

FIELD REPORT · VIEWS FROM INDUSTRY

I let non-engineers vibe-code our production codebase.

Eight months of giving AI coding agents to everyone in a B2B SaaS company. What shipped, what broke, and what I'd do again.

WHO'S TALKING

Hi, I'm Viljami. Distinguished Engineer at epilot.

Paluumuuttaja. Back in Finland since 2025. 15+ years building SaaS.

At epilot since 2021. Five years as Head of Engineering, now hands-on again as Distinguished Engineer.

Author of the Product Engineer Manifesto (productengineer.org). Maintainer of openapi-stack, used by AWS, GitHub and SAP.

The energy transition runs on software utilities don't have.

220+

customers across the
German energy market

135+

people, ~50 of them
product engineers

35+

nationalities,
fully remote

2017

founded in
Cologne

STADTWERKE

City-owned utilities

Think Helen in Helsinki or Tampereen Sähkölaitos.
Germany has hundreds of them, from tiny to some of
the largest utilities in the country.

GRID OPERATORS

The people who run the local grid

Think Caruna or Elenia. Every solar panel, heat pump
and wallbox needs a grid connection, and every
connection is a process.

SUPPLIERS & NEW PLAYERS

Energy brands selling the transition

Solar, heat pumps, wallboxes, dynamic tariffs.
Vattenfall moved its energy-services platform onto
epilot in 2025.

They used to sell one product: kilowatt hours. [epilot is where they sell, serve and connect everything else.](#)

FEBRUARY 2026

I gave every PM, designer and customer success person a coding agent and the keys to our codebase.

I expected a few translation fixes.

Everything else is evidence.

• 01 • THE GAME CHANGER

AI agents as coworkers. In Slack.

Not a tool you open sometimes. A teammate you tag in a thread, like anyone else, and get a pull request back.

• 02 • THE MISSED OPPORTUNITY

Let non-engineers change the software.

If your PMs, designers and customer success can't ship code, you're leaving the people closest to the customer on the bench.

We made the right bets **before AI was a thing.**

ARCHITECTURE

API-first

The product was built from day one to be driven by machines. Agents just happen to be machines.

INFRASTRUCTURE

100% serverless

No servers, no clusters, no ops team. ~300 repos, deployed to production 150+ times a week.

TALENT

Product engineers

Engineers who care about the customer and the business, not just the code. No specialists to hand things off to.

ORG

Small and flat

~50 engineers. No engineering managers. No platform team. Everyone owns what they ship.

Each one was a bet on speed and ownership. **Every one of them turned out to be a bet AI loves.**

ACT 01

We hired a new coworker. **It lives in Slack.**

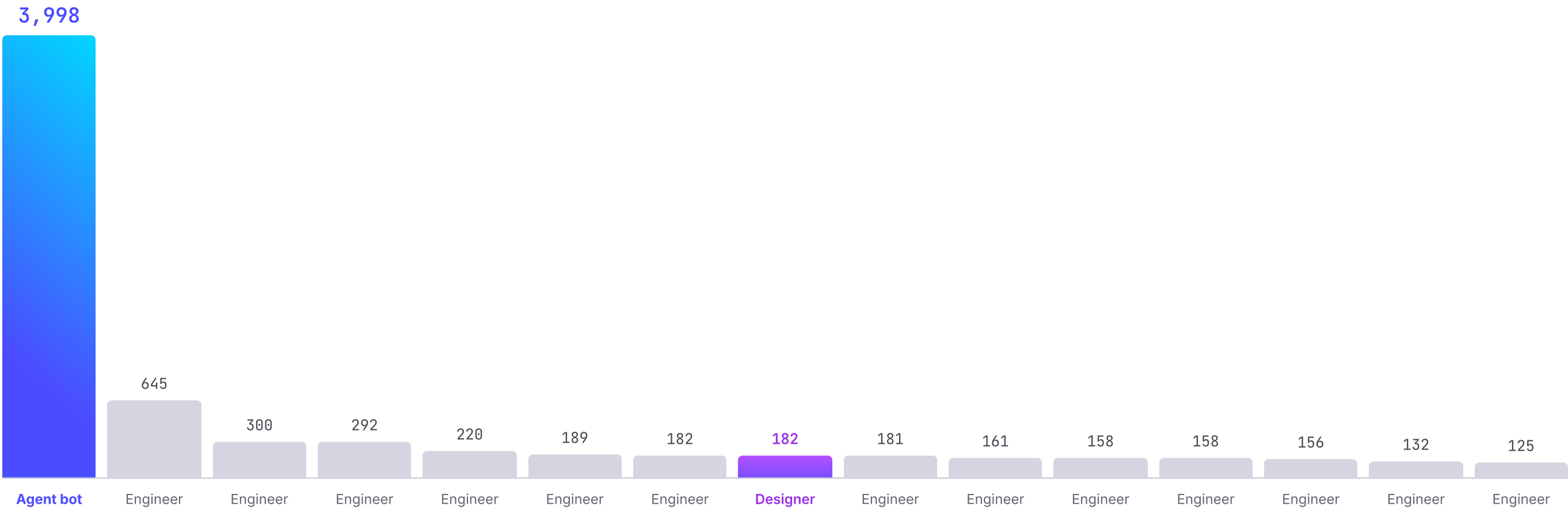
Our squads each have a Slack channel. In February the agents got one too. Then we opened it to everyone in the company.

Tag it like a colleague. **Get a pull request back.**

01 · ASK @mention in a thread In any channel. Plain language. The thread is the spec.	02 · WORK Cloud sandbox A copy of all ~300 repos plus our docs, conventions and design system.	03 · SHOW PR + preview link Posted back in the thread. Click it, see the change, test it.	04 · SHIP Engineer reviews Same review bar as any other change. Then CI ships it.
--	---	--	--

Within a week it stopped being a tool. **Then it became the most active member of the team.**

It became the **most active member of the team.**



3,998 PRs from the agent bot: **42% of all PRs, 6× our most active engineer, more than our top 22 humans combined.**

AI needs your context. **Not a better model.**

- WITHOUT IT

Plausible garbage

- Hallucinates your API patterns
- Picks the wrong library
- Ignores your conventions and naming
- A non-engineer can't tell it's guessing

- WITH IT

Code that looks like your engineers wrote it

- Knows the patterns, practices and why they exist
- Reads five years of RFCs before changing anything
- Uses the design system, in German and English
- Vercel's eval: **100% task success with a docs map, 79% without**

Every repo in one place. Everything we know, indexed.

