



tokenome

Your code says what changed. Tokenome says why, for 38% fewer tokens.

Code provenance for AI-written software

38% is context tokens, measured over forty controlled sessions; the method and the limits are later in this deck.

Local-first · Cross-tool · Your conversations stay on your machines

The reasoning is trapped in chat history

The diff still gets reviewed. The argument that produced it sits in a transcript on one laptop, in one tool, attached to nothing.

BEFORE

The reasoning lived in review

Design happened in issues, RFCs and pull request threads. Slow, but searchable by anyone, and attached to the code it justified.

NOW

The reasoning lives in a chat window

The alternatives considered, the constraint that killed the obvious approach, the reversal three days later: all of it in a conversation nobody else can reach.

RESULT

git blame names someone who did not decide

It returns the engineer who accepted the suggestion. Correct, and useless, when the question is why the code is shaped this way.

Nothing has vanished. Every one of those conversations still exists; none of them is linked to the code it produced.



Four questions your repository cannot answer

Each one is answerable in principle. Today each one costs an engineer most of a working session, or gets answered from memory and guesswork.

ONBOARDING

Why is it like this?

The new hire asks. The answer is in a transcript on a laptop they will never have access to.

INCIDENT

What did we already try?

Mid-incident, the fastest route is knowing which fix was attempted and why it was abandoned.

REVIEW AND AUDIT

On what basis was this written?

Regulated teams are being asked what produced a change, on what evidence, and with what human review.

DEPARTURE

The engineer left.

Their code stays. The reasoning behind it walks out of the building with their laptop.

Ask which one is live for you right now. The rest of this deck answers that one.



WHAT PROVENANCE IS

The line, the commit, and the conversation that caused it

tokenome

Dashboard

Search

Code context

Conversations

Journal

FAQ

Team

Sources

Settings

Code context

Pick a file and a line: tokenome blames the line and finds the conversations that were happening just before that edit.

/Users/Shared/acme-payments/ledger.py

line 31

Conversations before the edit

30 min

Find conversations

Line 31 of /Users/Shared/acme-payments/ledger.py

Last edited by Dana Whitfield · 9/27/2026, 10:50:00 AM

blame: live · b73adffffeae · Go back to direct account lookups

Conversations immediately before the edit

claude-code

claude-opus-4-1

acme-payments

9/27/2026, 10:15:00 AM

score 0.000

open

Two Northwind postings landed on their old account after this morning's migratio...

Q: Two Northwind postings landed on their old account after this morning's migration. Is that the cache?

Yes. The sixty second cache served the pre-migration account. Account lookups went back to direct reads, because a stale account misroutes money and the latency win is not worth that. Invalidating on migration is the option to try if p95 climbs again.

claude-code

claude-sonnet-4-5

acme-payments

9/25/2026, 1:05:00 PM

score 0.000

open

resolve_account is most of our p95 latency. Can we cache it?

Q: resolve_account is most of our p95 latency. Can we cache it?

Yes. Account lookups are now cached for sixty seconds, because tenant accounts almost never change and the lookup dominates p95 latency.

claude-code

claude-opus-4-1

acme-payments

9/21/2026, 4:24:00 PM

score 0.000

open

Support found two ledger entries with the same idempotency key, about 400ms apar...

Q: Do it, and keep the metric.

Removed the retry decorator, added the seen and get guard at the top of post_entry, and kept the ledger.post metric inside post_entry, so it fires once per posting instead of once per attempt.

Pick a file and a line

Line 31 of ledger.py, last edited under the commit "Go back to direct account lookups". That is what the repository knows.

Then read what caused it

Underneath: the conversation where someone asked whether the cache was misrouting postings, and the answer that sent account lookups back to direct reads.

