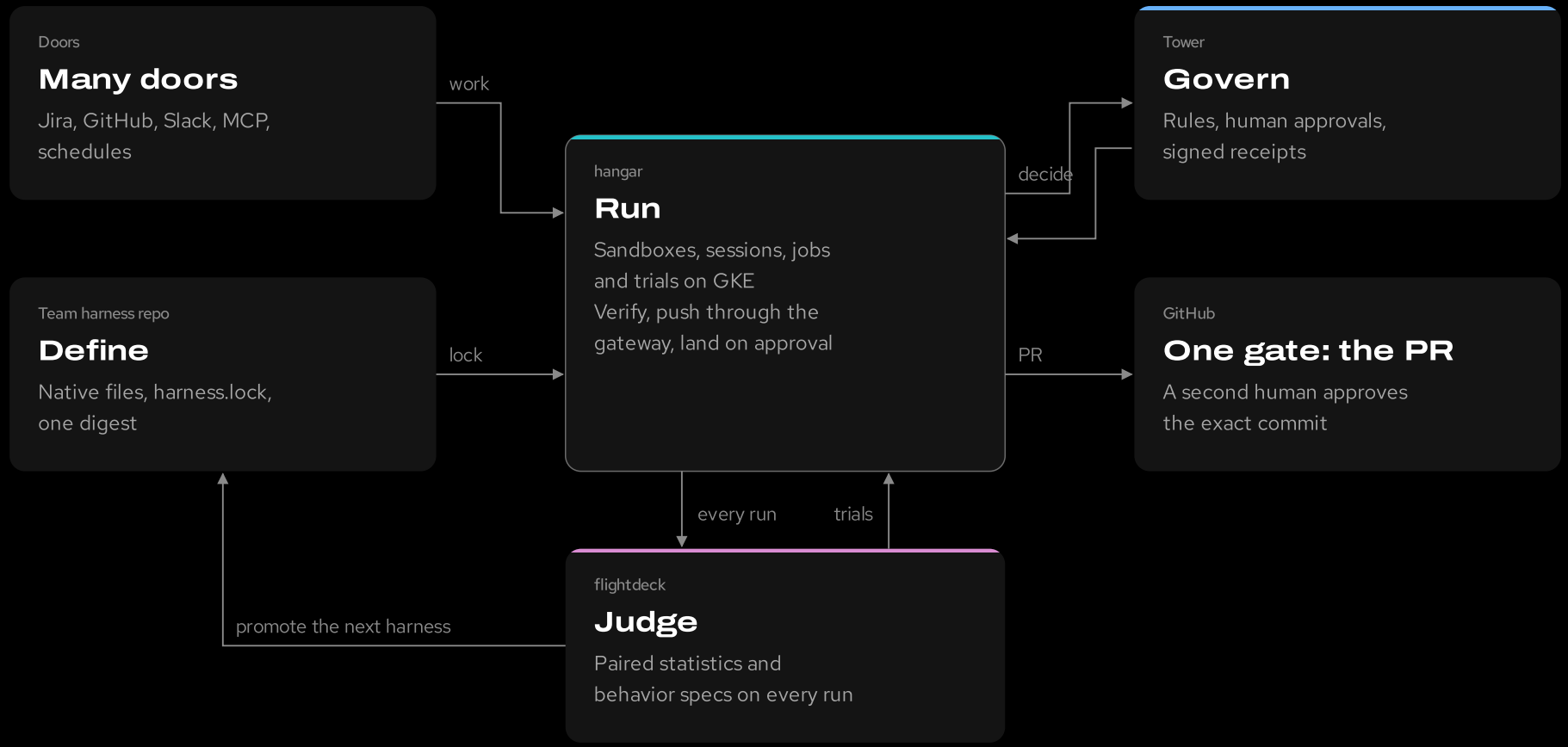
Untrusted input

Every ticket

Ticket text and web pages can steer an agent toward a push.

