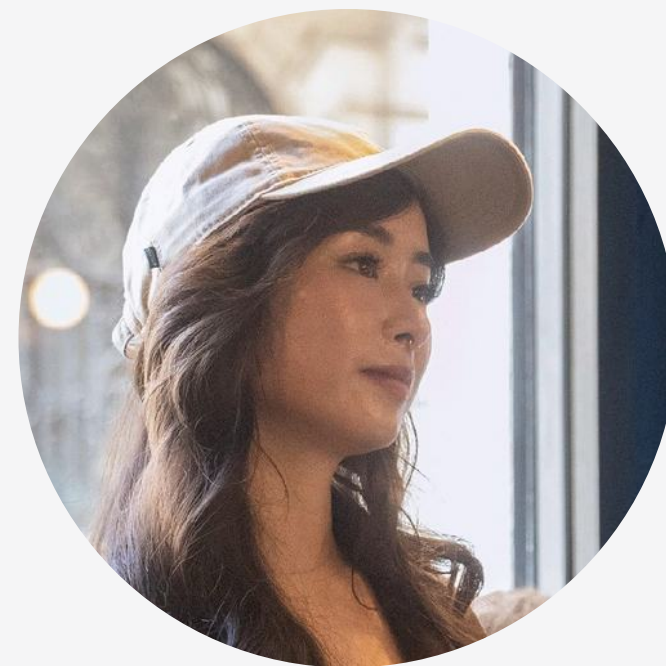


Enterprise grade,
startup made

2,000+ enterprise customers
3 months
3 engineers



Always

Be

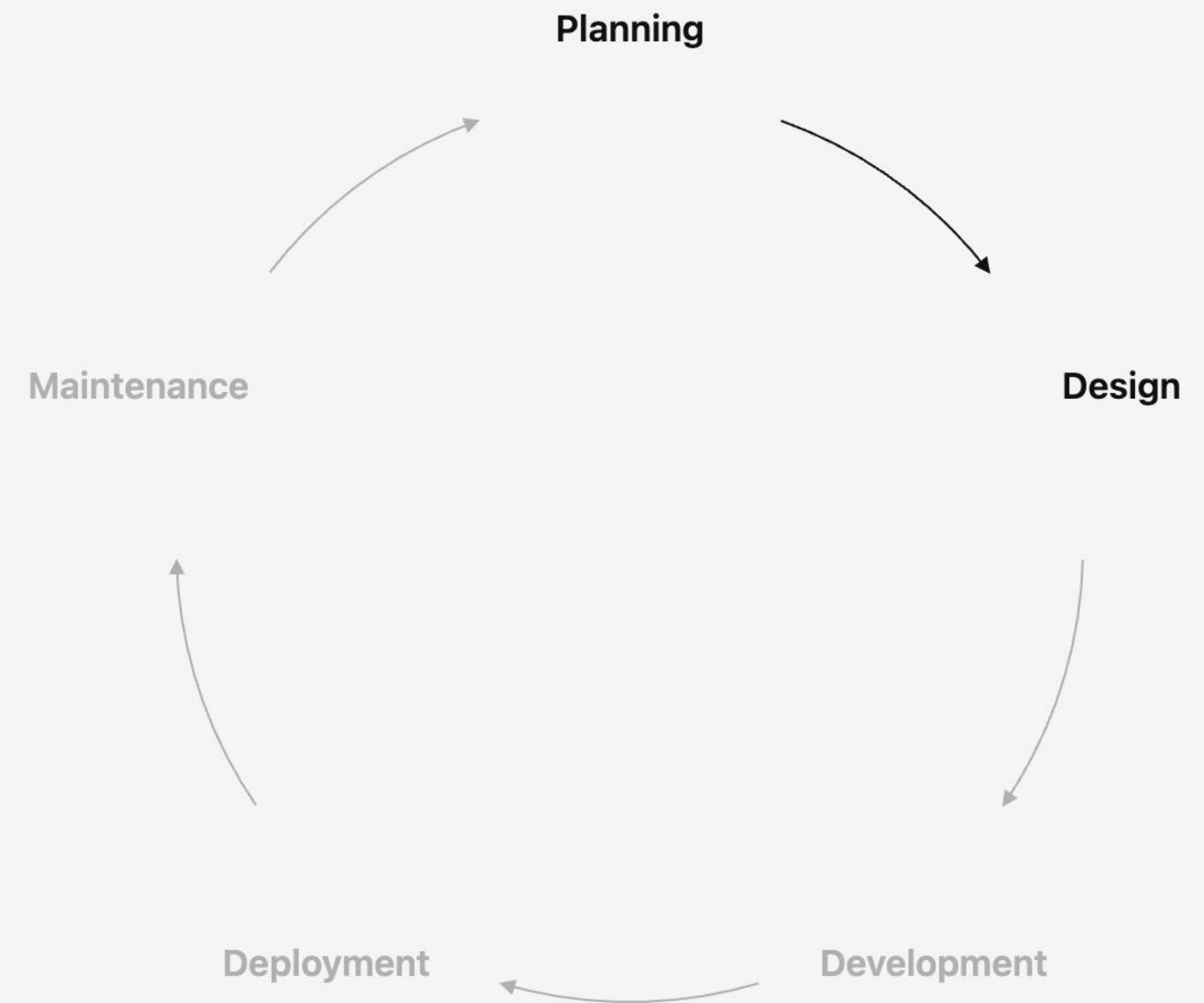
Carefully _ _ _ _ _

Driving

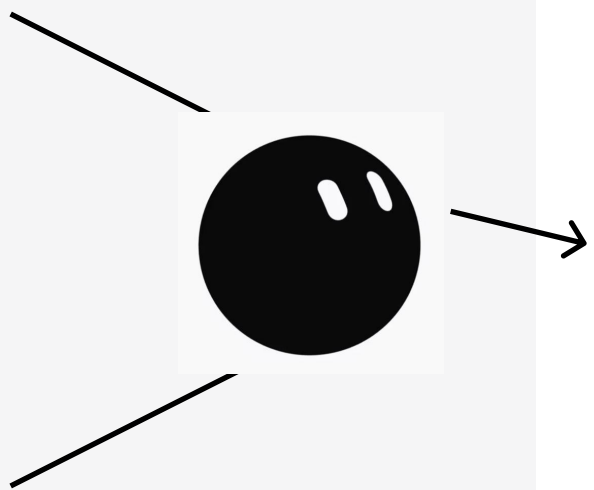
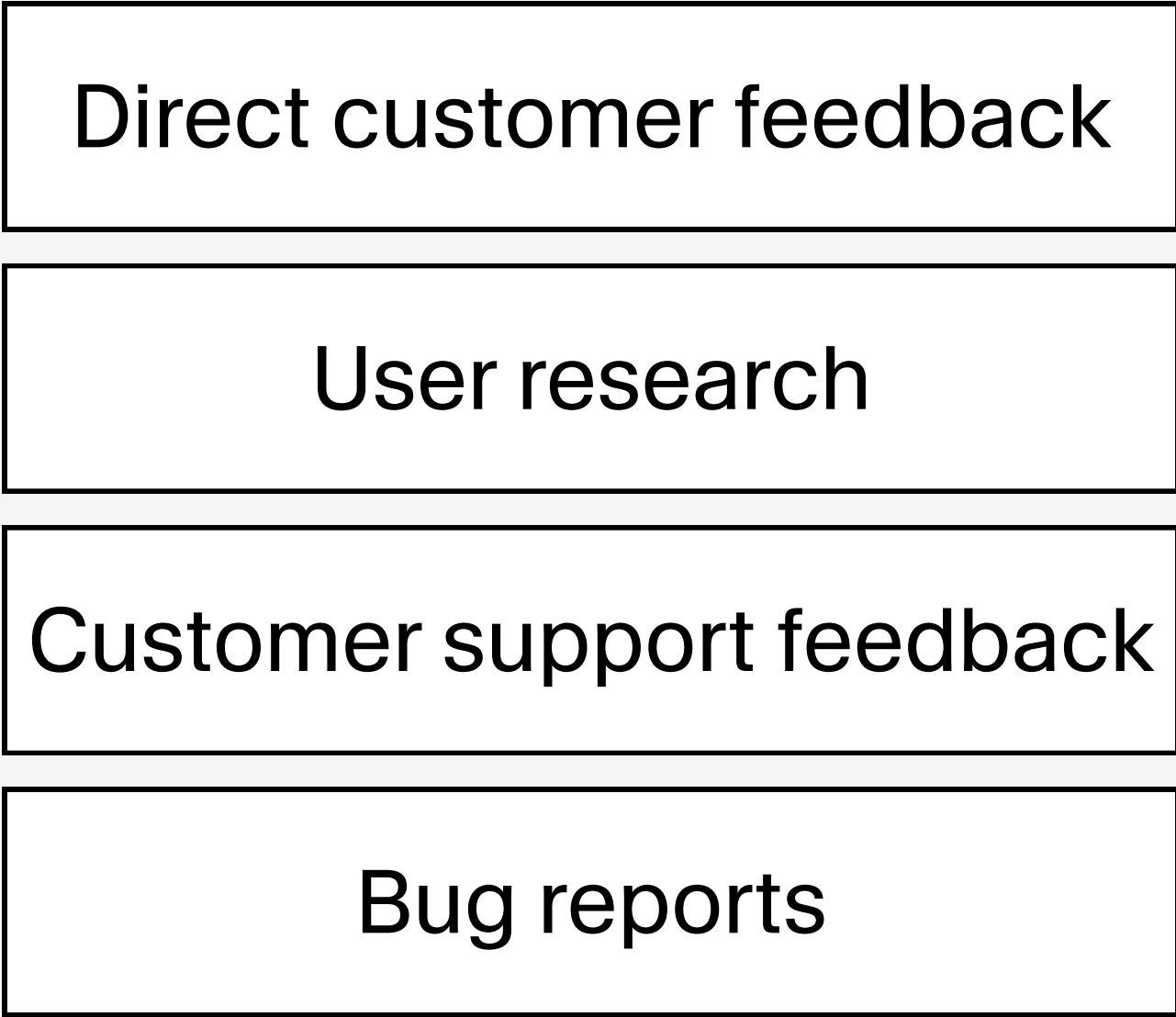
Everything

ForwardTM

Always
Be
Carefully planning
Driving
Everything
Forward



What **do** we need to build?



Commits To main: 5 Most Impactful Ships

Wednesday, April 22nd

- [New] <https://github.com/anysphere/eversphere/pull/103127> — *feat(anytool): organization create-team UX and WorkOS inheritance*
 - Why impactful: changes core org/team provisioning semantics (org-level WorkOS inheritance, billing behavior simplification), reducing enterprise onboarding ambiguity and misconfiguration risk.
- [New] <https://github.com/anysphere/eversphere/pull/98857> — *Org admin members: show only paid team memberships*
 - Why impactful: fixes confusing duplicate membership representations for org admins, improving trust in org member data and admin workflows.
- [New] <https://github.com/anysphere/eversphere/pull/103896> — *Aidan/paid wau growth segment tabs*
 - Why impactful: expands paid WAU growth segmentation in analytics UX; high leverage for product/revenue decision-making.
- [New] <https://github.com/anysphere/eversphere/pull/103891> — *remove v1 fireworks end state and swap to v2*
 - Why impactful: infrastructure migration cleanup that reduces dual-path complexity and aligns provider handling on v2.
- [New] <https://github.com/anysphere/eversphere/pull/103843> — *vscode: log slow DB close to sentry*
 - Why impactful: direct reliability/observability improvement for diagnosing shutdown latency issues in client runtime.