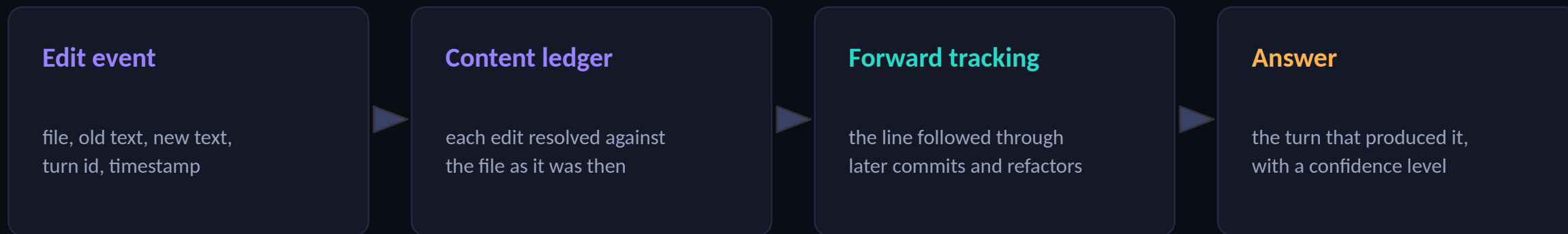
Across tools, not just one

The hits carry the assistant and the model that produced them. Here, three sessions from Claude Code over two weeks.

The project and its conversations are sample data; the software is not.

A record, not a guess

Every AI coding tool writes its edits into its own transcript: the file, the exact text replaced, and when. Tokenome keeps that as a ledger and follows the line forward through later commits.



It says when it does not know.

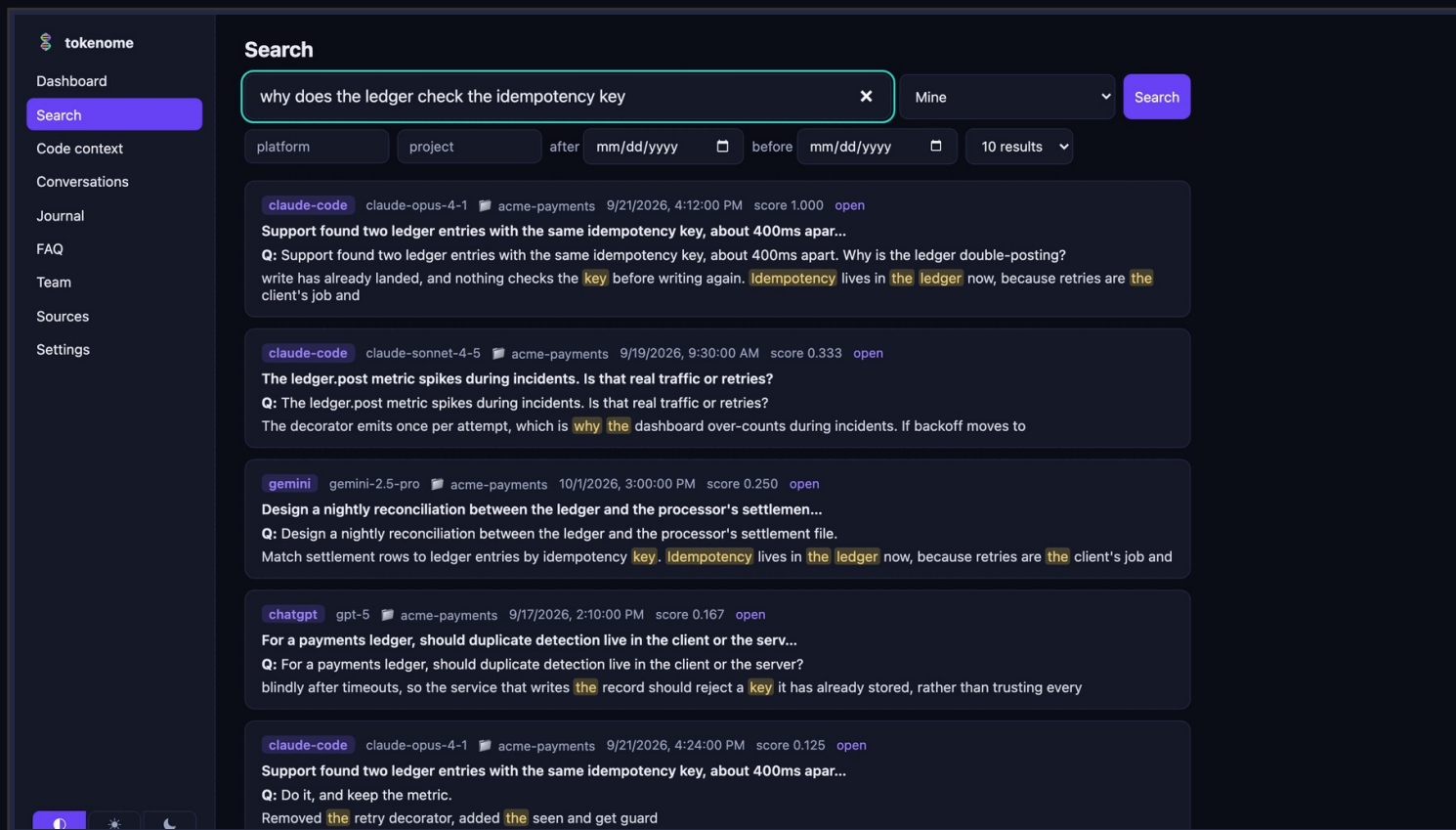
Provenance you cannot trust is worse than none. Every answer carries a confidence level, and "no recorded provenance" is a result the product is willing to return.