Nobody can watch sixty streams; the tools must bring what needs a person to them

Four planes, one loop, one gate



Each tool owns one plane and meets the others at versioned contracts

A Jira ticket to a merged PR

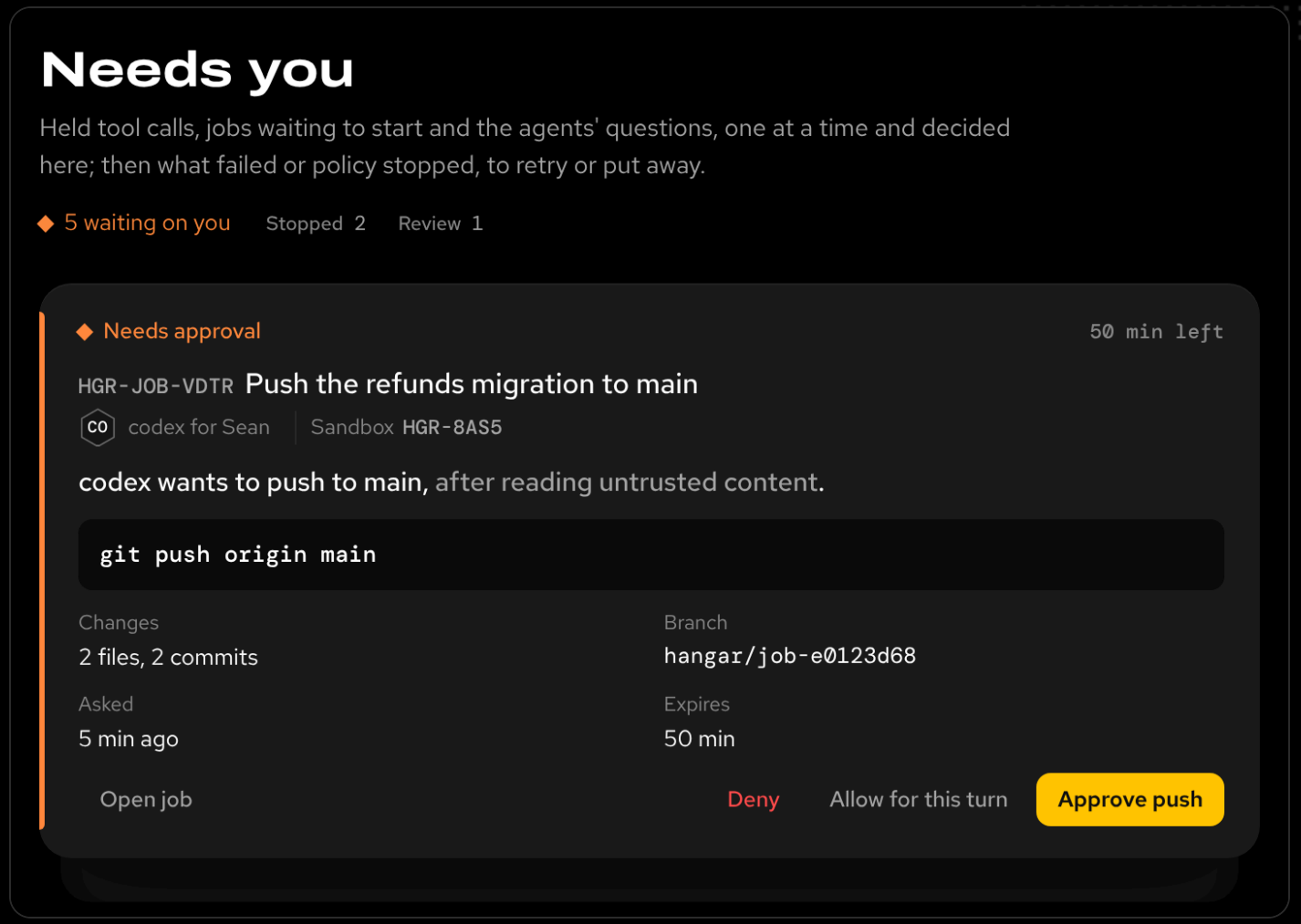
01	**Jira**	An engineer moves the ticket to “Ready for agent”	Queued
02	**hangar**	Ticket text is untrusted, so the job starts tainted	Admitted
03	**Sandbox**	The agent works with a pinned harness; every step is traced	Running
04	**Gateway**	hangar verifies, then pushes one commit as the GitHub App	Delivered
05	**Reviewer**	A draft PR waits for a second human to approve the exact commit	Needs approval
06	**Outbox**	The merged run is written once, for flightdeck to judge	Captured

Only approvals and failures light up; everything else stays quiet

Twenty-six agents. Five need you.

Needs you is the console's first tab: held tool calls and questions, one decision at a time, soonest to expire on top.

- Decide and the next one rises; the agent continues the moment you do.
- A deny needs a reason. It goes on the receipt and back to the agent.