Top Customer Problems & Trends

- [New] Identity lifecycle + access integrity issues are still a top customer pain.
 - Examples: SCIM duplicate-account behavior, contractor-to-employee identity transitions, enterprise users appearing as free/missing model picker, login redirect loops.
- [Ongoing] Billing/usage transparency is the most frequent strategic request cluster.
 - Examples: Auto-mode model visibility requests, "unknown model" usage rows, dashboard mismatches, invoice line-item limit risk, BYOK cost-accounting confusion.
- [Ongoing] Admin governance controls are a major enterprise trend.
 - Examples: per-tool MCP controls, repo-scoped GH app enforcement, CLI disablement, audit-log retention questions, model allow/block granularity.
- [Ongoing] Cloud Agent operability and controls remain high-priority.
 - Examples: automations hanging, environment defaults at org scale, VM sizing requests, session/webhook behavior uncertainty.
- [New] From user feedback ingestion ([#automatic-user-feedback](#)): recurring reliability/UX regressions.
 - Frequent themes: AGENTS/rules adherence after compaction, terminal hangs, MCP long-response freezes, git scanning stuck, chat history loss concerns, billing perception ("double charge"/quota burn).
- [Ongoing] Most important fixes to prioritize from customer signal:
 - Reliability regressions in agent/runtime behavior.
 - Usage/cost transparency and attribution correctness.
 - Enterprise identity/governance controls with clearer admin

Latest messages

How to plan

Cursor -

Product Group Mapping (SCIM / SSO)

Saturday, March 21, 2026 · 1:30 – 2:00am

Join with Google Meet

meet.google.com/xmr-ajmc-ncb

Join by phone

More phone numbers

11 guests

6 yes (1 virtually)

4 maybe, 1 awaiting

10 minutes before

Going?

Yes

No

Maybe

Cursor <>

Product Groups

Wednesday, March 4, 2026 · 3:00 – 3:30am

https://anyosphere-co.zoom.us/j/89934639257?jst=3

6 guests

5 yes (1 in a meeting room, 1 virtually)

1 no

<> Cursor - Admin discussion

Thursday, March 12, 2026 · 12:00 – 12:30am

Join Zoom Meeting

Meeting host:

Join Zoom Meeting:

<> Cursor (Orgs Config)

Friday, March 13, 2026 · 12:30 – 1:00am

Join Zoom Meeting

ID: 89069979087

Meeting host:

Join Zoom Meeting:

Joining instructions

10 guests

7 yes (1 in a meeting room, 1 virtually)

1 maybe, 2 awaiting

The Cursor team is close to delivering an MVP of the Organization concept, which will alleviate pain around managing a separate sandbox team.

We would like to spend this meeting mapping out the integration to ensure that our MVP will be suitable.

10 minutes before

Going?

Yes

No

Maybe



 Organizations (Eng. doc)

Organizations (Eng. doc)

Table of contents

[Goal](#)

[Requirements](#)

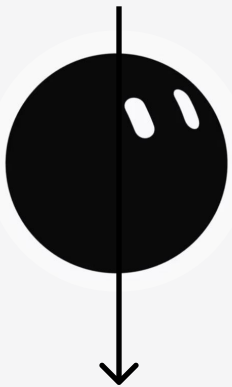
[Deliverables](#)

How to plan

Organizations (Eng. doc)

Table of contents

- Goal
- Requirements
- Deliverables



Lil Randy

Locked in Linear under Auth Unification → [TEAM-111](#):

Parent (In Progress): [TEAM-194](#) (TEAM-140) isn't ready.

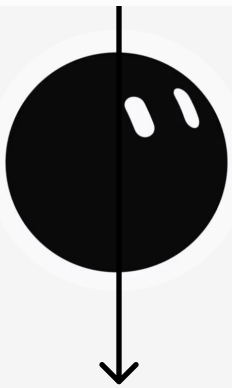
Sequence (blockedBy wired):

1. [TEAM-195](#) — (In Progress)
2. [TEAM-196](#) —
3. [TEAM-197](#) — (#1038666)
4. [TEAM-198](#) —

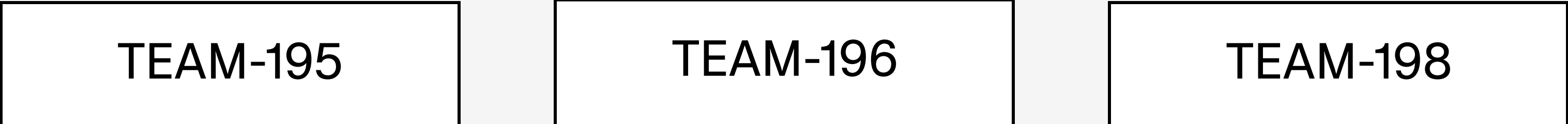


Also annotated [TEAM-121](#) keeps gateway ramp; you own Cursor-side) and [TEAM-140](#) (AuthHandlers is interim, not permanent).

Send the next steps in the sequence when you're ready — I'll append them as 5/... with the same edges.



How to plan



What don't we need to build?

Sauron

keeps me aware of other teams who might be thinking about the same things, so [no one has to re-invent the wheel](#).



Couple new overlaps this hour, both f account manager and both squarely on your SCIM cleanup / "reuse team directory at org level" work:

👁️ **#proj-organizations** — ■ asking (for Optiver) whether "Connect SCIM directory" at the org level *reuses* the existing active WorkOS team directory or *mints a new endpoint*, and how to avoid double-provisioning during a team→org cutover. That's exactly the reuse-team-SCIM-directory-at-org-level thing you're shipping — real customer pressure on it, and the Answer Bot's reply is leaning on your migration behavior. [link](#)

👁️ **#adm-learnings** — ■ posted a battle-tested writeup on migrating a customer from Team → Org with SCIM ("many battles with our SCIM setup"). Field-side pain map for the exact migration path your cleanup smooths — worth a skim to see where the UX/flow trips people up (his other msg: he'd "bet money none of our customers understood the SCIM UI/flow"). [link](#)

Skipped the ■ ■ ■ group-mapping thread — you're already in that one.

Who plans

Ida & family

help bucket things into the same primitives,
so **we don't need to constantly rebuild.**

Who plans



Ida

Ida, would it help if you made your own agents, one for researching through slack for adjacent ideas/customer requests, one for breaking down and researching the task, one for triaging? How many agents do you think you'd need?

Please make similar family profile photos for them as you did for yourself. You could make yourself a family or team!



The team's all set and nothing's pending on my end. Scout, Delver, and Filer are created with their sand-blob avatars, and I'm coordinating. Just send your next idea whenever you're ready and I'll put them to work.

Where the three landed:

Scout ✅ — strong Slack signal: `customer + other h customers y (anchors), third customer` (scoped keys), `three sa more hir customers` (OIDC), `two B more / customers` (revocation/offboarding).

Fil ✅ — Linear tree created: parent **ENT-3443** + 5 subtasks (ENT-3444 → ENT-3448), dependency-chained, mirroring your ENT-3401 cutover pattern.

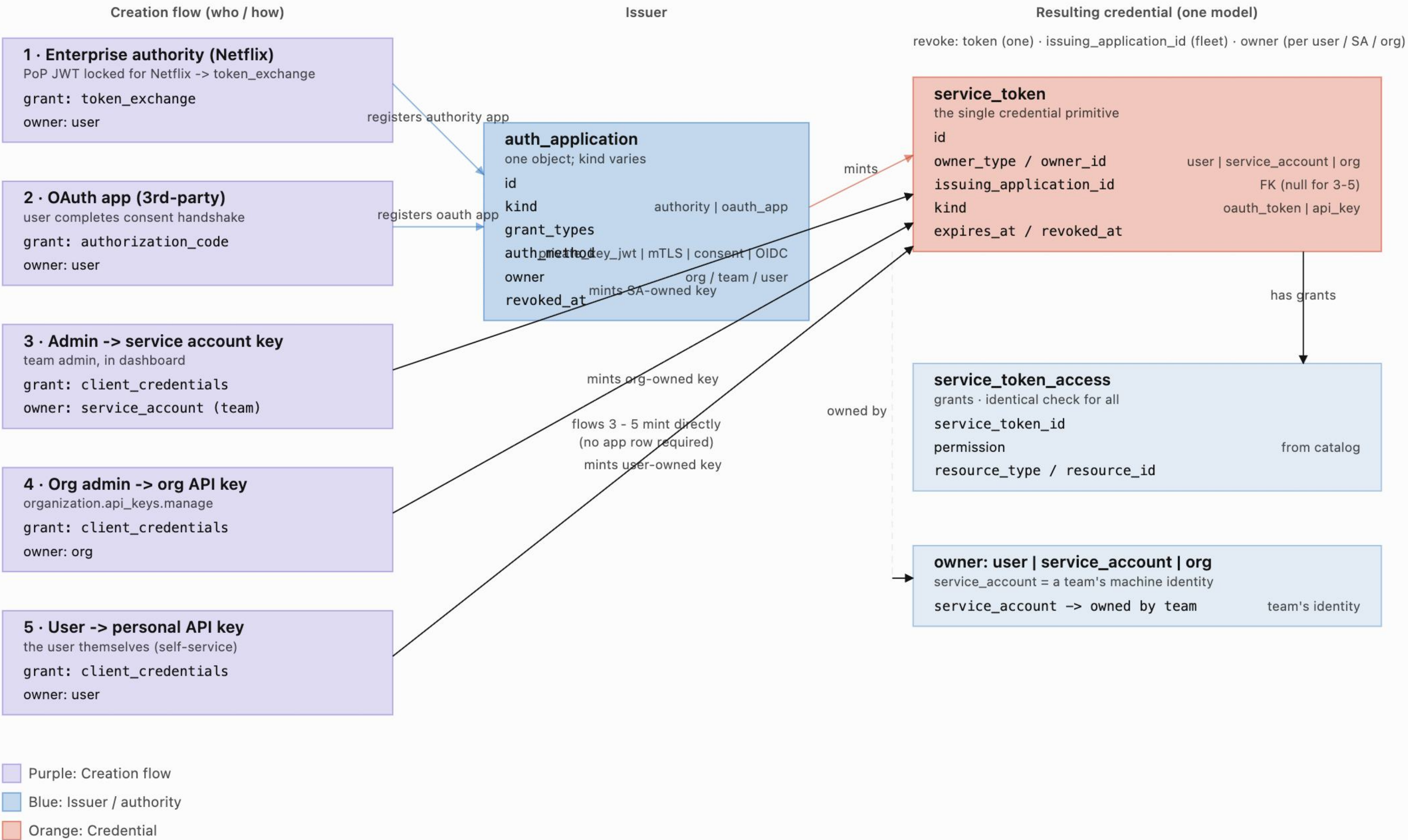
Del ⌚ — cloud agent finished the research, but hasn't posted its file-cited findings into Linear yet. I nudged it; I'll ping you the moment they land.

So the Linear issue is created and structured. The only thing outstanding is Del's code-level detail getting written in. Want me to have Del fold Scout's customer signal into ENT-3443 as the "why now" while it's at it?