It works on code that already exists.

Point it at the transcripts the team already has and it answers for the work already done, not only for the next commit. Attribution tools start logging from the day you install them.



One search across every assistant your engineers use



The project and its conversations are sample data; the software is not.

Claude Code, Claude, ChatGPT, Gemini

Captured where they already run, with a pluggable adapter for the next one. Tool actions are captured, not only the chat.

Keyword and meaning, fused

Typesense BM25 fused with vectors computed on the machine. Six hits here, across three different assistants, for one question.

No vendor can do this for you

Anthropic will never index your ChatGPT history and OpenAI will never see your Claude Code sessions. Each vendor remembers only itself.

THE RECORD THAT SURVIVES THE PEOPLE

The journal writes the decision down, with the quote that proves it

Search answers a question you thought to ask. The journal is what is there when nobody thinks to ask.

Journal · 2026-09-27 [← all days](#)

1 of 1 segments were substantive, across 1 conversation. Projects: acme-payments.

1 segments

Decisions

Account lookups went back to direct reads because a stale account misroutes money and the latency win is not worth that.

Account lookups went back to direct reads, because a stale account misroutes money and the latency win is not worth that.

reversed · said by assistant · [open the cited turn](#)

The project and its conversations are sample data; the software is not.

Written once a day

What was decided, and why, as a plain markdown file on disk. Each machine writes its own.

Cite it, or drop it

Every reason quotes the sentence it came from and names the turn. A claim that cannot be quoted is discarded rather than softened.

Reversals stay visible

The decision that got undone is the one people most need to find, so it is kept as an entry of its own.

WHERE THE PAYOFF SHOWS UP

Your agents ask the record themselves

```
coding agent · acme-payments MCP ⓘ

you    why is resolve_account not cached any more?
agent  calling tokenome · why_was
      {"decision": "account lookups are direct instead of cached", "top_k": 2}

tokenome 2 records, each with its conversation
2026-09-25 DECIDED
Account lookups are now cached for sixty seconds
because tenant accounts almost never change and the lookup dominates p95 latency.
receipt > conversation account-cache · doc 9ee4431f475c
2026-09-27 REVERSED
Account lookups went back to direct reads
because a stale account misroutes money and the latency win is not worth that.
receipt > conversation stale-account · doc e1560795724e

agent  calling tokenome · get_conversation
      {"conversation_id": "stale-account"}

2026-09-27 · segment 0 · doc e1560795724e
Q Two Northwind postings landed on their old account after this morning's migration. Is that the cache?
Yes. The sixty second cache served the pre-migration account. Account lookups went back to direct reads, because a stale
account misroutes money and the latency win is not worth that. Invalidating on migration is the option to try if p95 climbs
again.
```

A real MCP exchange against the sample project. The project and its conversations are sample data; the software is not.

An MCP server, already spoken

Registering it is one command. The assistant then calls `why_was`, `ask_faq` and `search_memory` on its own, instead of asking the engineer to re-explain the project.

Two records, each with its receipt

One decided, one reversed, each carrying the conversation it came from so the agent can open the cited turn.

This is where the 38% comes from

The agent retrieves instead of grepping its way back through the repository, and every operation avoided stops being carried forward.

38% less context, no measurable change in answer quality

Ten questions about past work, asked four ways by the same agent over the same corpus. Forty controlled sessions; context counted from provider usage records, not estimated.