• 01 • THE BARREL REPO

~300 repos, one sandbox

One repo that mirrors every GitLab and GitHub repo we own. The agent sees the whole system, not one service. Cross-repo changes in one session.

```
# AGENTS.md
THINK SMALL. SEARCH NARROW. READ LESS.
Docs first, code second.
One repo at a time.
```

• 02 • HOW-TO-EVERYTHING

893 docs, 17 domains, one line of context

Conventions, API design, backend patterns, business context, every RFC, framework docs. A compressed index rides along on every turn; the agent opens the file it needs.

```
[epilot Docs Index]|root: ../epilot-docs
|00-general:{code-style-typescript.md,naming-conventions.md,...}
|01-apis:{api-design.md,calling-apis.md,...}
|06-backend:{dynamodb-patterns.md,error-handling.md,...}
|10-frameworks/single-spa:{parcels-api.md,...}
|11-rfcs:{rfcs.md}
```

A junior who takes tickets. A historian who's read every repo.

TICKETS

"@Claude fix EPI-10660"

A pull request comes back. The basic loop.

AUDITS

"Which frontends still use Froala?"

Reads all ~300 repos and answers. Migration planned off the reply.

CROSS-REPO

13 services in one session

One message, sub-agents in parallel, one PR per repo. A day of copy-paste gone.

PRODUCTION

In the alerts channel

Reads the logs, traces the error across services, proposes the fix before anyone opens a dashboard.

Agents worked weekends. **Engineers didn't have to.**

Adoption spreads **by watching.**

It's public. Every request, every PR, every mistake happens in a channel. A PM watches an engineer tag the agent, and the next day the PM does it too.

It's where people already are. Nothing to install. No IDE, no terminal, no new tool. If you can write a Slack message, you can ship a fix.

The thread is the whole story. Request, discussion, PR, preview link and review in one place, readable by anyone.

THEN THE VENDOR CHANGED THE DEAL

Anthropic launched Claude Tag. And priced us out.

Usage restrictions. No proper environment variables. Sessions burning credits on top of the subscriptions we already paid for.

We had a choice. Give up the way we'd learned to work, or keep it. **We kept it.**

SO WE BUILT OUR OWN

Leidang. The coworker, on our own terms.

ANY MODEL

Claude, Codex and more

Agent-agnostic. Opened the door to OpenAI and other frontier models, on subscriptions we already have.

REAL MACHINES

Desktop, terminal, live preview

Cloud sandboxes where the agent can click through the product on its own.

SLACK BUILT IN

"@leidang ..."

Same habit as before. Mention it in a thread, work from where you already are.

ROUTINES

Agents on a schedule

Production alert analysis every 30 minutes. Bug triage. Tech debt sweeps. Ship digests.

Built in-house by one engineer, Suresh. Next: fully sovereign open-source models. [Open source under Apache-2.0, publishing soon.](#)

They work in the background. They always report to Slack.

EVERY 30 MIN

Datadog auto-triage

Production errors and alerts, triaged, trivial fixes as PRs

→ #alerts-production

ON EVERY DEPLOY

Pre-prod risk check

Classifies the risk of what's about to ship

→ #prod-deployments

ON EVERY E2E FAILURE

Self-healing E2E

Reproduces the failure, fixes it in the right repo

→ PR

SEVERAL TIMES A DAY

Bug bash

Picks up new bug tickets, opens PRs for confident fixes

→ #dev-squad-agents

DAILY

Translation fix

Works through open translation tickets

→ #dev-squad-agents

DAILY

Tech debt scan

Takes one backlog item, verifies it, fixes it

→ #dev-squad-agents

DAILY

Support digest

Yesterday's support tickets and the trends behind them

→ #dev-squad-agents

DAILY

Shipped summary

What actually landed on main across every repo

→ #dev-squad-agents

11 workflows, defined as markdown in the repo and code-reviewed like code. Nothing an agent does happens silently.

Auto-triage. Every 30 minutes, forever.

01 · WATCH Checks Datadog Errors, alerts and logs from production, every 30 minutes.	02 · TRIAGE Sub-agents per pattern Groups the noise into patterns, investigates each one across services.	03 · FIX Opens a PR When the fix is trivial. Same safe-PR rules as everyone. An engineer merges.	04 · REPORT Posts to Slack !! critical · ! major · ⓘ minor. "Don't be dramatic."
• IT LEARNS 209 learnings in its own memory file Read before every run, updated after. Changes to its own instructions go through a PR.	• FROM ITS MEMORY, VERBATIM "An MR opened in one window doesn't mean the bug stops recurring — the fix only takes effect once merged and deployed."		

One engineer on call. One week. From Slack.

95 agent sessions launched from the alerts channel	56 fix PRs merged	1 engineer on call
--	---	--

Production support at a scale no on-call rotation could staff. **One engineer with a team of agents.**

ACT 02

Then the rest of the company started shipping.

Same coworker, same Slack channels. PMs, designers and customer success started tagging it too.

They describe the problem. The agent finds the repo.

"@Claude disable the drag and drop of entity table columns for the embedded entity tables"

"@Claude when creating a new family in the Label Builder and the user clicks 'Create family', the drawer should close automatically. Fix that."

"@Claude add the missing German translations for the relation overview in the sidebar"

Product managers, verbatim. No repo names, no stack, no file paths.

If you can't see your change, **you can't own it.**

01 · EVERY PR

A review URL

Click it and you're in the real app with the change applied. No local setup, no terminal. Engineers run the app locally. A PM can't.

02 · EVERY AGENT RUN

Screenshots, posted for you

The agent screenshots its own change and posts it to the Slack thread and into the PR. Reviewers see it before they click anything.

03 · WITH LEIDANG

A live dev sandbox

Some designers didn't want to wait for a PR. They wanted to see the change live while they iterate. So now they can.