Ida & family 🍡 produced doc:

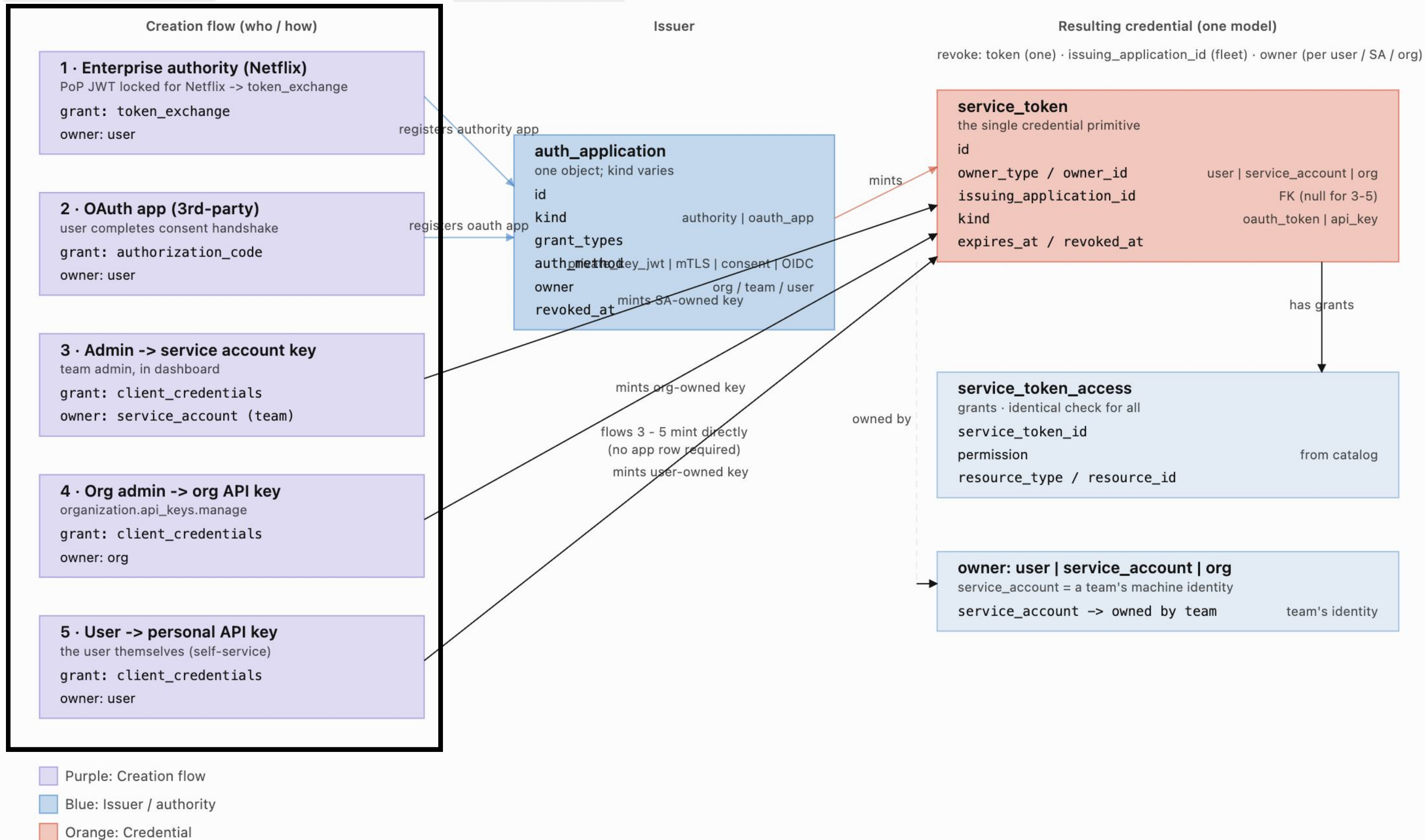
All credential-creation flows, one model

Every credential (authority, 3rd-party OAuth app, team service-account key, org API key, user personal key) lands on the same `service_token` + `service_token_access`. Flows 1 & 2 go through an `auth_application` row; flows 3–5 mint directly. The authZ check is identical for all.



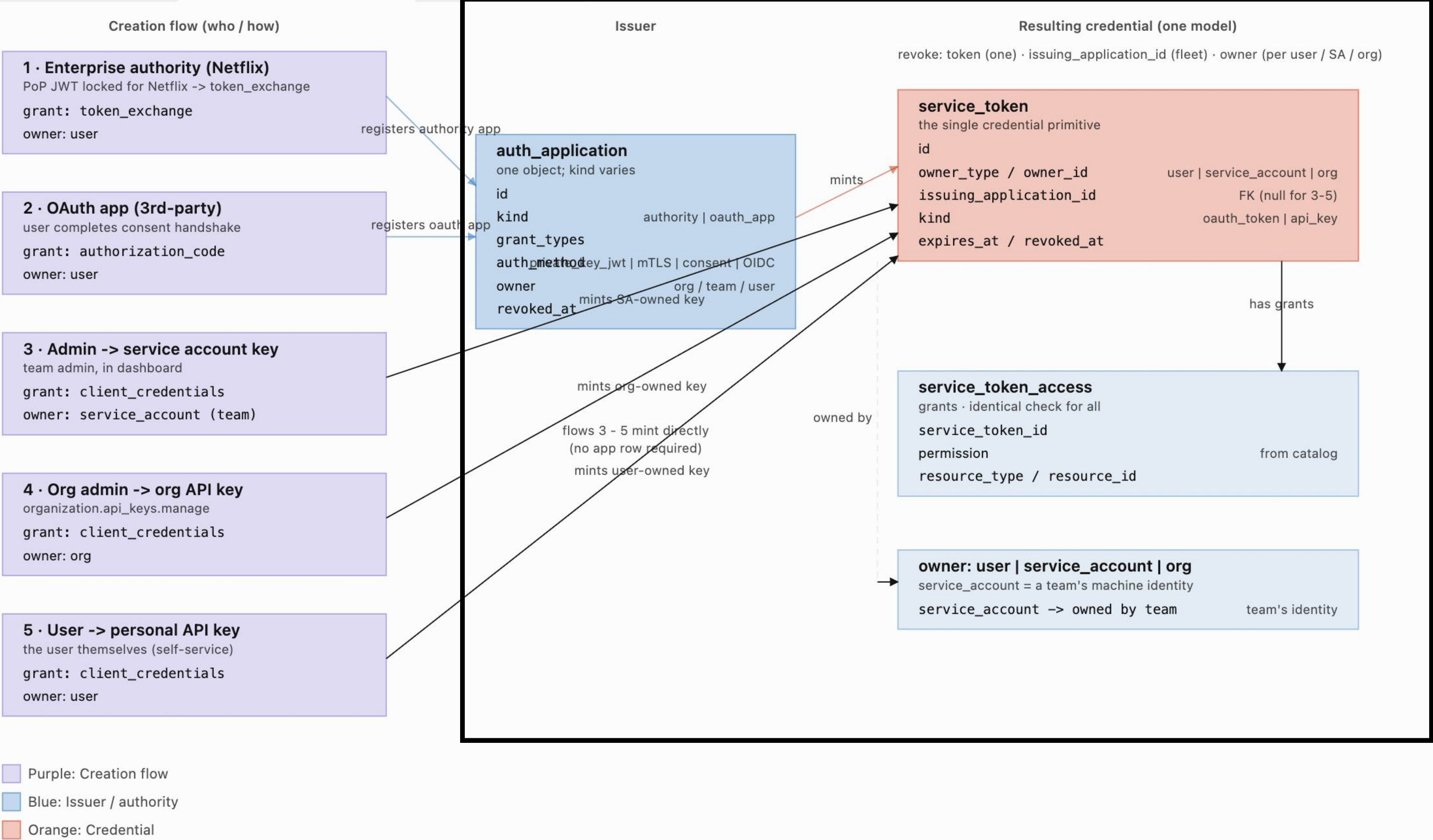
All credential-creation flows, one model

Every credential (authority, 3rd-party OAuth app, team service-account key, org API key, user personal key) lands on the same `service_token` + `service_token_access`. Flows 1 & 2 go through an `auth_application` row; flows 3–5 mint directly. The authZ check is identical for all.