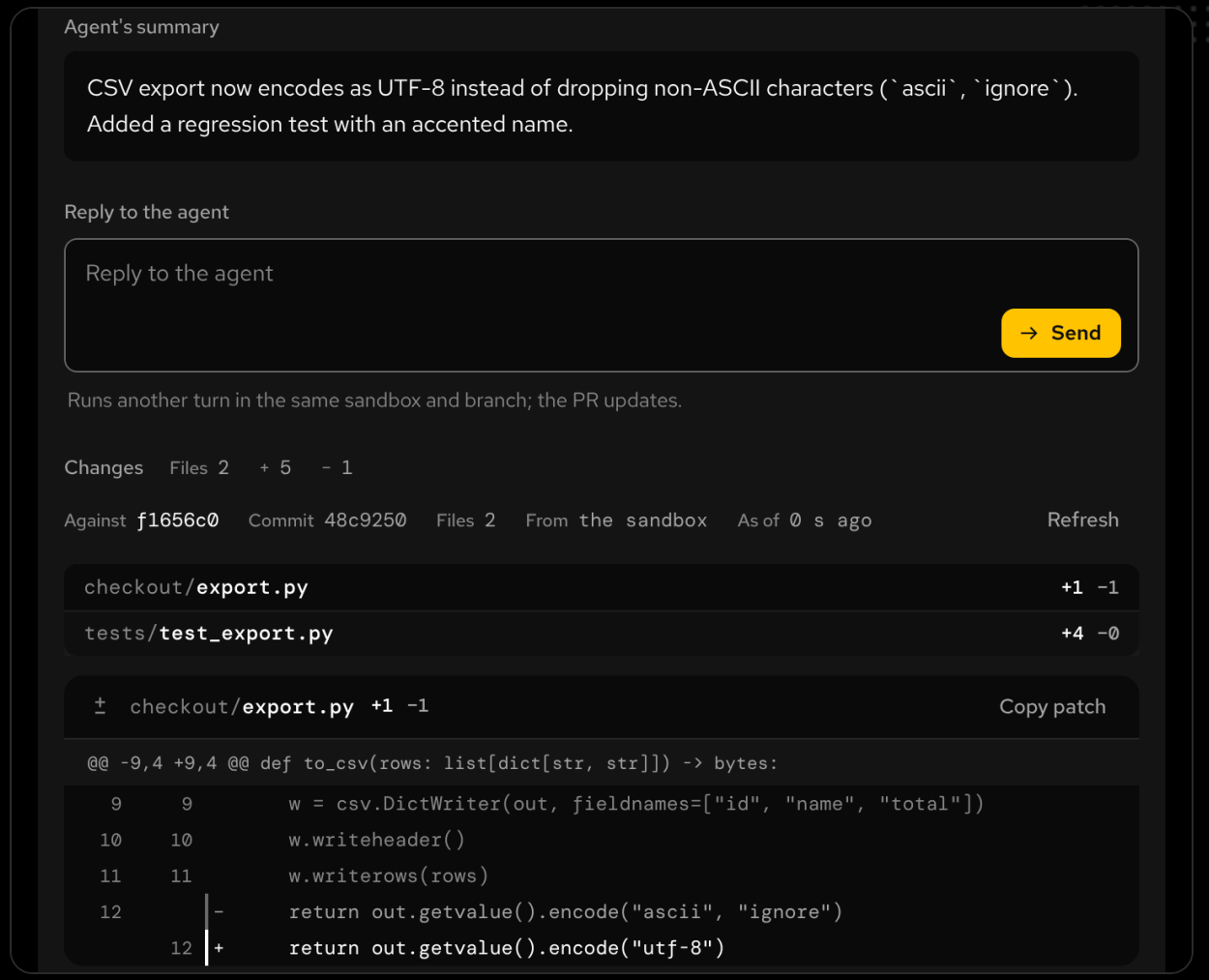
hangar, Needs you: the held push on top

Everything running stays grey; the one card that needs you is the only color

Review the change, reply to the agent

A job's detail carries the agent's summary, a composer and the change as a greyscale diff. Your reply runs the next turn; the PR updates.

- `+5 −1` across 2 files, against the base commit, from the sandbox.
- Only a failing test lights a hunk; additions and deletions stay grey.