Engineers had local dev. **Everyone else needed the feedback loop built for them.**

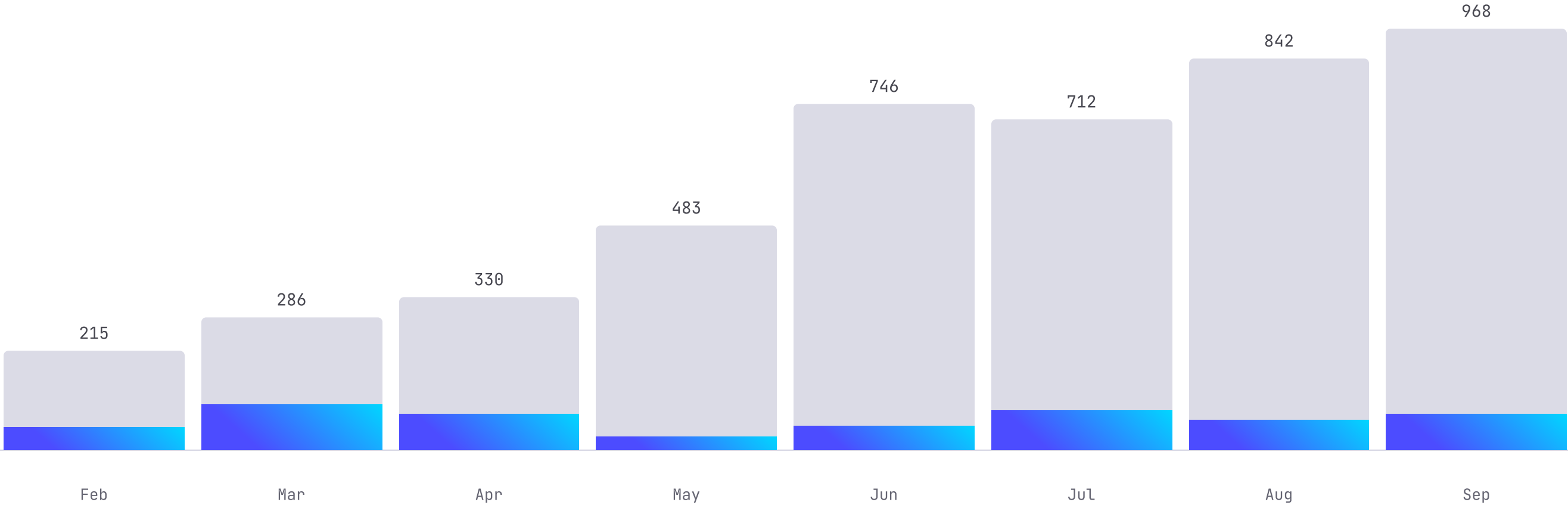
Agent-made pull requests, 8 months in.

7,831	4,582	807	16	272
agent PRs opened	merged to production	PRs driven by non-engineers	non-engineers shipping code	repos touched

4,582 merged PRs. 807 driven by non-engineers, 543 of them merged. Remember: I expected translation fixes.

It never plateaued.

Merged agent PRs of which driven by non-engineers



The top seven: 3 designers, 2 PMs, 2 from customer success.

	ROLE	PRS	MERGED
Bernhard	Product designer	215	200
Angelika	Product manager	141	124
Michael	Customer success	109	46
Kristin	Product manager	86	39
Marta	Product designer	71	27
Helya	Product designer	59	43
Lina	Customer success	53	24

Not translation fixes. Product.

PM Unread-only filter for the email inbox	PM Resizable thread list in the central inbox	PM Partner org @mentions across frontend, API and library	PM "Missing data?" banner for partner users
PM Workflow builder: explain locked loop steps, EN + DE	QA Dreadnaught: a release management control center	CUSTOMER SUCCESS Bulk sharing for saved views	CUSTOMER SUCCESS Purpose ID search in the Label Builder
CUSTOMER SUCCESS Protected environment variables	QA Snapbug: one-click AI bug reports in the browser	DESIGNER Design system contrast to WCAG AA	DESIGNER Flows sidebar rework, round after round
DESIGNER Variable Builder redesign behind a feature flag	DESIGNER Scroll markers for long entity pages	DESIGNER Org switcher bug: five symptoms, one ticket	LEADERSHIP Legal pages for the login portal

Kristin: 86 PRs. The inbox she owns, improved by her.

K

Kristin

PRODUCT MANAGER

Feb 27

Filter to show only unread emails

PR

Unread email filter

PR

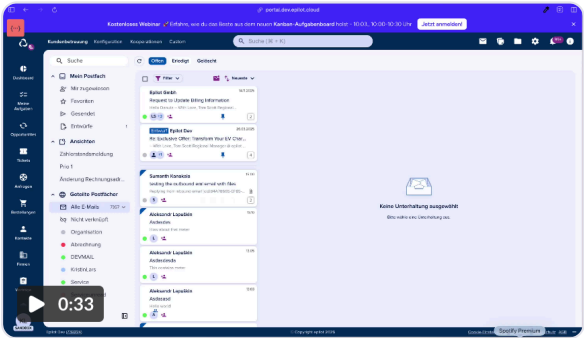
Translations

@Alex

 cc

@Leonel

0:33



2

3

K

Kristin

PRODUCT MANAGER

Mar 13

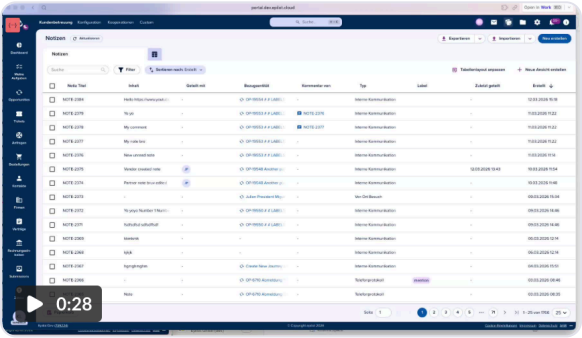
Expand shared inboxes by default

PR

PR link

@Praneet

0:28



1

1

2

K

Kristin

PRODUCT MANAGER

Mar 24

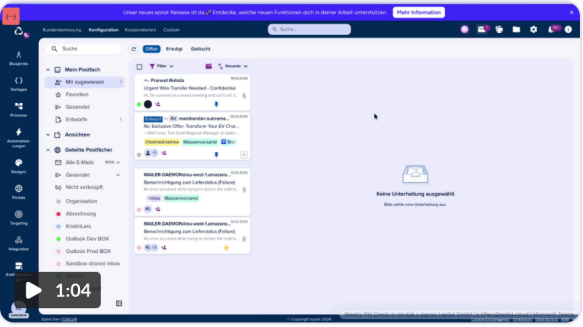
Resizable thread preview list in the central inbox