CONFIGURATION	CONTEXT		TURNS	COST
grep the repo, no index	12,515,065	baseline	252	\$14.70
index available, agent chooses	9,481,199	-24%	203	\$12.60
routed by question type · ships as the default	7,751,761	-38%	161	\$11.43
index first, always	6,652,191	-47%	148	\$11.18

38% is context, not the bill.

Dollar cost in the same arm fell 22%, from \$14.70 to \$11.43. The two numbers differ because cached and output tokens are not priced like fresh input.

The routing rule ships with it.

Left to choose freely, the agent reached for the index in only 4 of 10 sessions. Routing by question type is what turns the capability into the result above.



What we do not claim

The measurement is small and we would rather read its limits out loud than have a skeptical engineer find them later.

Not the same answers

Ten questions cannot prove answer parity. What the data supports is that no quality difference rises above the run-to-run noise, and that is all we say.

One measured deficit

Forcing the index on every question costs about 1.7 points of retrieval recall, roughly two standard deviations. That is why the shipped default routes by question type instead.

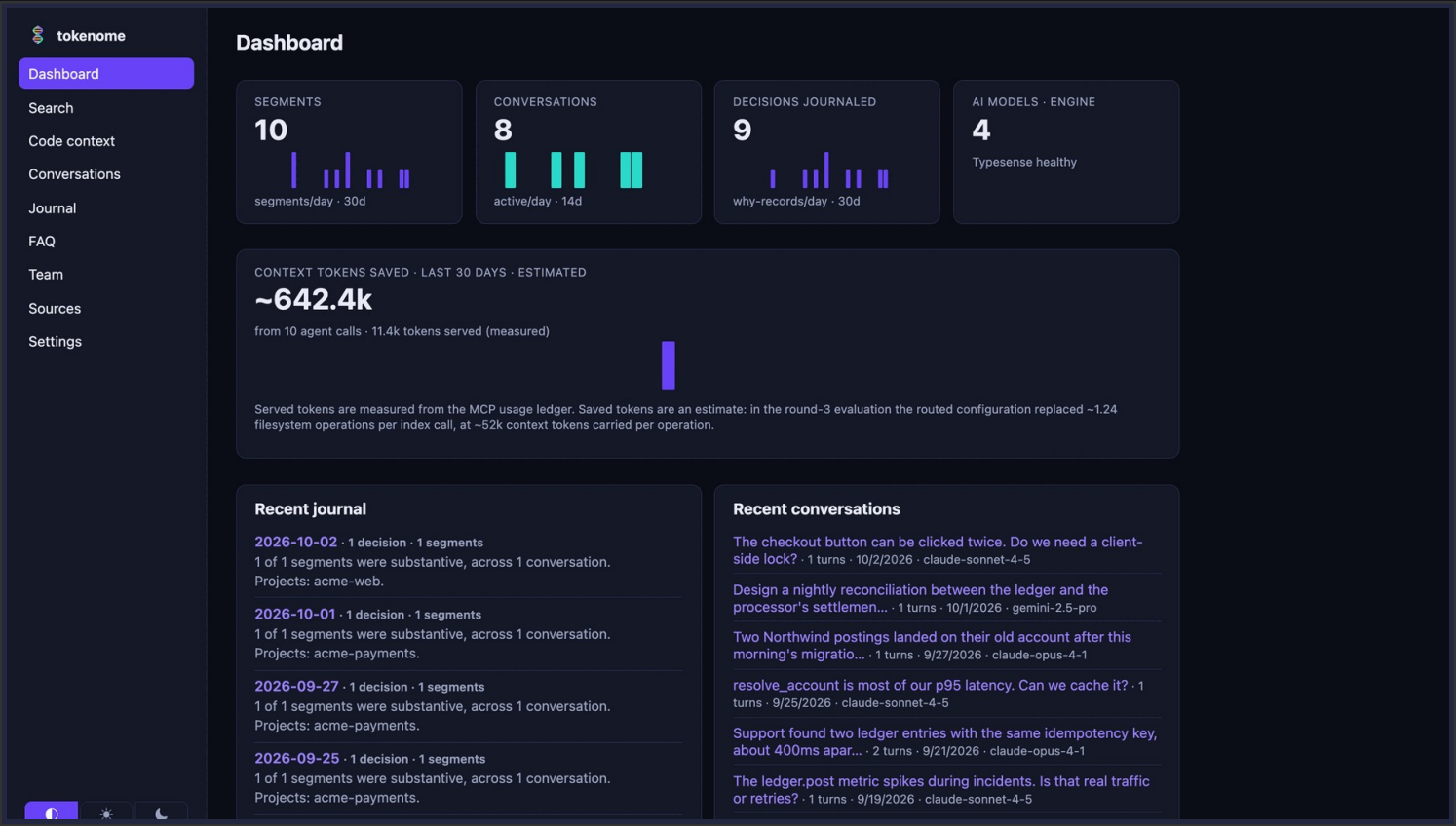
One corpus, one agent

Forty sessions, one codebase, one model family. Your corpus will differ, which is what running it on your own repositories is for.

The evaluation page publishes the arms, the noise floor and the raw counts, including the ones that do not flatter us.



Free on your machine. Paid when the answer has to leave it.



The project and its conversations are sample data; the software is not.

One network call, disclosed: the app checks the releases page for updates. It sends no conversations and no code, and there is nothing else to fetch.

The whole product, on one machine