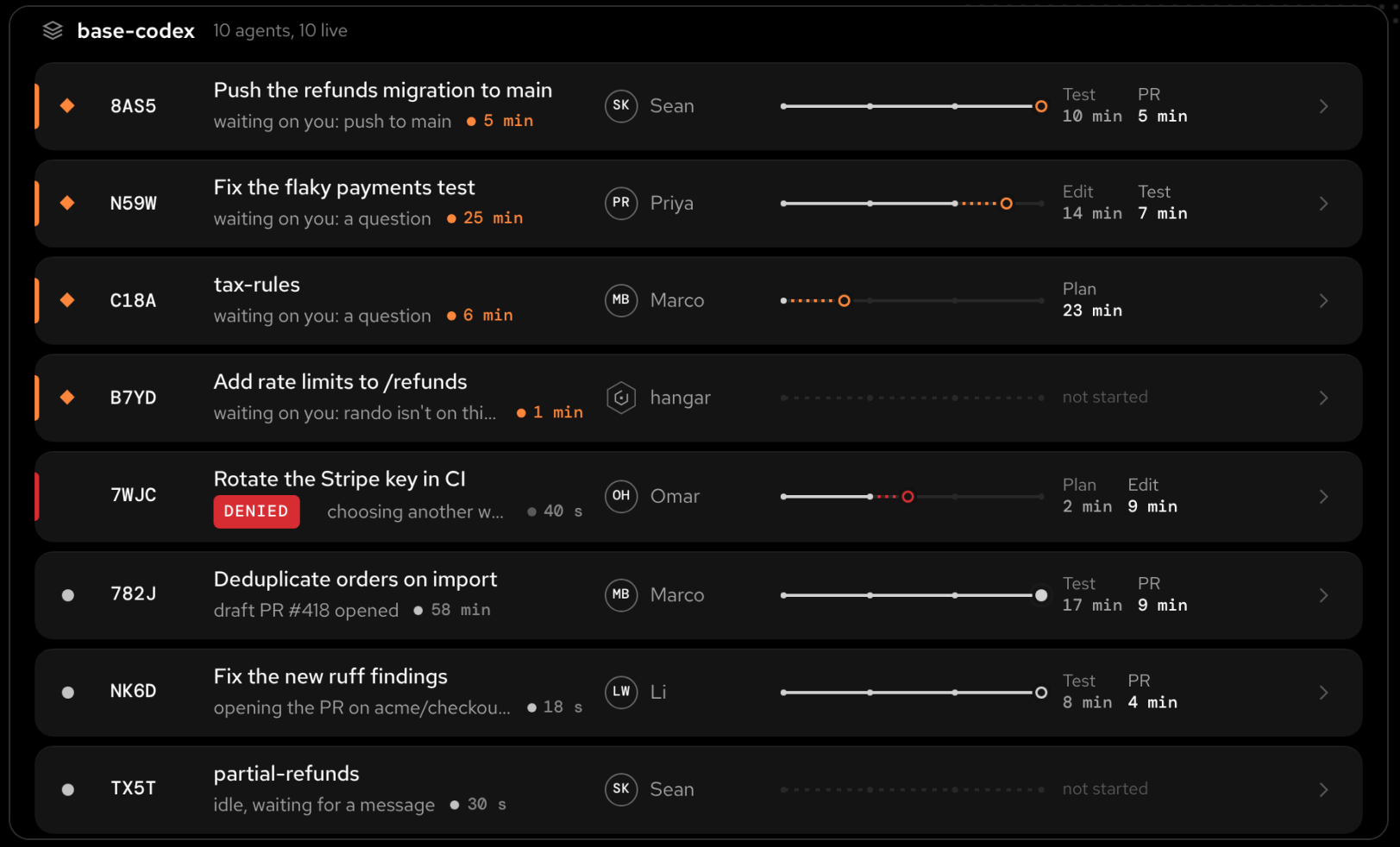


hangar, a job: the summary, the reply box and the diff

The whole fleet, one row per agent

List and Board are two views of the same agents. **25 of 120** slots in use.

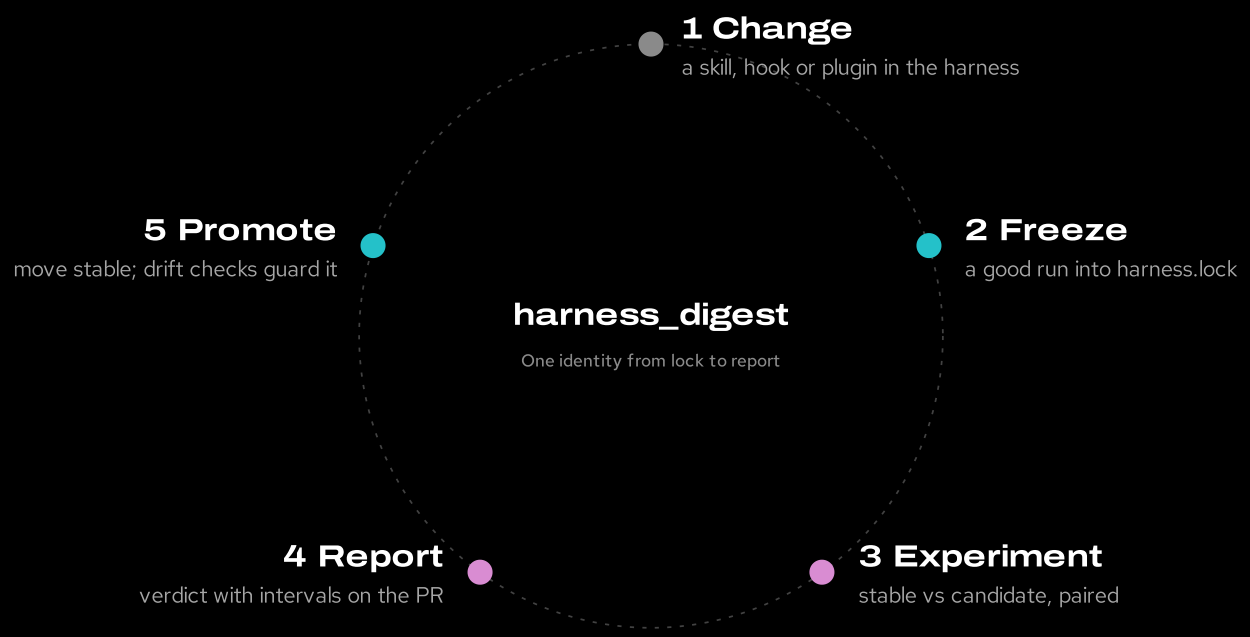
- Each row shows what the agent is doing now, its owner and its stage: plan, edit, test, PR.
- Open an agent in place to decide what it holds without leaving the Fleet.