PR

PR link Preview link

@Praneet

1:04



3

2

K

Kristin

PRODUCT MANAGER

Apr 13

New entry point to create saved views

PR

epilot360-messaging

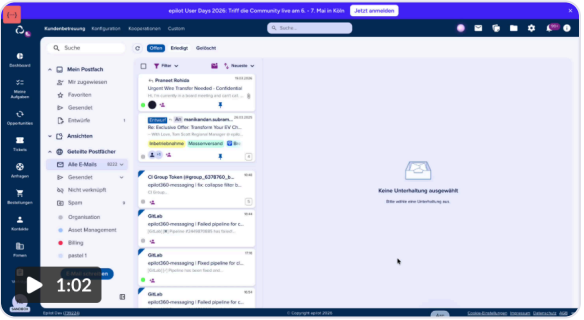
Review link

@Joe

 cc

@Praneet

1:02



2

1

1

Not every attempt is smooth. That's fine.



Kristin
PRODUCT MANAGER · #DEV-SQUAD-SIGMA

Apr 13 · 8:23 PM

Help needed @Sumanth @Praneet 🐸

I tried to build a "small" feature, which allows us to assigning Labels from email template automatically to a generated email. **Turns out it is more complicated than I thought.**

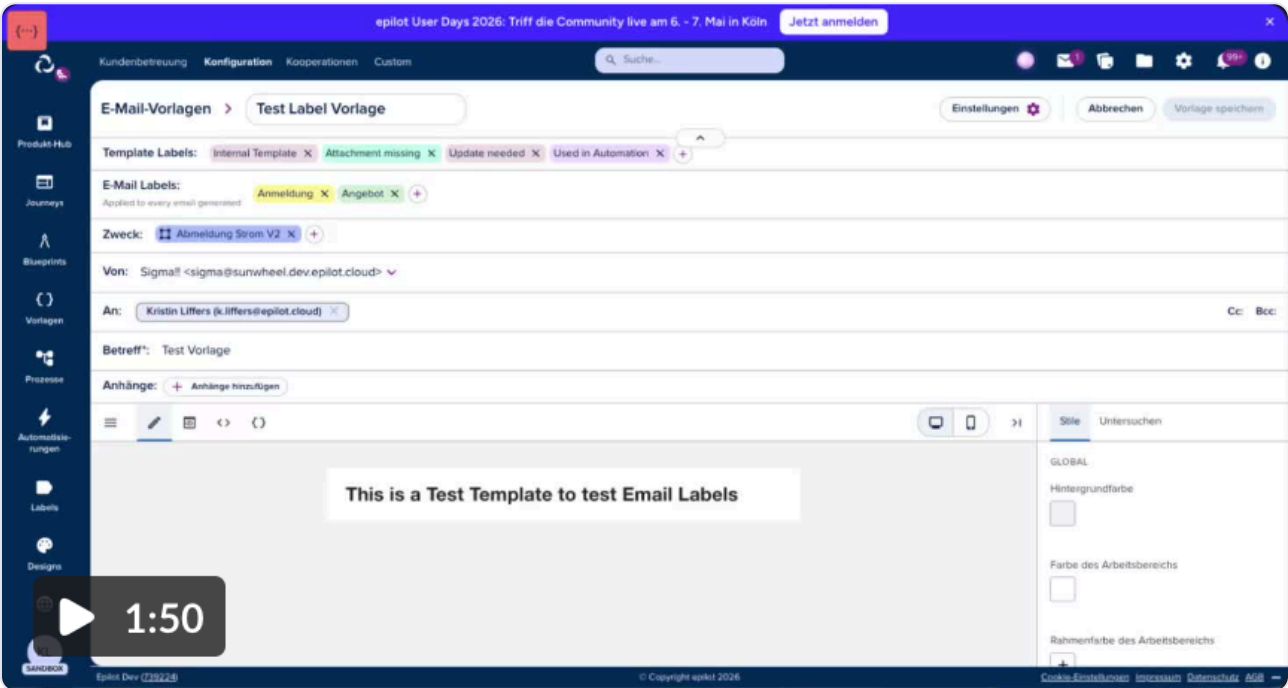
🌟 Claude session

Frontend PRs: [PR !1014](#) [PR !167](#)

3

2

1



Angelika: 141 PRs. Her UX, her fixes.



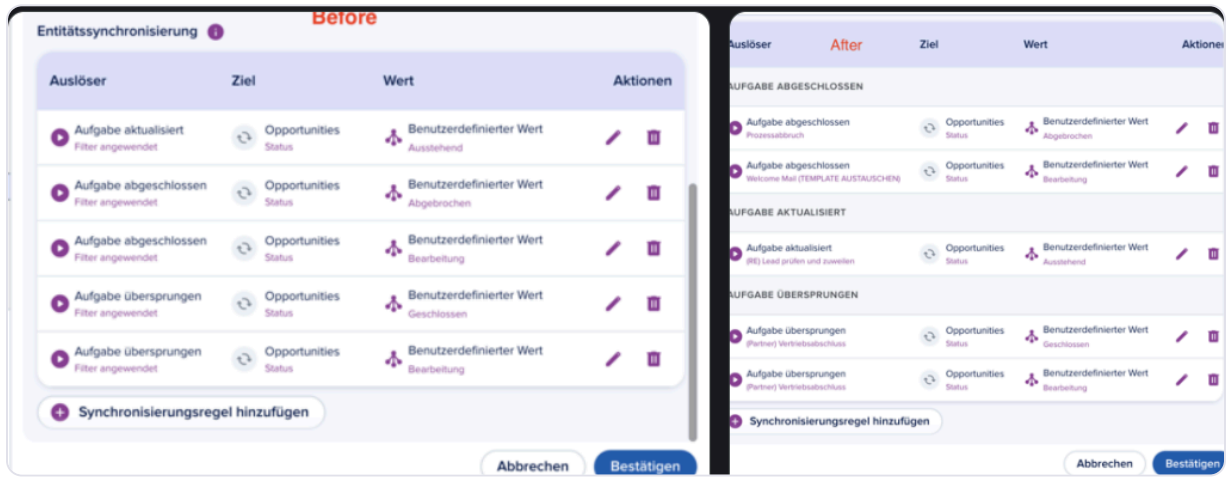
Angelika
PRODUCT MANAGER

Apr 28 · 3:18 PM

Improve Entity Sync UX Part II

PR PR in workflows-hub 👁 @Vlad @Daniel @Markus

Two pain points fixed: the trigger column finally tells you what it does, and long rule lists are grouped by trigger event.