Capture from every assistant, hybrid search, code provenance, the journal and the FAQ, the web UI and the MCP server. Provenance is in the free tier permanently: if it only worked when you paid, you could never check that it works.

Not a trial. Not a freemium clock.

No account, no sign-up, no card. Nothing leaves your machine. No cloud embeddings: the model is in the download and runs on the CPU you already have.

HOW YOU GET IT

Install it today: the app, the package, or the plugin

github.com/tokenome/releases, or the package index. Nothing to configure either way, and it indexes the transcript history already on the machine.

THE APP • MACOS AND LINUX

A signed and notarized DMG, AppImage or deb

macOS on Apple Silicon: drag Tokenome to Applications and launch it. Linux on x86_64 and aarch64: run the AppImage or install the deb. The AppImage and the DMG update themselves, and every download carries its .sha256.

THE COMMAND LINE

uv tool install tokenome-ai

The package is tokenome-ai. It installs three commands: tokenome for the CLI, tokenome-mcp for the server your agent talks to, and tokenome-web for the web UI. uv brings the Python with it, so use uv, not pip.

CLAUDE CODE

One marketplace, one plugin

claude plugin marketplace add tokenome/releases, then claude plugin install tokenome@tokenome. The agent can then ask your history itself; from the app, tokenome claude install does both.

Everything is in the app download: the Python runtime, the search engine and the embedding model are inside the bundle. Nothing else to fetch and no model download on first run. The app also installs the command-line tool for you: Settings, then Install command-line tool.

No Windows build yet. tokenome.ai/windows says so and takes the request; we will not put a date on it.