hangar, the Fleet as a list

Agents at work stay grey; only the ones that need a person carry color

Change the harness on evidence, not impressions

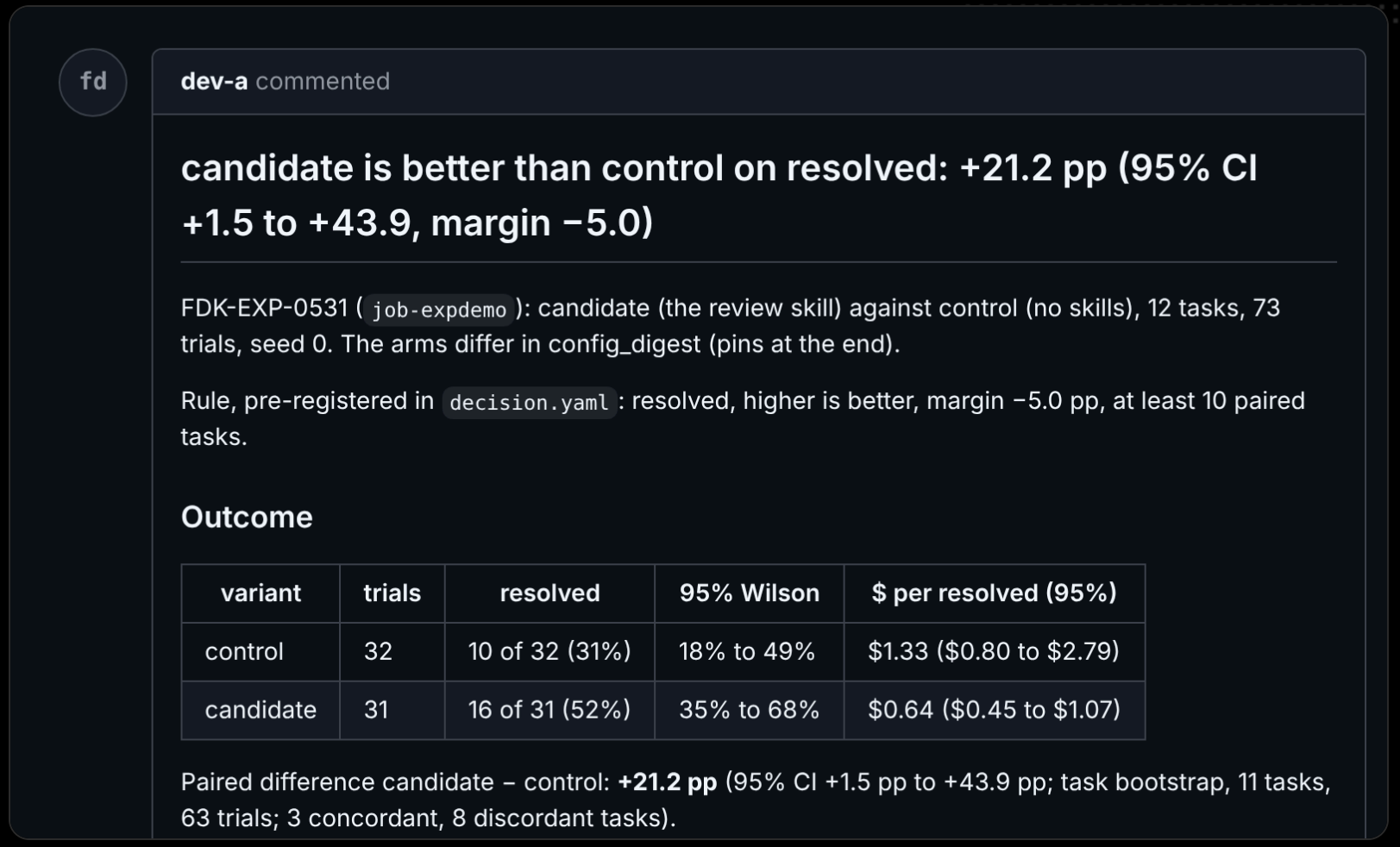


Infrastructure failures are retried and left out of every statistic

The verdict lands on the harness PR

flightdeck posts one comment on the pull request that changes the harness and updates it in place: the verdict, both variants' pins, the outcome and its interval.