👁 1 ✅ 1 🙌 3



Angelika
PRODUCT MANAGER

May 5 · 4:16 PM

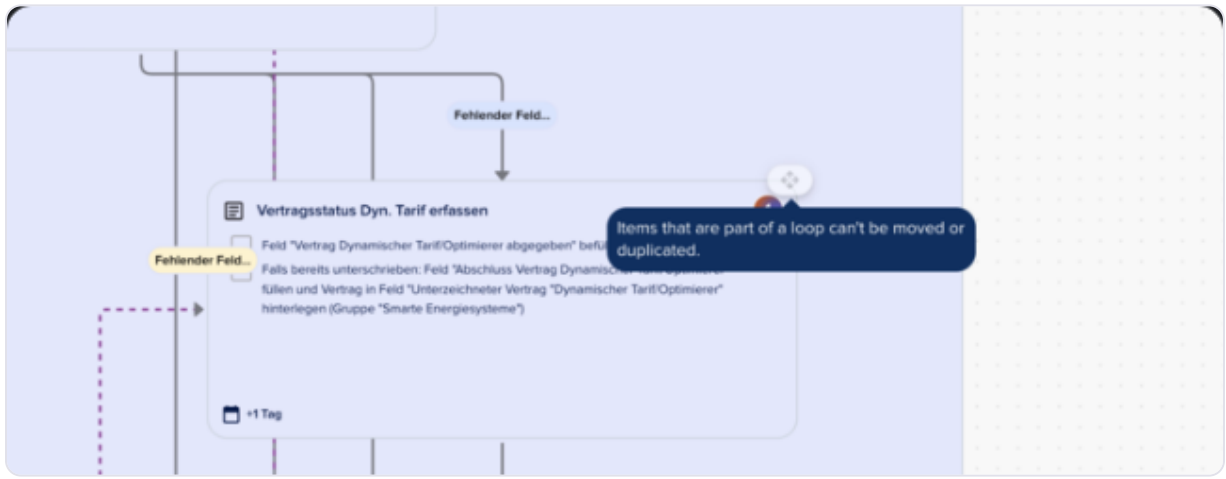
Disable move tasks button instead of hiding it for looping tasks

PR workflows-hub PR root-config

🇬🇧 Items that are part of a loop can't be moved or duplicated.

🇩🇪 Elemente, die Teil einer Schleife sind, können nicht verschoben oder dupliziert werden.

Please double check that this does not break anything in the existing logic 🙏



✅ 2 🙏 1 🧠 2

Three repos. One PM.



Angelika
PRODUCT MANAGER

Partner users grouped by organisation in @mentions. Mention a whole partner org and its members get notified. Expand a user group to see who's in it.

3 PRs · 3 repos · reviewed by 2 engineers



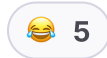
Ernesto
PRODUCT ENGINEER

"Solid work." · "Changes on Notes side are solid, both API and MFE."



Angelika
PRODUCT MANAGER

"You can thank Leidang, I didn't do much. **I only told it where it can find everything.**"



Her first PR. Better than most tickets.



Lina

CUSTOMER SUCCESS · #DEV-SQUAD-SIGMA

Mar · 3:06 PM

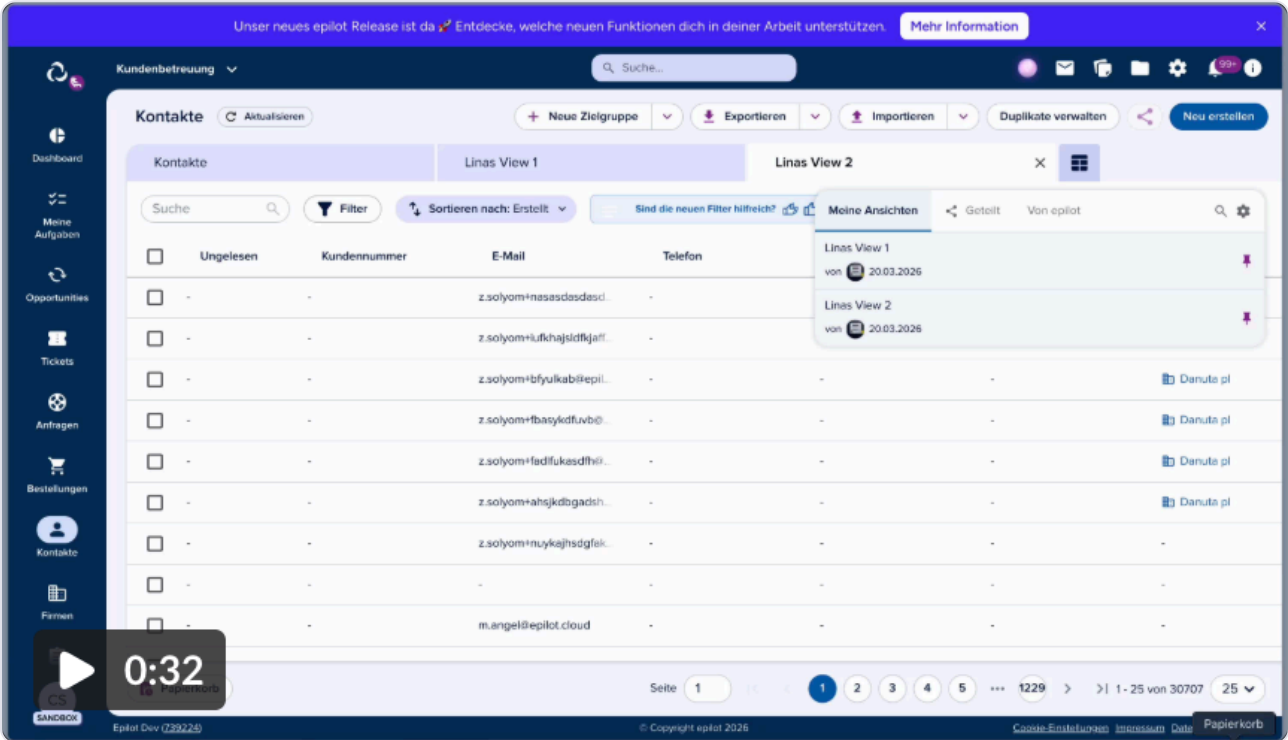
Hey [@sigma-eng](#), I figured that since I already have "Engineer" in my title, I guess it's about time to post my first MR. 🤔