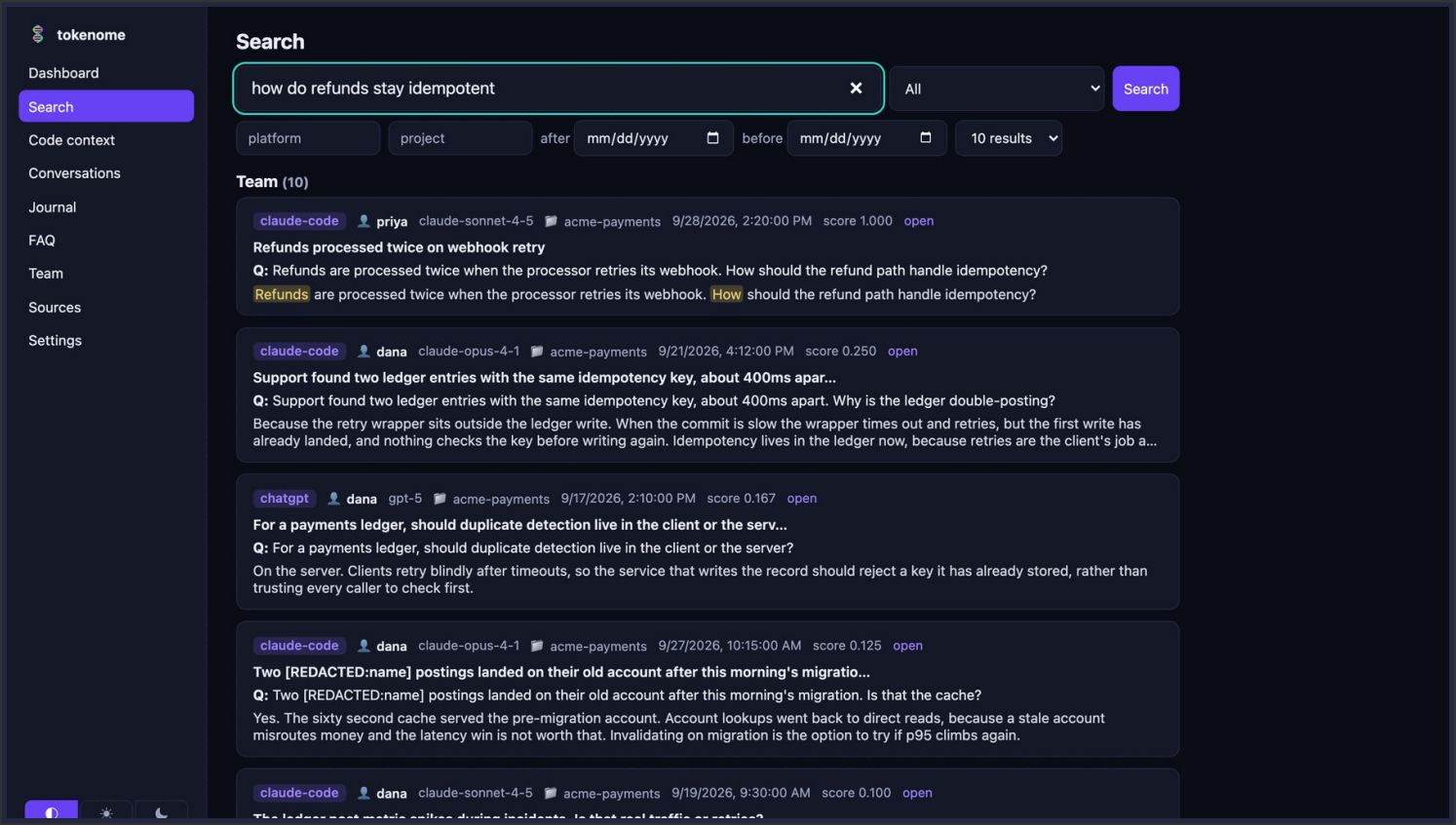
TOKENOME TEAM

One search across the team. On a server you run.

Real, in beta, and free during the beta. There is no end date on the beta; when it ends, the price on the next slides is the price. Each laptop keeps its own private index, and a project has to be opted in before a single line of it is mirrored.

What it does · The admin console · The security posture · How it is run · What it costs

One search covers the team, with who said it



The project and its conversations are sample data; the software is not.

Mine and Team, in one query

The same question runs over your own index and the shared one. Here the team hits come back attributed to priya and to dana.

Attribution, not anonymity

A result is only useful if you can go and ask the person. Every shared hit names whoever shared it, and the project it came from.

It degrades politely

If the server is unreachable the search still returns your local results, and the queue on disk ships the rest when it comes back.

Nothing is shared until someone shares it

tokenome

Dashboard

Search

Code context

Conversations

Journal

FAQ

Team

Sources

Settings

Team

http://127.0.0.1:8612

enrolled as dana

Check server

Push now

Backfill of acme-payments finished: 8 segments shipped, 1 held back for review.

Held back from sharing (1)

The share gate found something it could not make safe. Nothing here has left this machine. Release sends the redacted form; discard keeps it local for good.

acme-payments · reconciliation #0

matches your exclude pattern settlement file

Design a nightly reconciliation between the ledger and the processor's settlement file.

Release

Discard

Projects

Sharing a project mirrors its new conversations to the team server as they are indexed. Backfill sends what is already indexed locally. Unsharing stops future sharing; retract also deletes what was sent.

Project	Local segments	Shared	
acme-payments	9	☒	Backfill Retract
acme-web	1	☐	

What I've shared

8 segments on the server. Retention is set by your admin; a window means older conversations are removed automatically.