- **Candidate is better than control** on resolved: +21.2 pp, CI +1.5 to +43.9.
- Paired by task; infrastructure failures are excluded and listed.



GitHub's comment on the harness PR, written by flightdeck

— Governance

Risky actions fail closed and leave a receipt

Taint

Tracked

Reading a web page, a remote MCP server or a ticket marks the run.

Push

Fetch-only

Sandboxes can't push. hangar delivers one commit through its gateway.

Decision point

Ask first

Release, landing, egress and secrets go to Tower: allow, deny or wait.

Approver

Never you

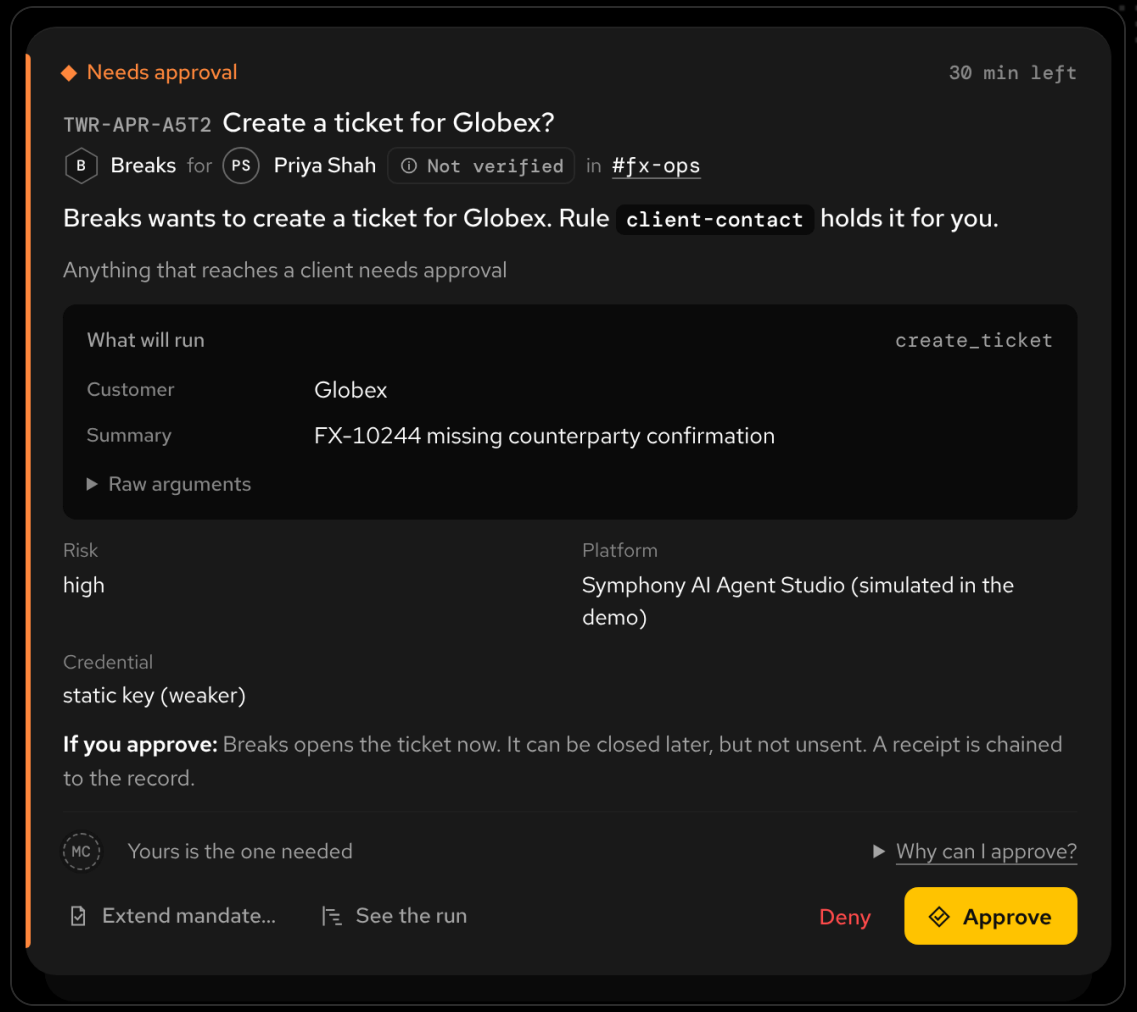
The requester never approves their own risky work. Every decision is signed.