Pain: When I have multiple views that I want to share with the same person or group, I currently have to do it individually for each view. This wastes time unnecessarily and is very frustrating.

My solution: Offer bulk actions for views. Now I can check any view I want to, for example, share or delete.

PR [Add multiselect to saved views manager for bulk sharing](#)

What do you think?



From first PR to shipping to prod.



Lina
CUSTOMER SUCCESS

Sep 2026

"I noticed that it wasn't possible to search for purpose IDs in the Label Builder... I've fixed this."

before and after recordings attached · 53 PRs since February



João
DISTINGUISHED ENGINEER

"The change seemed good  **deploying the change to prod** 😊"



From "can you" to approved PR in ten minutes.



Michael
CUSTOMER SUCCESS

12:26

"@Claude can you add translation for this load more button"



Michael
CUSTOMER SUCCESS

12:35

Pull request: add translation for the load more button



Viljami
ENGINEER

12:36

"Perfect, thank you! 🙏" · merged



"Overengineered or worth it?"

Judgement, not permission.



Michael
CUSTOMER SUCCESS

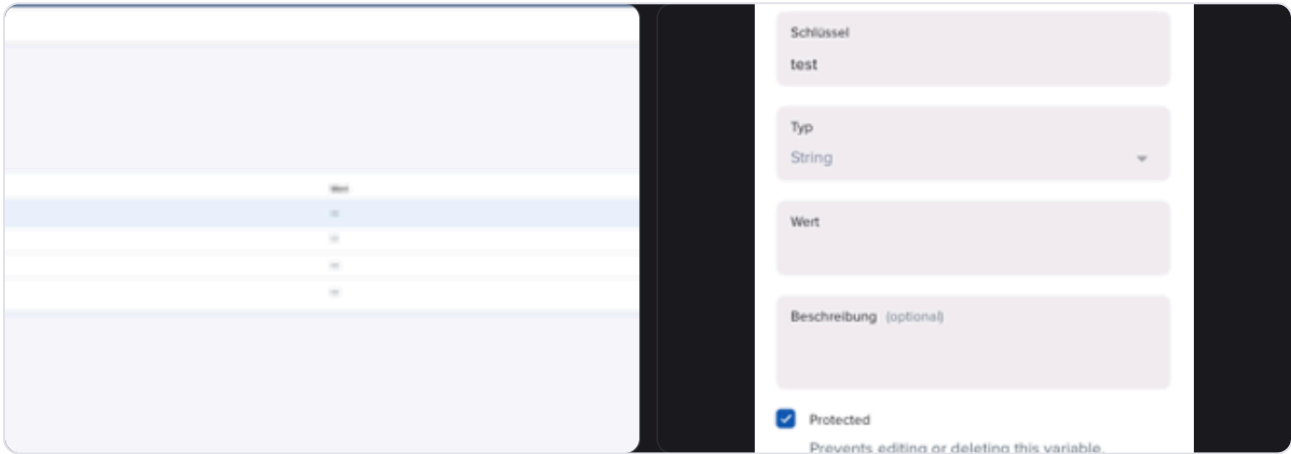
Mar 30 · 1:31 PM

hey @Viljami mind checking this one, when you got the time:
It's about protecting integration related env. variables. Currently a pretty cheap implementation, similar to locking feature in EB. More better solution might be to add a permission "edit protected env variables". But if we want this, i think i would go with the easy solution first.

WDYT about this in general. Overengineered or worth it?

[PR epilot360-environments](#) [PR epilot360-root-config](#) [Review link](#)

1 2 3



Not a Figma file. A working prototype.

**Wilian**

DESIGNER · #DEV-SQUAD-TAURO

Apr 1 · 4:39 PM

Wrapping the day on a good note @tauro 😊 I've touched up a couple of things and it's looking promising to me. Check out the recording 🎥

Things I've changed:

- Added a minimalist toolbar for visualization settings
- Added a design selector to preview the card with different designs
- A product / price selector to change the entity pairing on the tile
- Now the preview can be accessed also from the price entity