All (8)

acme-payments (8) · kept until you delete it

When	Project	Conversation	Excerpt
2026-09-27	acme-payments	Two [REDACTED:name] postings landed on their old account after this morning's migration... #0	Two [REDACTED:name] postings landed on their old account after this morning's migration. Is that the cache?
2026-09-25	acme-payments	resolve_account is most of our p95 latency. Can we cache it? #0	resolve_account is most of our p95 latency. Can we cache it?
2026-09-21	acme-payments	Support found two ledger entries with the same idempotency key, about 400ms apart. #1	Do it, and keep the metric.

The project and its conversations are sample data; the software is not.

Words the team puts in the gate's dictionary, a customer codename for instance, are substituted before anything is shared.

Opt in per project, per person

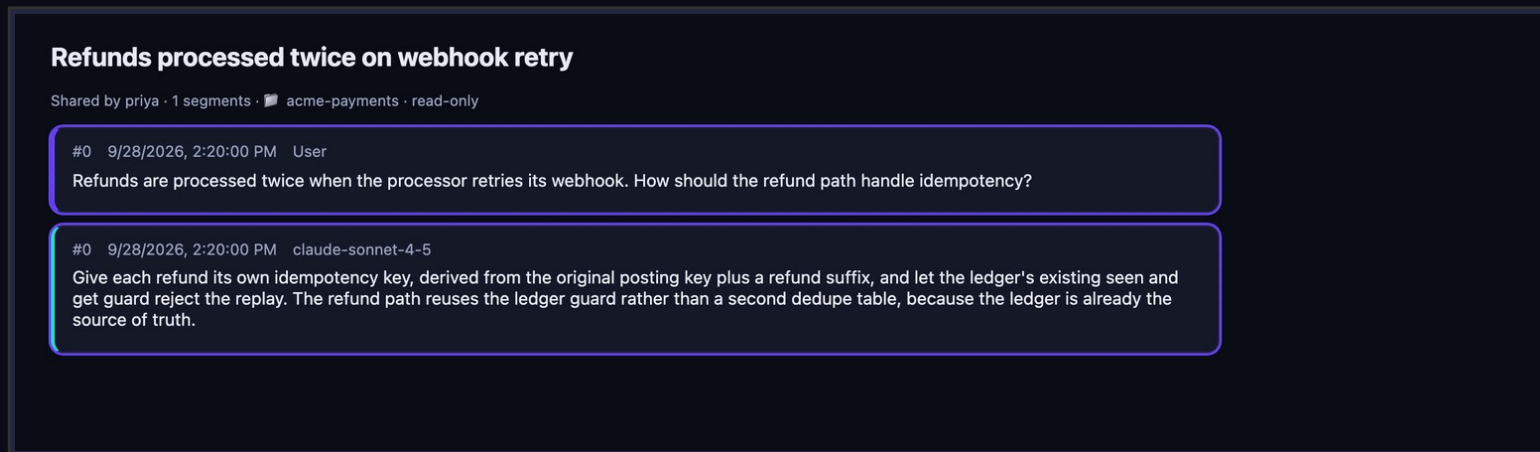
Nothing is shared by default, and a conversation with no project is never shared. Backfill sends the history you already have; retract stops the sharing and deletes what was sent.

The gate holds back what it cannot make safe

Secrets are redacted on the laptop before the batch is written to disk. Here the backfill shipped 8 segments and held 1 back, because it matched an exclude pattern the team set. Release or discard is the owner's call.

Open the conversation itself, read-only

A hit is a pointer. The thread behind it is where the reasoning actually is.



The project and its conversations are sample data; the software is not.

The whole thread, not just the hit

A shared conversation opens in a reader that names who shared it and the project it belongs to. Read-only: nothing on your machine changes, and nothing on theirs does either.

This is what onboarding looks like

The new engineer reads the argument that produced the code, in the words of the people who had it, rather than booking an hour of someone's attention to hear it second hand.

Who has access, on which devices, and how to take it away

tokenomeTeam

Acme Payments
no team id (beta)

ADMINISTRATION

- People and devices
- Sharing and redaction
- Retention and holds
- Audit log
- License and seats
- Settings

FOR EVERYONE

- My footprint

Sign out

Administration / People and devices

Free beta, no licence required. ThemeAKAisha Karim · admin

People and devices

Who has access right now, on which devices, and how to recover when a laptop is lost.