Rollout: off, then shadow, then enforce, one decision kind at a time

The Desk: what needs you, what finished, who you approve for

Tower's home for a person: start work, approve or deny the cards that need you, and see the day's receipts.

- Each card says what will run, why it asked and what approving does.
- Every decision files a receipt, chained into the record.



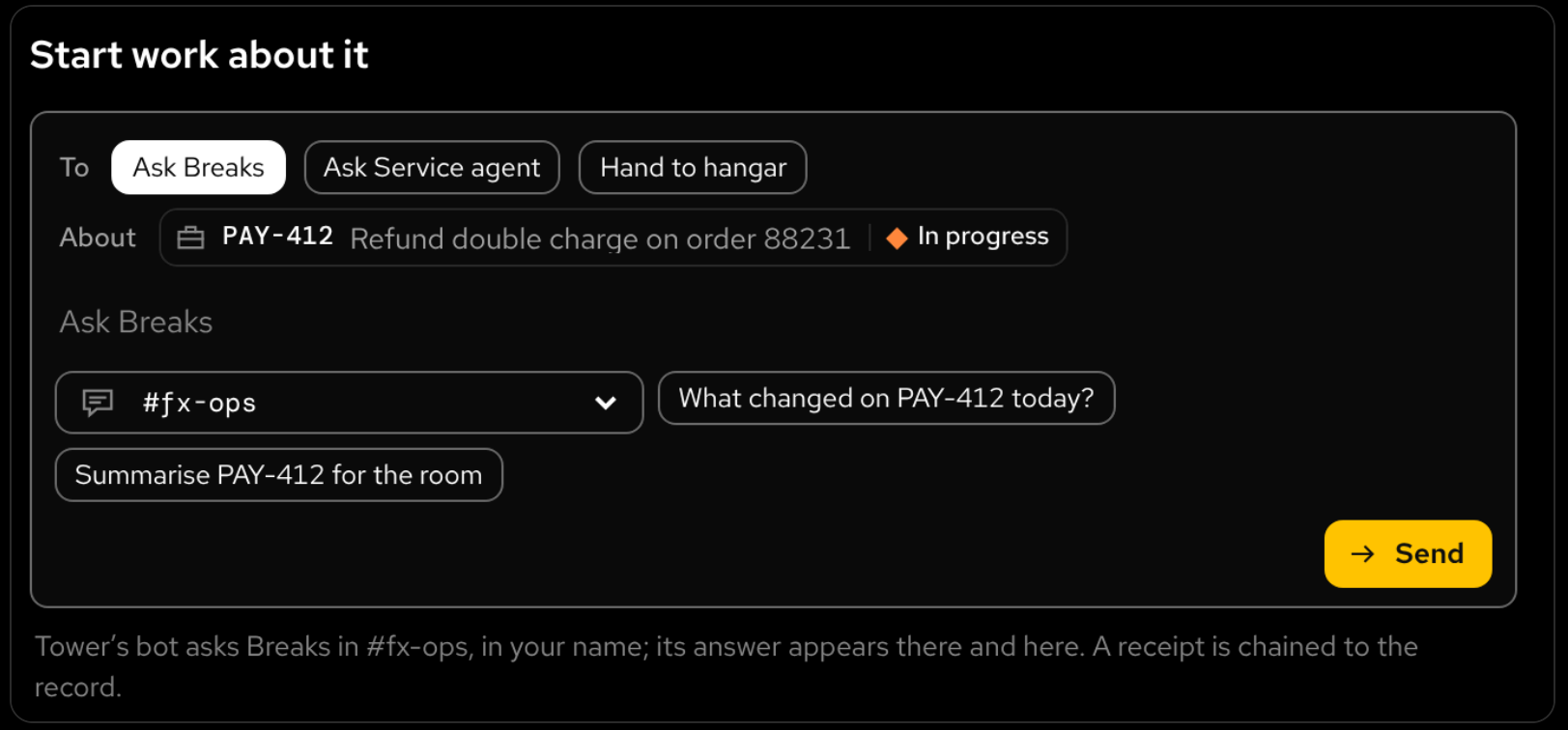
Tower, the Desk: one approval with Deny and Approve

What will run, why it asked and what approving does, before the button

Business objects by reference

A Jira issue, a pull request, a hangar job: each is a chip, and each chip opens a page with what touched it and what is in play.

- Tower keeps a reference, never a copy; title and state are read from the source.
- The receipt carries the reference, hashed into the chain.