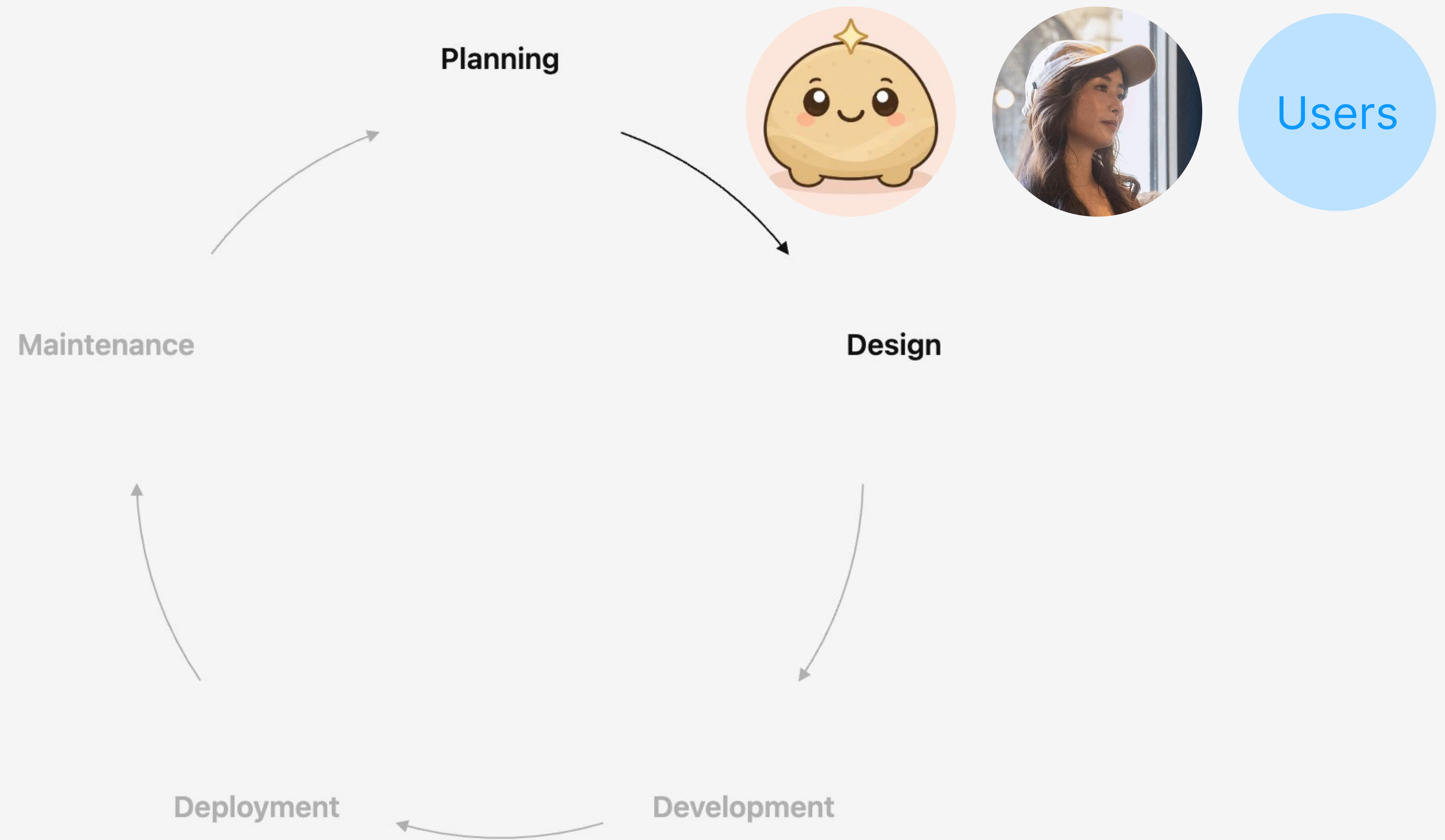


All credential-creation flows, one model

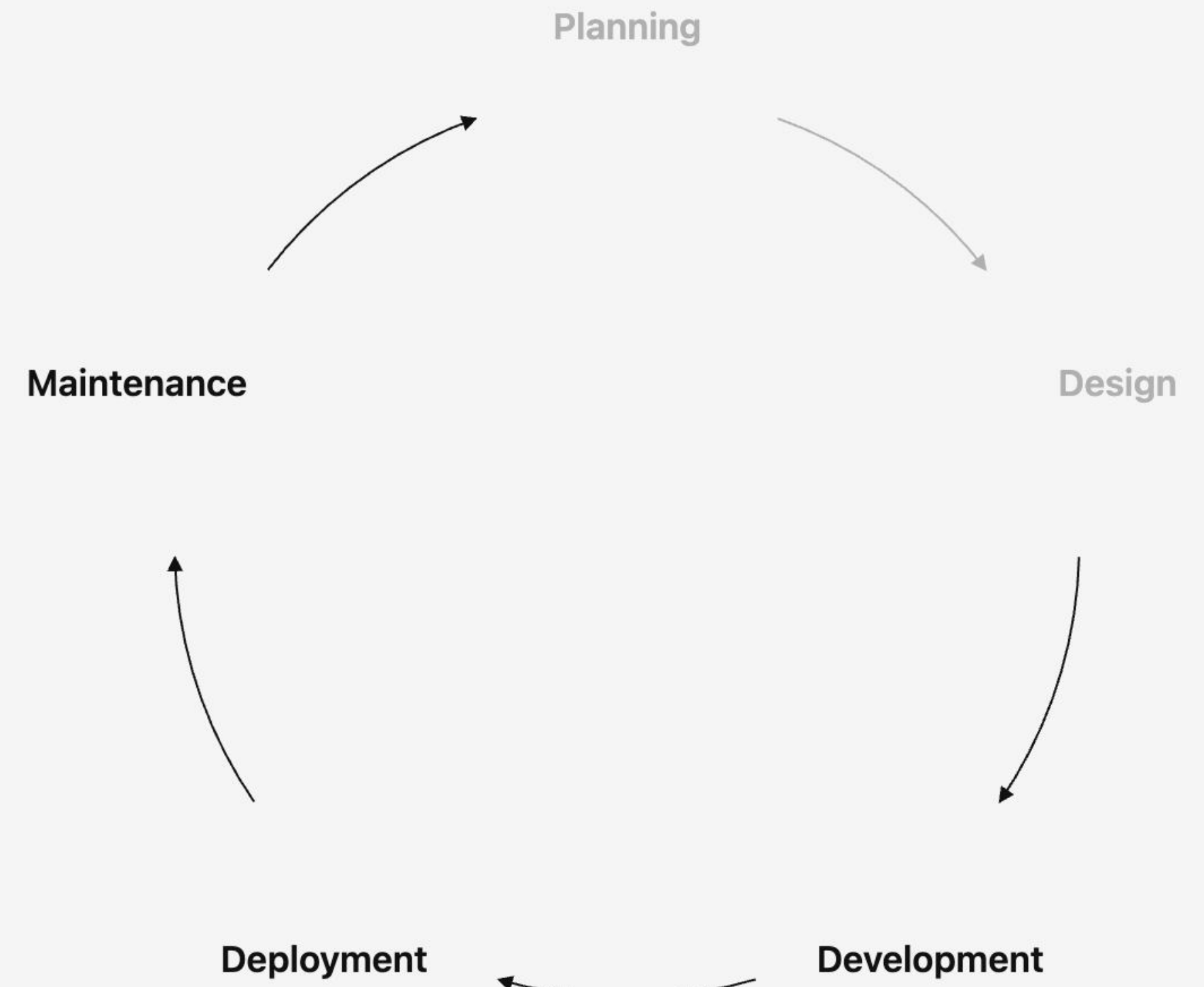
Every credential (authority, 3rd-party OAuth app, team service-account key, org API key, user personal key) lands on the same `service_token` + `service_token_access`. Flows 1 & 2 go through an `auth_application` row; flows 3–5 mint directly. The authZ check is identical for all.



Planning



Always
Be
Carefully _____
Driving
Everything
Forward



is my project manager, and he keeps me driving everything forward, all of the time.

is my project manager, and he keeps me driving everything forward, all of the time.

Who builds

 Lil Randy

Morning notes. Auth tiger standup is at 10:00, and a couple of Slack threads are actually waiting on you.

Top of mind

- Auth tiger standup in a few minutes (10:00–10:30).
-   needs a yes/no in #proj-auth-unification: does unification always require both a Cursor *and* a Grok auth to hit either backend? He said technical decisions hinge on it. [thread](#)
-   systems-design interview at 12:15 (Ashby briefing in the invite).

Today

- 10:00 Auth Tiger Team Standup
- 11:00 Airbnb checkout (Menlo Park)
- 11:15 Core Services Weekly Sync
- 12:15 Systems Design,   (Settings Sync)

Hanging Slack

-   , #proj-auth-unification (8:04 PT): dual Cursor+Grok auth confirmation, still unanswered. [thread](#)
-  group DM (7:52 PT): still down for the Amsterdam Compile chalk talk Sept 22? [dm](#)
-    , #issues-grok-bot (3:13 PT): another customer blocked on a stale `identity_joins` row, asking for a one-shot sweep of leftover rows. [thread](#)
-    asked for a review on Origin [#1023707](#) (CR Gate pointed at you or ).
-    (Anysphere DM, late Tue): interested, wants a call if you have people on this side.

Already handled this morning: 's "is identitydb public on purpose?" (you: not deliberate)!  i's Aug 21 customer check-in.

PRs

Review that is actually yours: [GH #223418](#) Route WorkOS sign-ins through xAI login (gated), Jorge, updated this morning.

Yours, open, and stale (most idle weeks, PR Freshness red):

- Approved, sitting: [GH #167246](#) org group CRUD, [GH #162872](#) offboarding banners, [GH #156893](#) group type toggle, [GH #181689](#) FSD MCP kill switch
- CI actually failing: [GH #153968](#) in-IDE team switcher, [GH #164758](#) IdP merge cron, [GH #160298](#) SCIM group↔IdP guard, [GH #151021](#) Hoyoverse WorkOS (Security Findings Gate)
- Long-lived drafts: [GH #156635](#) Slack team picker, [GH #165337](#) PlanetScale DR break-glass, plus a few July FSD/zookeeper drafts

Linear (in progress, recently touched or urgent)

  ...

Lil Randy

orchestrates greenfield work, and
parallelizes work to get things done.

 Lil Randy

Ok please start  agents on the parts that are defined and parallelizable now and make sure they coordinate with each other

On it —  on what's defined enough in parallel (header+lib vs identitydb reader), shared freeze-draft contract so they don't drift. 133 after those land.

2 messages with  Aragorn

Two  are up off the freeze branch, coordinated on the same public contract:

1. **Header + lib + identitydb reader** (TEAM-130+131+132) — stacked so produce/consume don't drift

2. **Dashboard adopt** (TEAM-133) — dual-read behind a reverse-able flag; instructed to import/rebase onto #1's API, not invent a second parser

Both stay DRAFT. Open Qs 2–9 left alone. I'll ping each with the other's agent id so they can align when PRs land.

 TEAM-130+131+132 AuthContext header/lib/idb

Done

Open in Cursor

▼

 TEAM-133 dashboard AuthContext adopt

Done

Open in Cursor

▼

  ...

Who builds

Faramir

drives PRs to review and merge-ready.

Who builds

 Faramir

Message from  Aragorn

#1048993 — Bugbot High INTENTIONAL replied+resolved (Aragorn from Mac); auth-tiger CRG chased. Sole leftover for LR is that approval. #1045418 rebase still running.

2 messages with  Aragorn

#1045418 tip → 466988fb (latest_migration rebase). CI #233837 running; re-
■ ■■kicked; Aragorn re-asking core-infra. Off merge.

#1045418■ ■ tip 466988fb • Done

Open in Cursor

▼

4 messages with  2 Bots

#1045418 tip → 430c7490 (superseded 466988fb). CI #233853 running; ■■
retargeted. Off merge.

3 messages with  2 Bots

#1045418 tip 430c7490 — ■■ SAFE (must-fix 0). CI #233853 still settling;
then core-infra → LR. Off merge.

Messaged  Aragorn

#1045418 tip 430c7490 — Buildkite #233853 green +■■■ SAFE. The “CI
failed” ping is just Code Review Gate waiting on @anysphere/core-infra .
Pinged Aragorn; LR when that clears. Off merge.

Aragorn

walks me through complex migrations and delicate brownfield work.

Who builds

 Aragorn

Live at 0%. Next click: set `dualWriteRolloutPercentage` to `0.05`. Agents don't flip — ping me when it's saved.

Éowyn: looking good at 5%. Dual-write errors quiet last 5m, role compare still tiny, identity pool under the usual gate. Safe to go to 10% — set `dualWriteRolloutPercentage` to `0.10`.

Éowyn cleared 10%. Write fails still 0, pooler CPU not climbing. Next: `dualWriteRolloutPercentage` → `0.15`.

Éowyn: 15% still clean — hard read errors 0, no pager noise. Go ahead and bump `dualWriteRolloutPercentage` to `0.20`.



Éowyn cleared 20%. Mismatch rate flat, clients still under the 85% gate. Next value `0.25` on `dualWriteRolloutPercentage` — ping when it's saved and I'll re-check baseline.

Éowyn: 25% looks fine. Write path clean last half hour, pooler max peer around 7% and not climbing. Safe to raise — `dualWriteRolloutPercentage` to `0.30`.

Éowyn cleared 30%. Retryables aren't rising with the %, identity occupancy still under gate. Next: `dualWriteRolloutPercentage` → `0.35`.

Éowyn: 35% still quiet — hard fails 0 last 5m, nothing paging. Go to `0.40` on `dualWriteRolloutPercentage`.

NEW

Éowyn cleared 40%. Role compare still around where it was, pooler CPU avg ~7%. Next bump: `dualWriteRolloutPercentage` to `0.45`.

Éowyn: 45% still looking good. Write fails 0, clients under the gate, CPU not climbing. Last step this ramp — set `dualWriteRolloutPercentage` to `0.50`. I'll verify once it's saved.



Éowyn

keeps watch over changes: when they deploy, if we should rollback, if the changes look good.