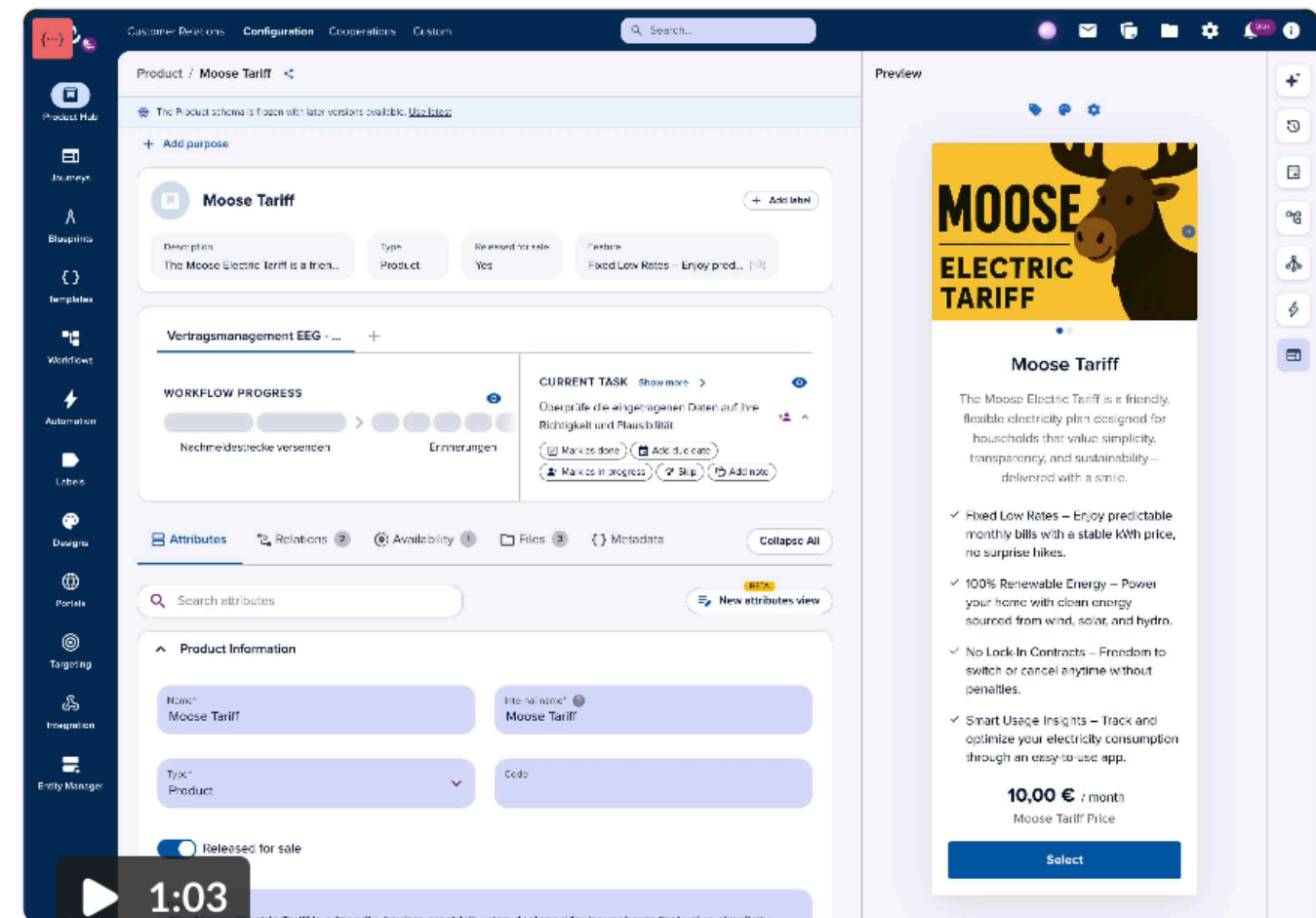
⚠️ I'm not planning to merge this. Just getting hands dirty with the idea.

**João**

PRODUCT ENGINEER

4:44 PM

@Wilian fuckin legend 🍊🍊🍊🍊🍊



The design proposal arrives as a preview link.



Wilian
DESIGNER

Scroll markers for long entity pages. Draft PR and a live preview.

"Cool idea! I like it, especially that you made it discreet when not hovering."

Senior product engineer



Wilian
DESIGNER

Sandbox badge reworked for both navigation versions. Shrinks to a dot when the nav collapses.

"This looks good. I have merged this and **enabled for all** again!"

Team lead, engineering



Helya
PRODUCT DESIGNER

"I'm now implementing some smaller changes with Claude code myself to keep things moving... **If you see a reason why I shouldn't, pls lmk.**"

June 2026 · 59 PRs since February



Designers fix the design system themselves.



Bernhard
PRODUCT DESIGNER

Input placeholder contrast now meets WCAG AA across every field state.

~~1.9 : 1~~ → **4.65 : 1** in light mode.

PR in the design system · sent using @Claude



4



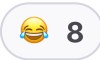
3



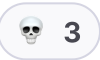
Bernhard
PRODUCT DESIGNER

On a highlighted left border in a redesign:

"Claude loves this border. It's basically the — of Claude Design."



8



3



2

A designer takes the bug ticket.

The engineer brings the context.



Marta
PRODUCT DESIGNER

Picked up a bug ticket in the org switcher. Five symptoms, two PRs: the fix and the translations.

71 PRs since February



Nishu
TEAM LEAD, ENGINEERING

"We intentionally removed the popup on org change." · kept 4 of 5 fixes

"I fixed and merged this ✅"



The person closest to the problem writes the fix. **The engineer holds the context nobody wrote down.**

Our QA built **two products of his own.**



Niko
QA · #DEV

May 19

Dreadnaught is live: Release Management Control Center

Replaces the manual Jira dashboards, scattered Slack pings, and my copy-paste workflows with a single source of truth.

- Live bug metrics and blockers, per-team status
- Post-release analysis across releases
- E2E coverage: 102+ Playwright specs, 18 domains
- Feature flags and customer-reported bug trends

The whole thing was built in a few weeks with Claude Code.



Niko
QA · CHROME WEB STORE

Jul · Sep

Snapbug 🐛 bug reports in one click

Screenshot the tab, the AI drafts the ticket with epilot knowledge, you eyeball it, it's filed under your name.

- Record button: the repro clip rides along on the ticket
- Redacted console and network logs attached
- Now files product ideas too: "the problem, not the wish"



Vlad
HEAD OF ENGINEERING

"Everyone using Snapbug in the Breakathon to report bugs"

ACT 03

What broke.

And the guardrails we added after.

A FEW WEEKS IN

Someone pushed unreviewed code. It nearly reached production.

Non-engineers got Maintainer rights in GitLab, same as every engineer. Nobody thought twice. Code went straight past review.

Nothing broke. We downgraded everyone who isn't an engineer that same day. **We got lucky.**

One rule everyone can remember.

AGENTS AND NON-ENGINEERS

Open pull requests. As many as they want.

ENGINEERS

Review and merge. Same bar as any other change.

SANDBOX

No prod access

Agents run in cloud sandboxes.
Nothing deploys outside CI.

IDENTITY

Dedicated bot users

Narrow scopes. Code access only.
Zero customer data.

VISIBILITY

Public channels

Agent work happens where
engineers can see it.