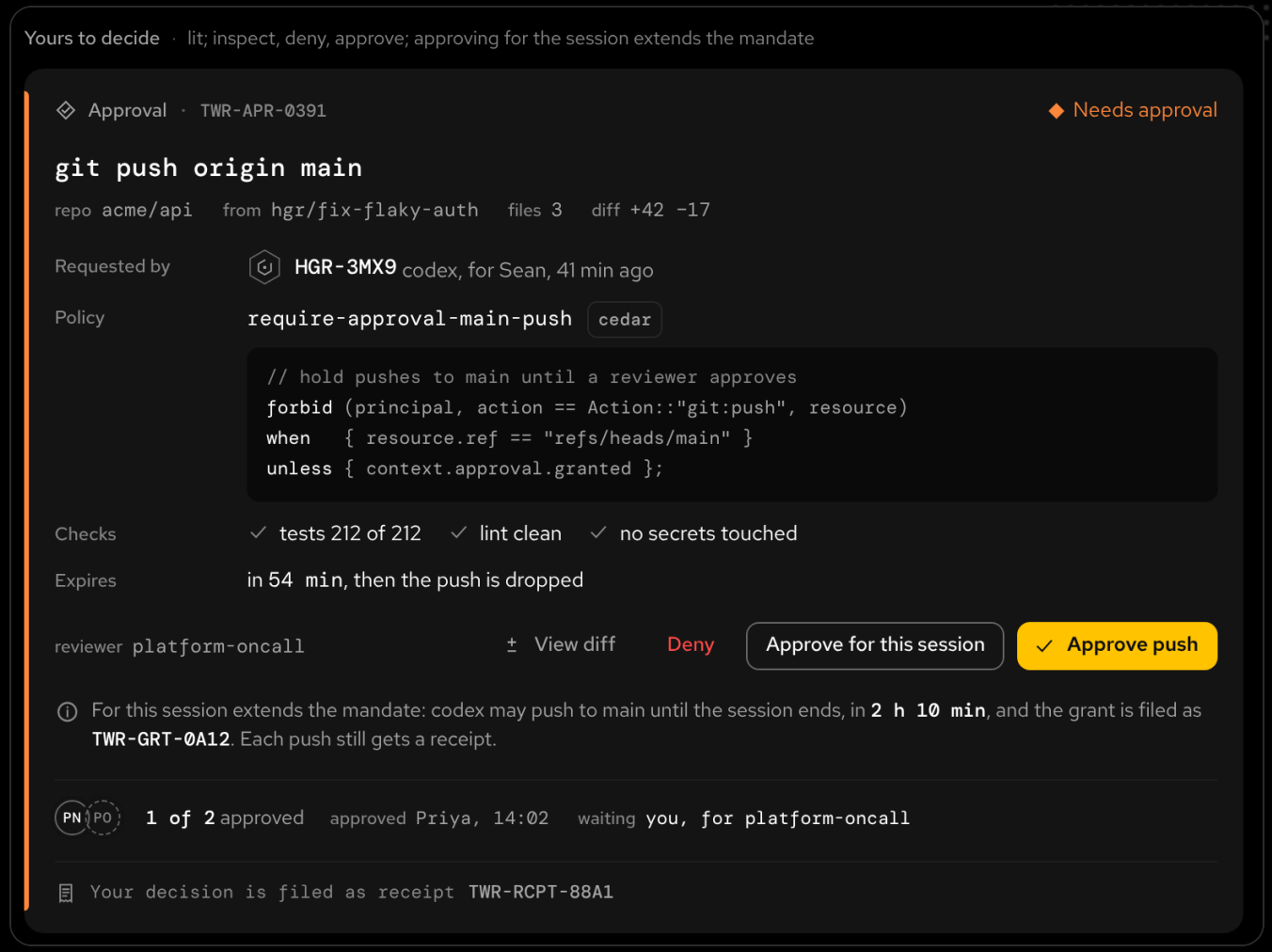


Tower, Jira issue PAY-412: start work about it

One design system, every screen

Airframe holds the parts a person needs to work with an agent: approval card, question, composer, diff, object chips. Every screen in this deck is built from it.

- Tokens and `af-` classes only; a test in each repo fails on a color literal.
- Night and Day, forced colors and reduced motion, in one stylesheet.



Airframe, the ApprovalCard

One stylesheet, every console and docs site; the tool repos vendor a tagged release

— Where it stands

Running and measured today, governed next

On main

Safe to run

- Sandboxes, jobs and sessions on GKE
- Intake posture, taint gates, sealed graders
- Manifests and traces with harness digests
- flightdeck data plane and statistics

On main

Measure everything

- Harness lock: freeze, check, thaw
- Run outbox and nightly compactor
- Reports, behavior specs, pinned trials
- Slack inbox, Tower's decision API

Next

Governed and decided

- Tower decisions in shadow, then enforce
- One real flightdeck decision
- Team harness, frozen from a good run
- Outbox and compactor on in the cluster