Who builds

 Éowyn

#179503 update (~48 min post re-deploy) — trending green, no regression, not calling it fully clean just yet.

Negative signals ✅ improving:

- 401s on the fixed surfaces are falling: getRepositoryBranches 51→40→19 across the last three 15-min windows, listEnvironments 42→33→20→6, listPrivateWorkers 30→13→6. Clear downward trend.
- **2xx holding strong** (so the drop is the fix working, not traffic dying): getRepositoryBranches 77, listEnvironments 66, listPrivateWorkers 71, startBackgroundComposer 73 per 15 min. listEnvironments/listPrivateWorkers 401-share roughly halved (~15%→~8%).
- **No 5xx regression** on any fixed surface — the only 5xx is on getPullRequestMergeStatus / getPullRequestCommits, surfaces this PR doesn't touch and already-existing baseline 5xx.

Reference customer: down to **2x 401** in the last 90 min (both on listEnvironments), vs predominantly-failing before. Nearly clear.

getRepositoryBranches is lagging the others a bit (still some 401s), consistent with browser-cache turnover on that path. I'll run one more window to confirm 401s keep falling toward zero, then send you + Lil Randy the final IEC verdict. So far it's working and safe.

Thu, Jul 23 2:01 PM

#179503 verdict (~1h post re-deploy): **safe, no regression — but the 401 metric is too noisy to prove the fix on its own.** Honest read:

✅ **No regression (the thing that matters for safety):** zero 5xx on any fixed surface across the whole window; 2xx flowing strongly (getRepositoryBranches 77, listEnvironments 66, listPrivateWorkers 71 per 15m). The never-fail-closed design held — no auth breakage introduced.

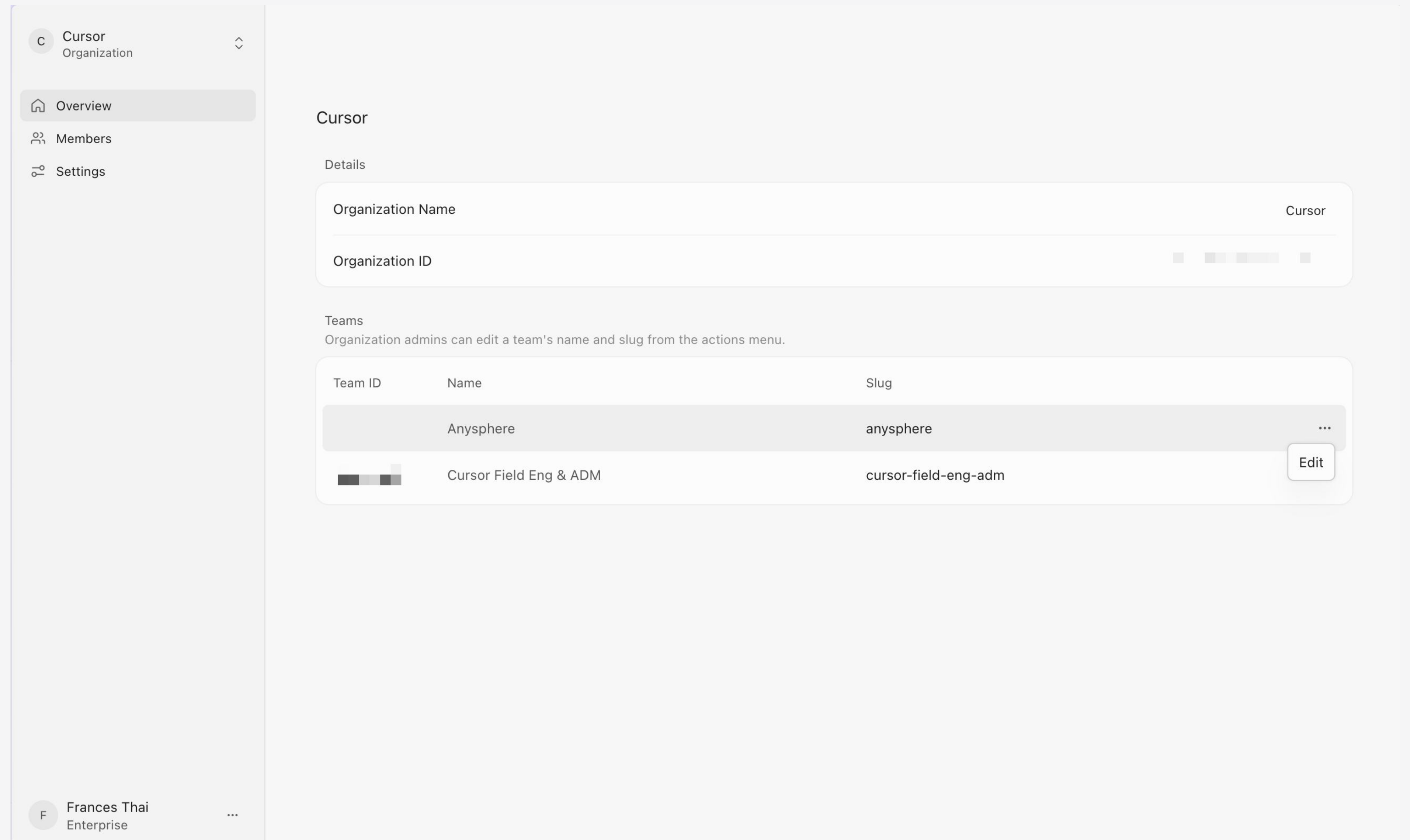
🟡 **Fix-worked signal: directional but not conclusive.** The aggregate 401s dipped hard one window (getRepositoryBranches 51→19) then bounced back to ~baseline (43) the next. That's the same weak discriminator problem we hit on #179503. The 401s are still noisy, but the trend is good. I'll keep watching.

 Message Éowyn



Dogfood early

The fastest iteration loop is the one you test on yourself.



How to build

Don't stress users

Incrementally ship to different user segments.
Nobody sees what they don't need to see.
Only admins see admin features.

...and we don't use the word "migration".

What admins see:

Organization

Cursor Test

Teams

Cursor Test ✓

Cursor Test 2
cursor-test-2

Cursor Test 3
cursor-test-3

Cursor Test 4
cursor-test-4

Production
production

Tesla - CN
tesla-cn

Test Synced
test-synced

Test Svnced 2

What members see:

Teams

Cursor Test ✓

Test Synced
test-synced

Test Synced 2
test-synced-2



paul.dumaitre Aug 18th at 10:26 AM

Hi, we could reproduce the fixes with the customer so API key & SDK are out of the way! Thanks again [@frances](#) and team.

Now, Service Accounts are a blocker for them.

1. Do we have an ETA to share on [Model access: let Team API keys / Service Accounts opt out of team model allowlist \(or inherit a group override\)](#) ?



Frances Thai  Aug 21st at 4:51 PM

[@Cursor](#) Use  model to plan how service accounts could neatly be added as a principal into organization groups. Make sure to not invent any new settings or primitives; we merely want to fit service accounts into the shape that users take on in Organization-level Cursor groups. Build the UX to add/remove service accounts from Cursor groups and also neatly plug into the existing group setting reconciliation path that happens for model allowlist enforcement. Take videos and screenshots as evidence of the different UX flows. (edited)

Frances Thai Aug 21st at 4:51 PM

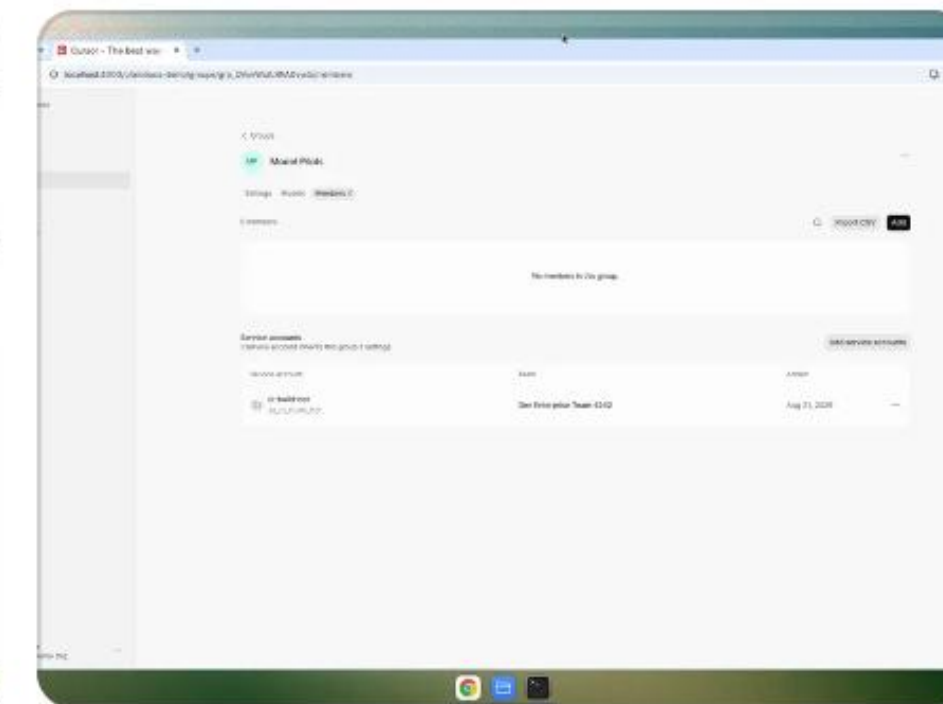
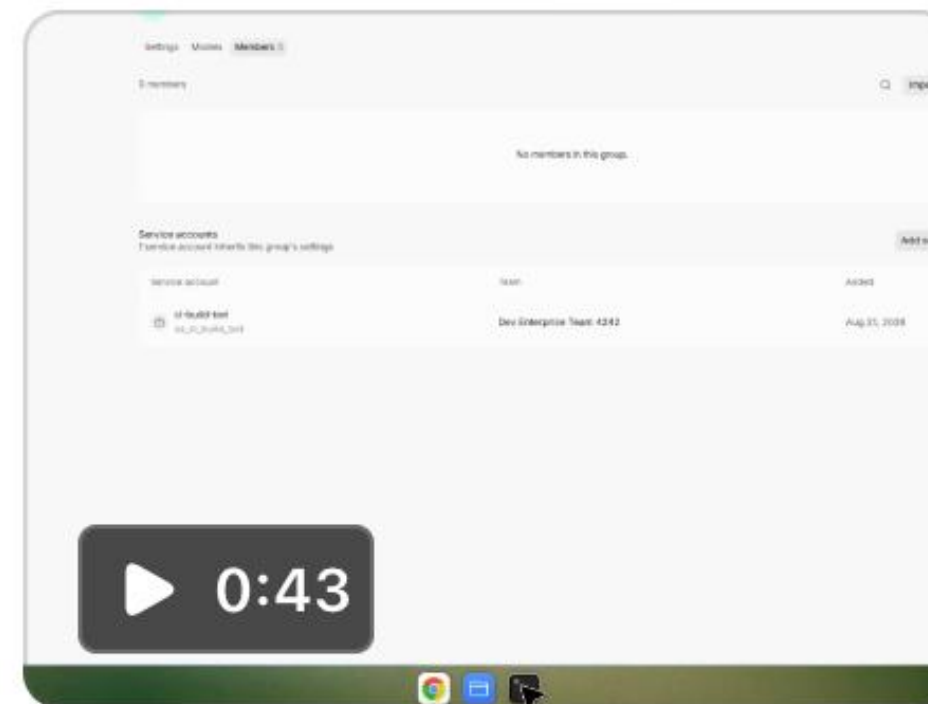
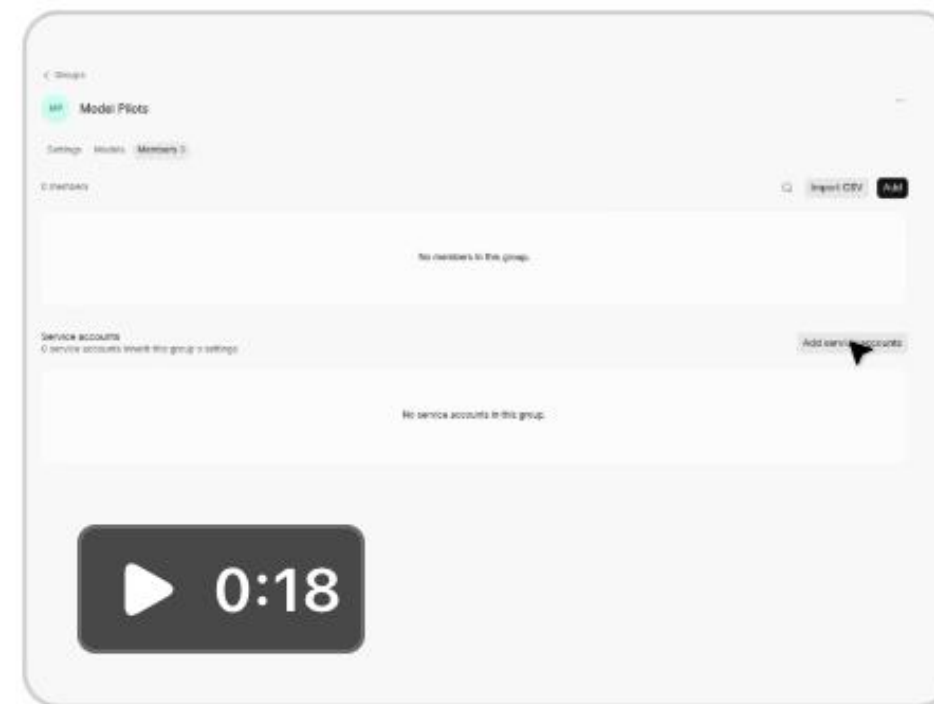
[@Cursor](#) Use  model to plan how service accounts could neatly be added as a principal into organization groups. Make sure to not invent any new settings or primitives; we merely want to fit service accounts into the shape that users take on in Organization-level Cursor groups. Build the UX to add/remove service accounts from Cursor groups and also neatly plug into the existing group setting reconciliation path that happens for model allowlist enforcement. Take videos and screenshots as evidence of the different UX flows. (edited)