OWNERSHIP

You trigger it, you own it

The output is yours, not the agent's.

Lock down merge permissions before you open access. Not after.

No drama. Just a review request.



Angelika
PRODUCT MANAGER

Mar 11 · 10:10 PM

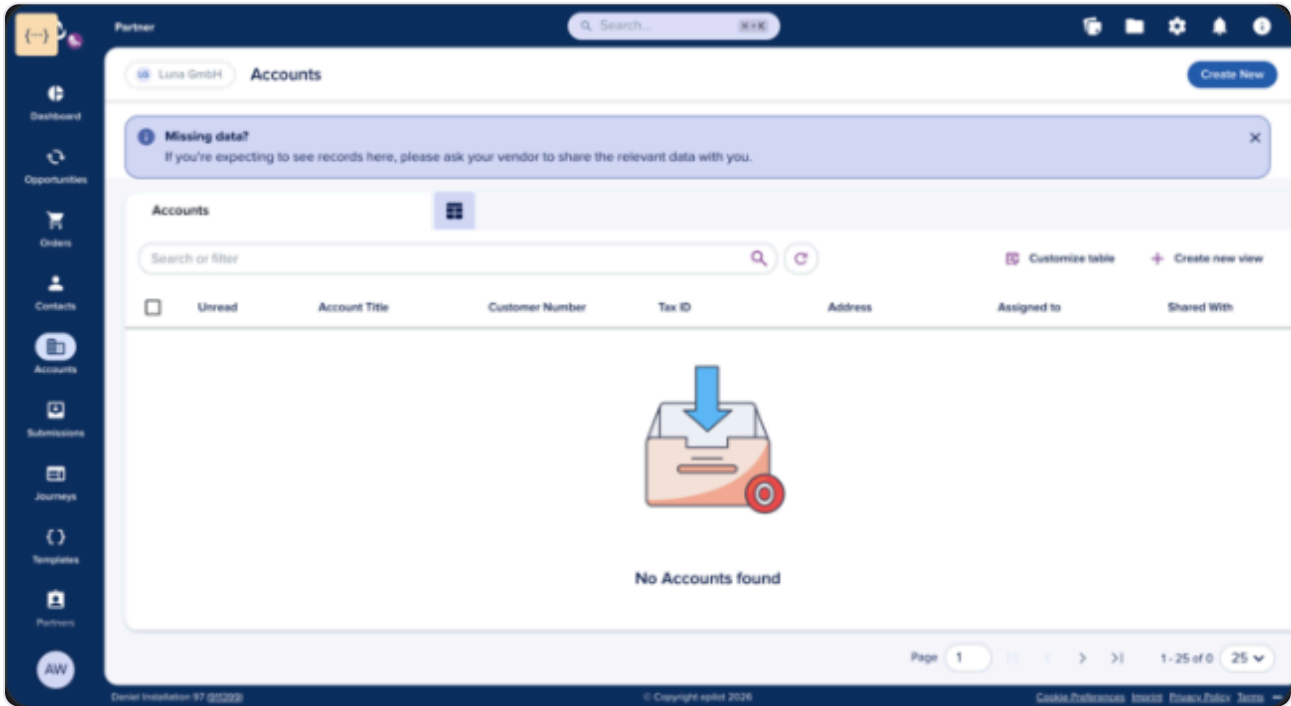
PR Partner should see a banner if the entity table is empty **PR** Translations

@Ernesto

We now have a banner that tells the partner to ask the vendor if any data is missing when the entity table is empty. 😊

→ Since I don't have the permissions to merge anymore, please review the MR and then merge it. Thanks 🙏

1 1 2



We locked ourselves out of production AWS.

An agent with an engineer's prod credentials created a stack in production, then "cleaned up" by deleting it. Outside CI/CD. Monitoring caught it. Nothing broke.

Detection is not prevention.

01 · REQUEST Portal behind SSO Pick a level and a duration. Write why.	02 · APPROVE In Slack, in public A second factor on a separate surface. Self-approval is fine.	03 · CONNECT Console link or jump host The jump host has no AI agents. None allowed.	04 · EXPIRE Gone on its own Default level is read-only.
---	---	---	--

A default AWS profile can't tell you apart from your agent. **No laptop reaches production anymore. The announcement got cheers.**

Adoption went **up**, not down.

36 → 1 PRs non-engineers merged themselves per month*	362 non-engineer commits in their peak month (Sep)	3.4× monthly merged agent PRs, March vs Sep
---	--	---

Controls didn't slow anyone down. **Guardrails build trust, and trust is what drives adoption.**

* One exception: a designer who now owns our design system merges there himself. Trust is earned.

What still goes wrong.

REVIEW LOAD

Review is the new bottleneck

Writing code got cheap. Saying "no, not like this" didn't. A few senior people merge a big share of everything.

WRONG FIXES

Not every PR should ship

A CS fix for a duplicate-detection gap: "not the proper approach." The engineer fixed the root cause instead.

RETRIES

Agents are confidently wrong

"Claude found this after he was 2-3 times wrong." Non-engineers learn to push back. Most do.

TRUST

Engineers over-trust too

"Unfortunately engineers also just trust whatever AI gives them, so we have some weird translations." A PM, about engineers.

67% of non-engineer PRs get merged. The agent average is 59%. **The ones that shouldn't ship, don't.**

Four things I'd do again.

01 · HIRE THE COWORKER Put the agent in Slack, where the work is discussed. Public channels. Tag it like a colleague.	02 · OPEN THE CODEBASE Give PMs, designers and support the same agent. Preview links on every PR so they can see what they changed.	03 · GUARDRAILS FIRST Lock merge permissions before you open access. Lock every laptop, and every agent, out of production.	04 · WRITE IT DOWN AI needs your context, not a better model. Put every repo in one place and index how things work.
---	---	---	--

The people closest to your customers can change your software now.

Are you letting them?

The bottleneck was never who can write code. It's who understands the problem.

Questions → Your turn.

WANT THIS IN YOUR ORG?

Find me after, or on LinkedIn.

I help engineering orgs open their codebase to AI agents and to the people closest to the customer. One day on site: what to unblock, which guardrails first, and what your first 30 days look like.

linkedin.com/in/viljamikuosmanen

viljami.io

productengineer.org