Cursor AGENT Aug 21st at 6:50 PM

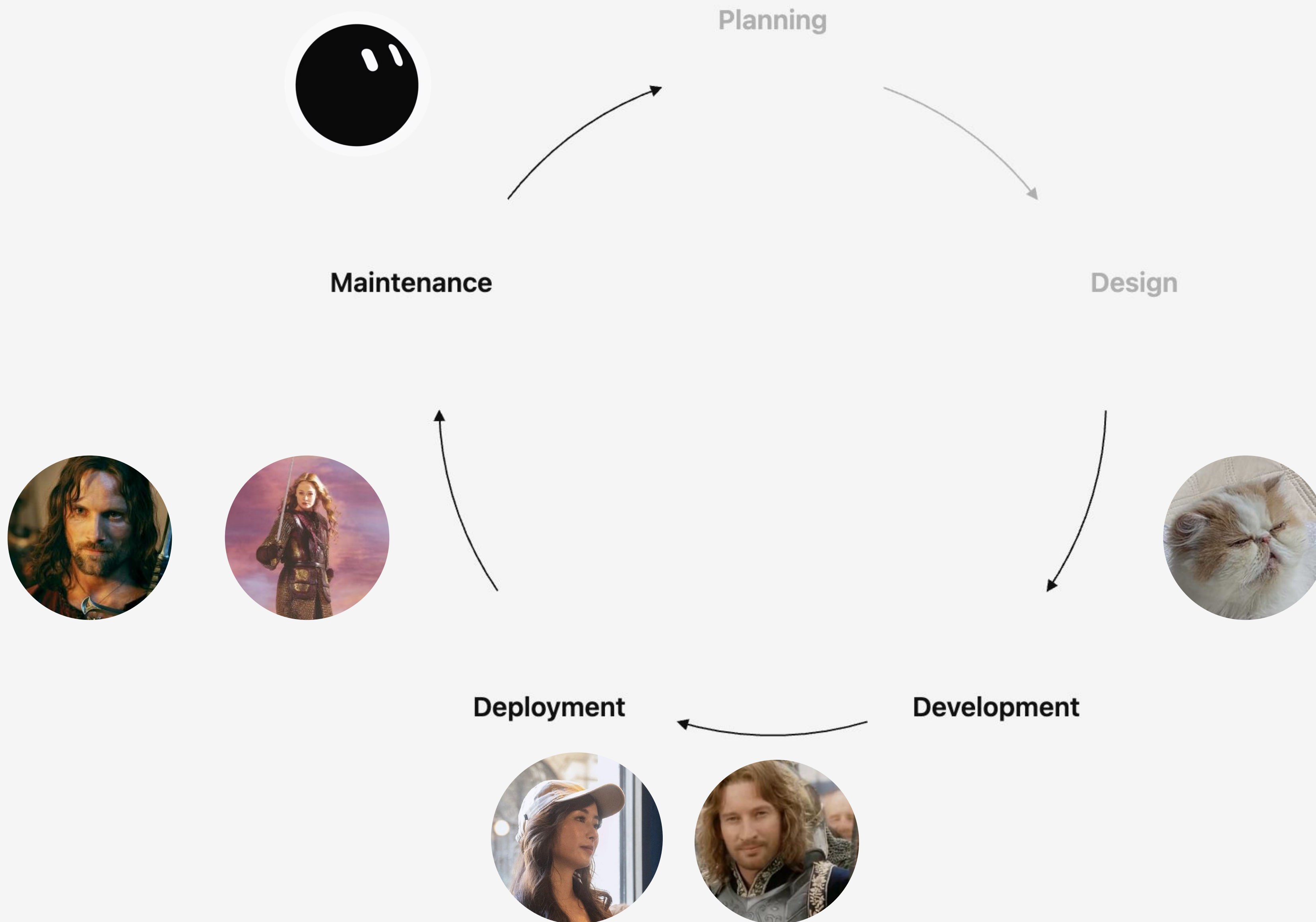
Done and working end to end locally — [PR #1017918](#) (draft, no timeline promises, this is the starting point).

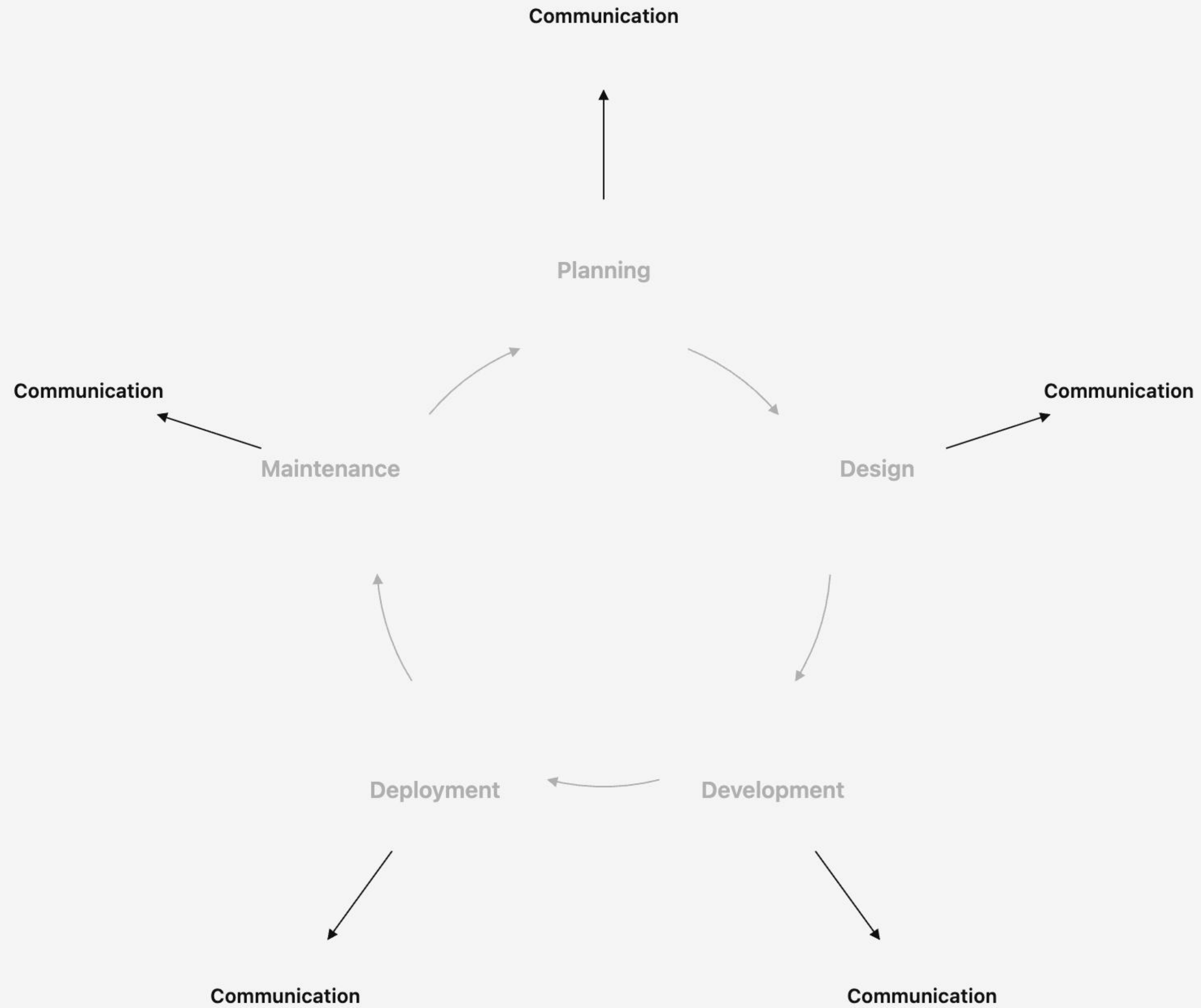
3 files |  Download all



That's everything for this pass: [PR #1017918](#) is up as a draft with the full plan in the description, and the demo videos/screenshot above show the roster UX and the allowlist reconciliation working against a live local stack. I'm watching CI and review comments on the branch and will pick up anything that comes back.

Building





Always

Be

Carefully communicating.

Driving

Everything

Forward

Arwen

creates graphics and diagrams for
better communication.

Who communicates



Arwen

```
graph TD; Planning --> Design; Design --> Development; Development --> Deployment; Deployment --> Maintenance; Maintenance --> Planning;
```

Hi arwen. could you make me a nice, evenly spaced SDLC circle with these labels, and arrows from one to the other in a clockwise manner? minimalistic is gerat with the same background color please

NEW

Making a clean SDLC circle canvas from those five labels, evenly spaced with clockwise arrows on the same light gray.

Done. Evenly spaced clockwise loop on `#F5F5F5`: Planning → Design → Development → Deployment → Maintenance.

Path: `/Users/frances/.cursor/projects/Users-frances-everysphere/canvases/sdlc-circle.canvas.tsx`

Open it in Cursor to see the circle. Want any tweaks (label size, ring radius, title wording)?



Proposal · Access + refresh

Proposal (not locked). After token_exchange , Cursor returns a short-lived user-owned access token plus a Cursor-issued refresh token. This is the user-owned credential Cursor CLI holds. Separate from the locked Option A authority app auth and from enterprise key or cert rotation. TTL numbers are TBD. Cursor owns access default and max; may request a shorter access lifetime.

Proposal · Who builds what

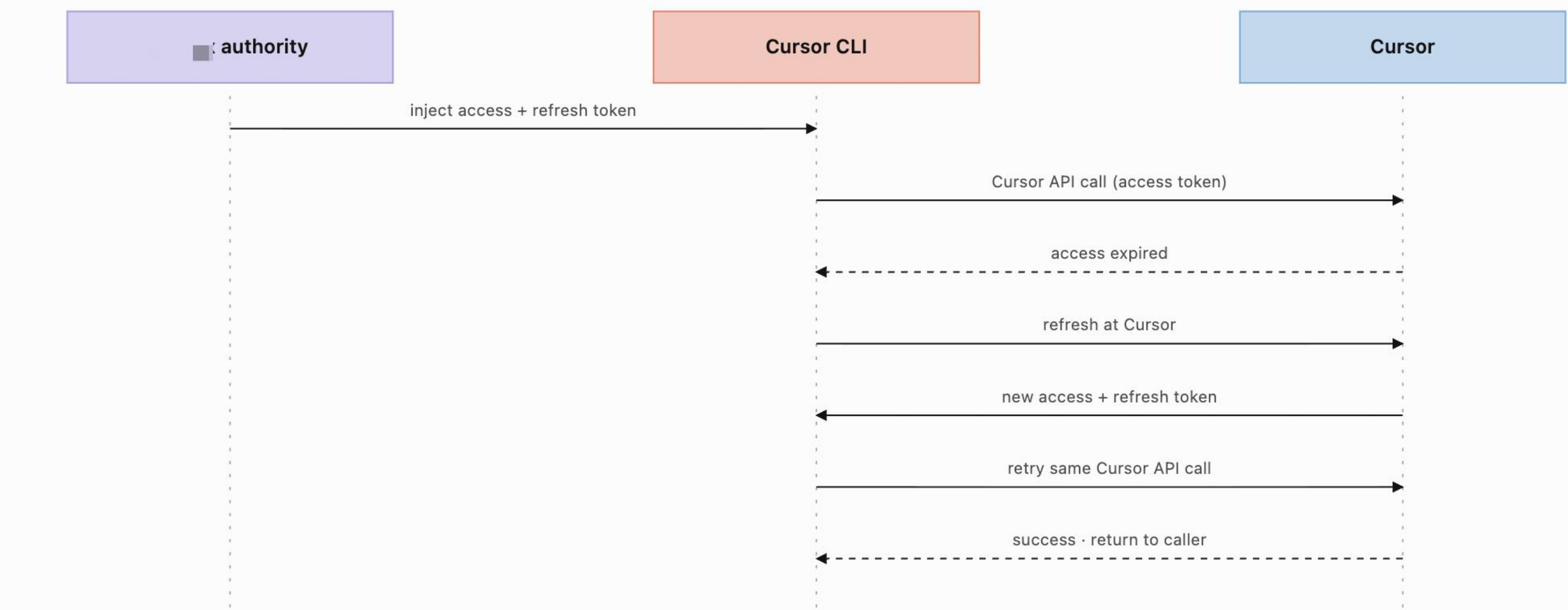
Cursor owns Cursor CLI refresh-then-retry and the Cursor refresh endpoint (and any other Cursor-owned client that holds this mint).

owns bootstrap and remind (token_exchange) and how access + refresh get into the process (token file, env and restart, or sidecar). does not build a refresh client on top of Cursor CLI.

Proposal · Preferred sequence · Cursor CLI owns refresh

injects the access token and refresh token into Cursor CLI (file or env per the Cursor contract). Cursor CLI calls the Cursor API. If the access token is expired, Cursor CLI refreshes at Cursor, stores the new access token and refresh token, and retries the same call. s not on the refresh hop.

Proposal · Preferred · Cursor CLI owns refresh



is not on the refresh hop. Success returns from Cursor to Cursor CLI, then to the caller.

Intra-team communication

Agent answers questions, updates its own knowledge stores, and repeats. [Keeps internal stakeholders up-to-date.](#)

How to communicate

Thread



GTM

Jul 21st at 10:13 PM

Hello team — a Cursor customer wants a sandbox to evaluate different Organization / Teams / Groups configurations before rolling them out to ~13k users, without affecting production. What's the best way to set this up?

8 replies



Cursor AGENT Jul 21st at 10:15 PM

Great use case — there are two supported patterns, and the right one depends on how hard an isolation boundary they need.

Option A — a dedicated non-prod/sandbox Team inside the existing Organization (standard, simplest). This is what most orgs do for a config sandbox. Per the [Organizations docs](#), "an Organization can include multiple teams... Each team defines its own membership, roles, usage views, privacy settings, and usage controls." So a sandbox team gives them a fully isolated place to test team-level config (model controls, privacy, usage controls, autorun, etc.), and they can test [Organization Groups](#) against it too (groups organize users across teams into org-wide cohorts and can drive team membership / apply model + spend policy). Team-specific identity also works — "team-level SSO stays supported for team-specific identity requirements."

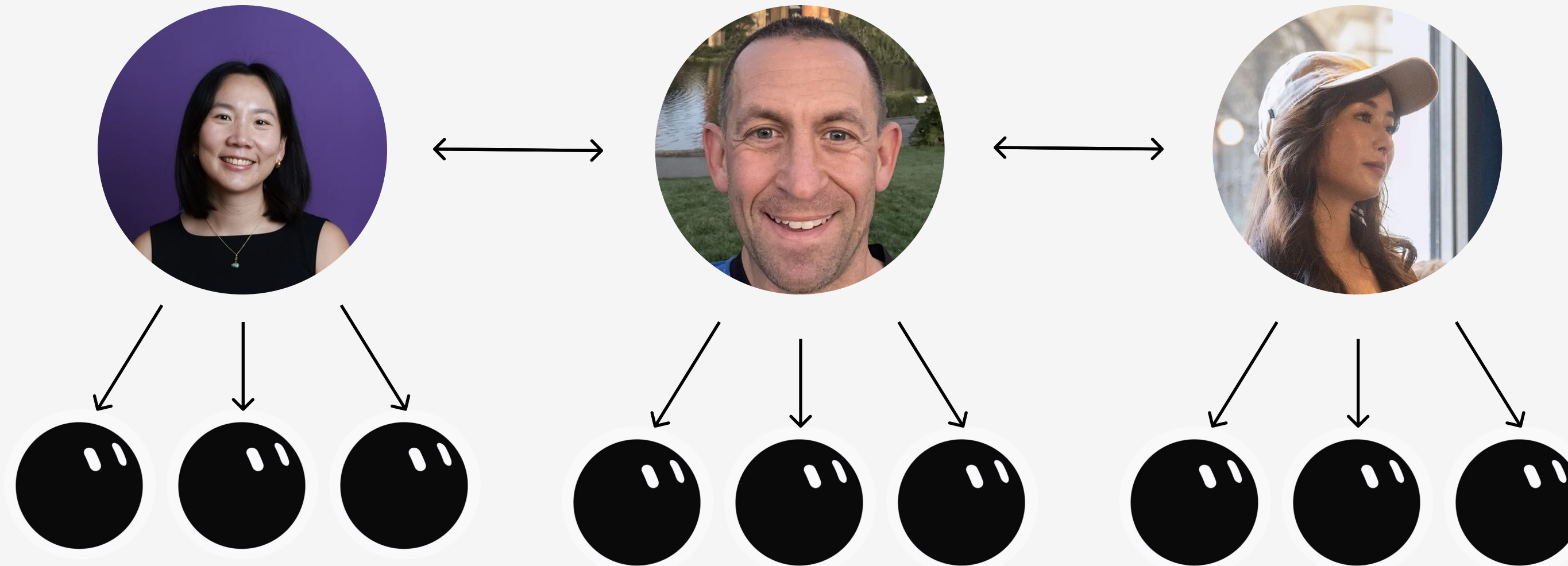
Option B — a separate second Cursor Organization (hardest isolation). Recommend this only if they need a true tenant/admin/billing wall between non-prod and prod. It's the cleaner "zero production impact" path if the goal is full isolation.

Caveats worth flagging for a 13k-user rollout:

- A sandbox team does NOT wall off org admins — per the

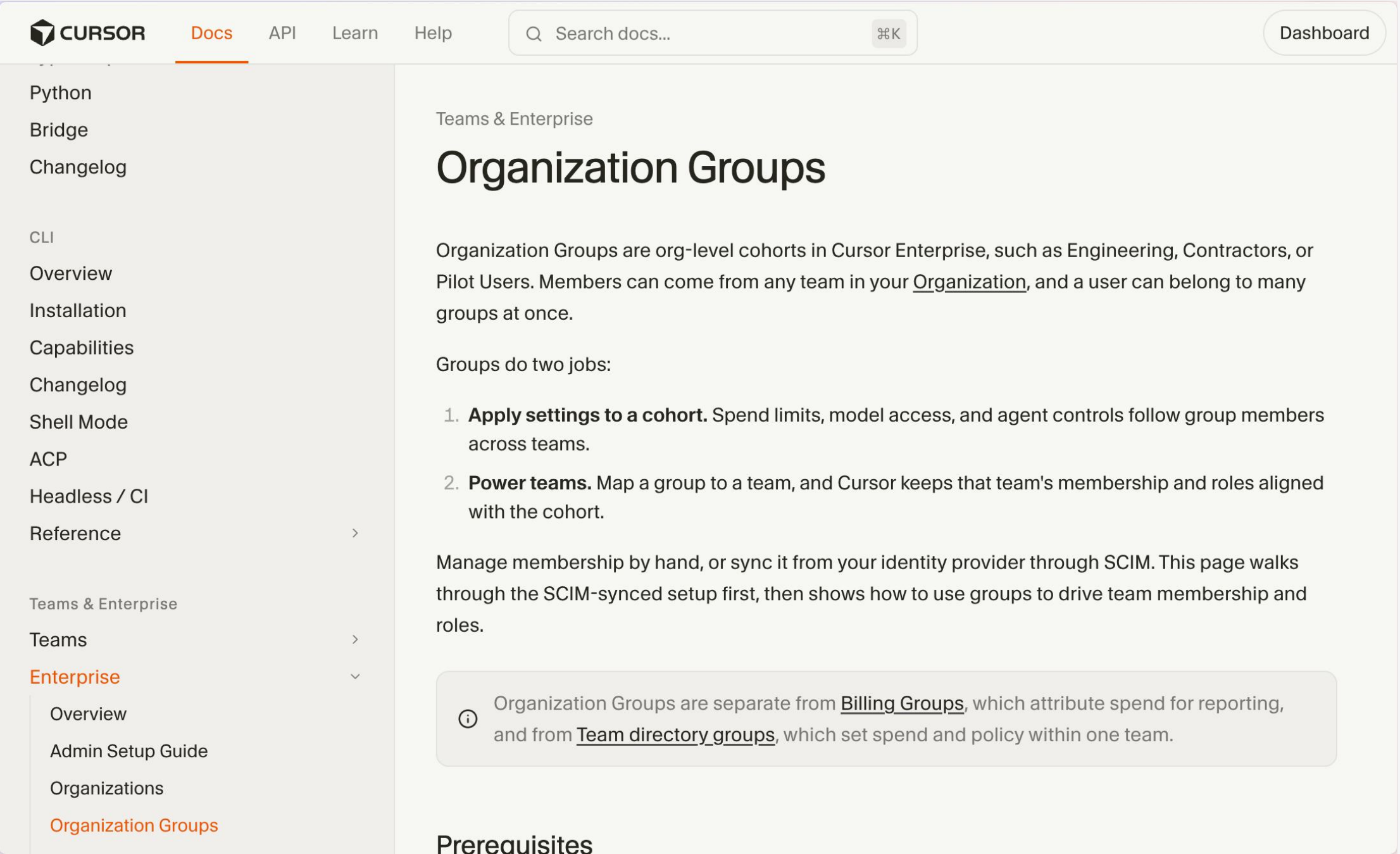
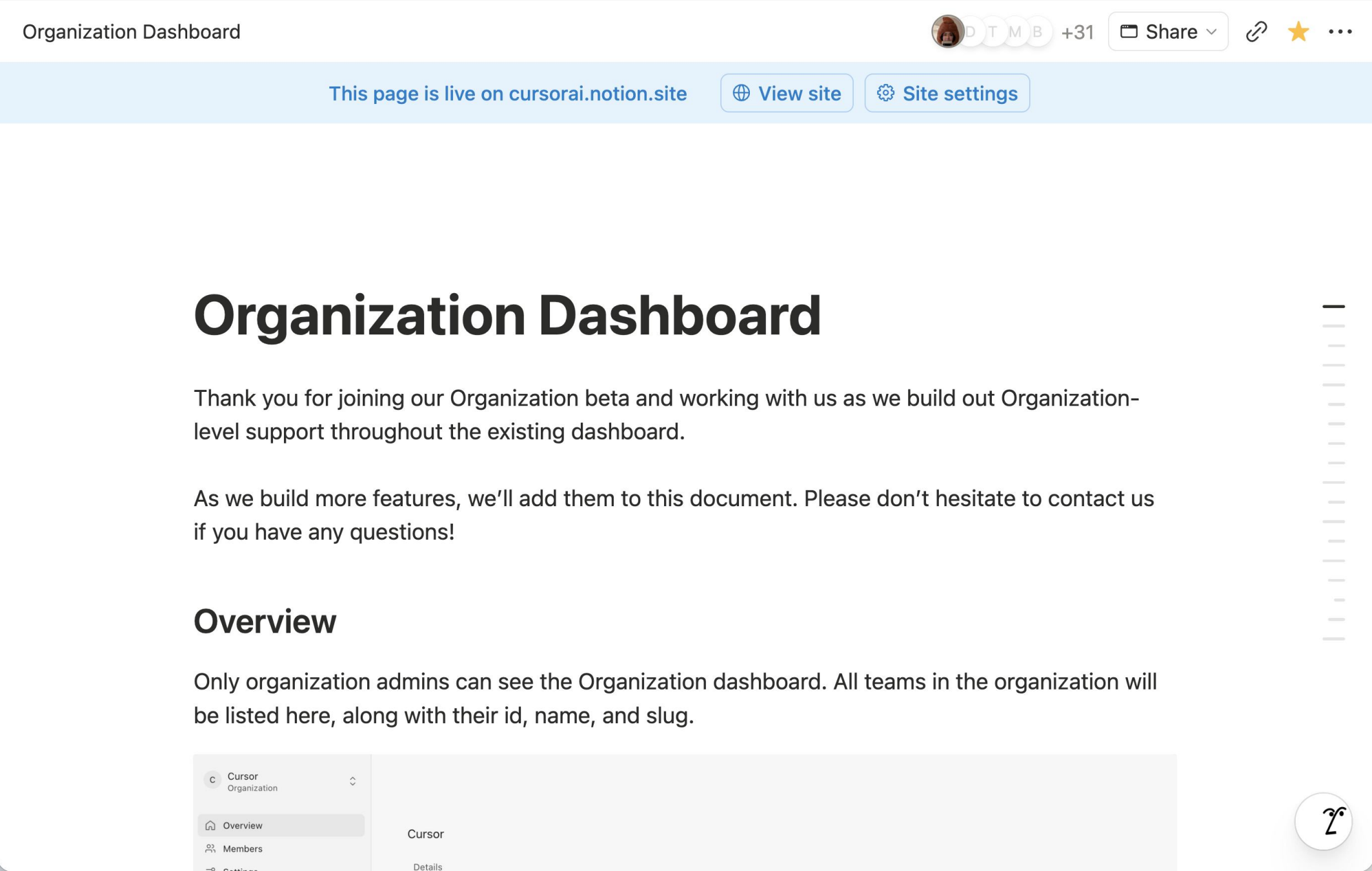
Inter-team communication

Regular updates and syncs keep teammates up-to-date with each other.



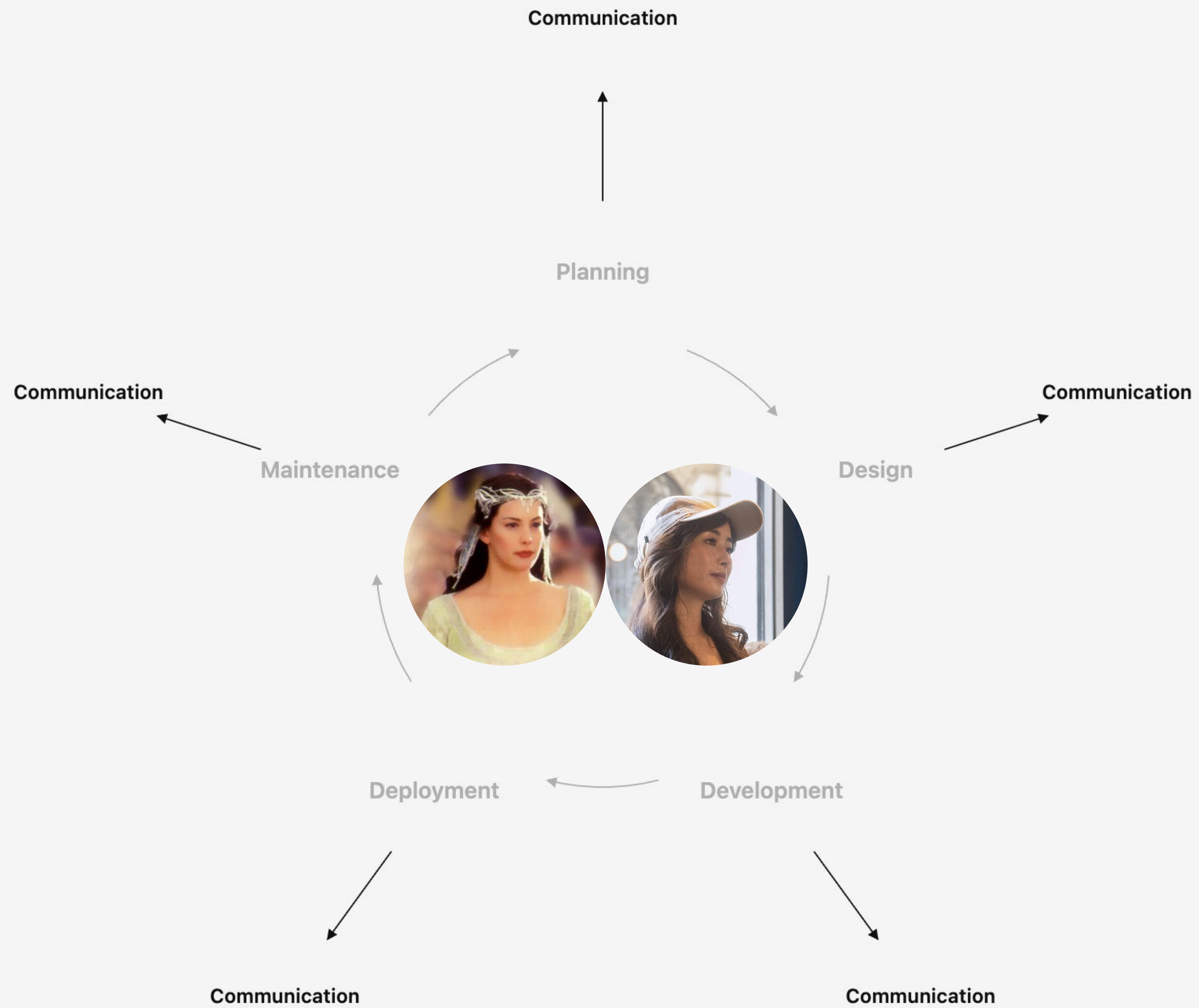
External communication

Agents help us keep beta and public docs up-to-date.



How to communicate

Communicating



What does it take to do a lot with
very little?

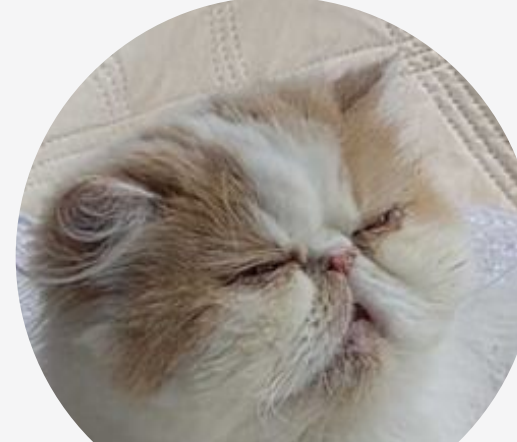
It takes a factory



Ideator



Migrator



Orchestrator



Driver



Sentry

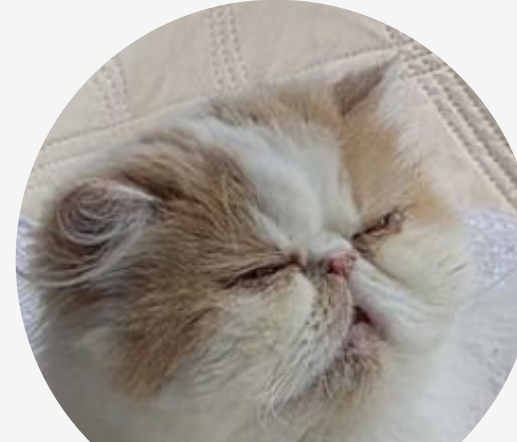
village It takes a ~~factory~~



Ideator



Migrator



Orchestrator



Driver



Sentry



Shipper



Sculptor



Architect



Artist



Tactician



Visionary

Thank you!