! Only one admin. If Aisha Karim's laptop is lost, recovery needs a recovery code or access to the server. Make another member an admin · Recovery codes

Members (15)Groups (4)Invites (1)Stale devices (1)

Members

No spend column here, by design.

Filter by name

Invite member

	NAME	ROLE	GROUP	DEVICES	LAST SEEN	SHARED PROJECTS	STATUS
AK	Aisha Karim	admin	No group	1 of 1	now	2	active
CJ	Clara Jensen	lead	Mobile	1 of 1	now	1	active
DF	Diego Ferreira	member	Payments Core	1 of 1	now	2	active
EP	Elena Petrova	viewer (finance)	Mobile	1 of 1	never	0	active
JW	Jonas Weber	member	Platform	1 of 1	never	0	active
KM	Kofi Mensah	member	Risk	1 of 1	never	0	active
LH	Leila Haddad	lead	Risk	1 of 1	now	2	active
MO	Maya Okafor	lead	Payments Core	1 of 1	now	2	active

Pick a member

Choose a name to see their devices, change their role and groups, recover a lost laptop, or offboard them.

The project and its conversations are sample data; the software is not.

Members, groups and devices

Four roles: admin, lead, member and viewer, with invites and stale devices on their own tabs. Open a person to see their devices, change their role, recover a lost laptop, or offboard them.

Device keys, not passwords

Enrollment generates a keypair on the laptop and the private half never leaves it. Revocation takes effect on the next request.

No spend column, by design

The console is for access and sharing. It is not a surveillance dashboard, and the engineers can see that it is not.

An append-only audit log, filterable and exportable

 tokenome Team

Acme Payments

no team id (beta)

ADMINISTRATION

- People and devices
- Sharing and redaction
- Retention and holds
- Audit log**
- License and seats
- Settings

FOR EVERYONE

- My footprint

Sign out

Administration / Audit log

Free beta, no licence required.

Theme

AK

Aisha Karim · admin

Audit log

Append-only. 67 rows since Oct 03, 2026. Search queries are never shown as text here.

ACTION

All actions

ACTOR

Anyone

FROM

mm/dd/yyyy

TO

mm/dd/yyyy

Apply

Clear

Export CSV

Entries

TIME	ACTOR	DEVICE	ACTION	TARGET	DETAIL
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	console.login	session a57eb9cb	hours: 8
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	console.link	Aisha Karim / aisha-mbp	
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	auth.login	Aisha Karim / aisha-mbp	
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	admin.retention.update		after: {"overrides": {"ledger-service": 0, "payments-api": 90}, ...}, before: {"overrides": {}, "retention_days": 0}
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	admin.hold.create	24821b5a8e78441ebd87774bd3a9c86f	kind: owner, pin_usage: False, reason: Departure review, value: 26dfa8256640419ab65b22cb19165c95
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	admin.hold.create	2024c2941f9e4d40ac64dc6c2b5ea0de	kind: project, pin_usage: True, reason: Audit FY2026, value: ledger-service
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	auth.login	Aisha Karim / aisha-mbp	
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	segments.upsert		indexed: 9, received: 9, reembedded: 0
Oct 3, 02:50 PM	Aisha Karim	aisha-mbp	auth.login	Aisha Karim / aisha-mbp	
Oct 3, 02:50 PM	Clara Jensen	clara-mbp	segments.upsert		indexed: 6, received: 6, reembedded: 0
Oct 3, 02:50 PM	Clara Jensen	clara-mbp	auth.login	Clara Jensen / clara-mbp	

The project and its conversations are sample data; the software is not.

Every administrative action

The time, the actor, the device, the action and the target, appended and never edited. Filter by action or by actor, bound it by date, and export the result as CSV.

Search queries are never shown as text

What a person searched for is their business. The log records that a search happened, not what was in it.

Enough for the review you will have to pass

This is what a security reviewer asks for first, and it is in the product today rather than on a roadmap slide.

Every member can see exactly what the server knows about them

 tokenome Team

Acme Payments

no team id (beta)

FOR EVERYONE

• My footprint

Sign out

For everyone / My footprint

Theme

PR Priya Raman · member

My footprint

Exactly what this server knows about you, and who has looked.

What you share

Content leaves your laptop only from these projects.

PROJECT	SEGMENTS	KEPT
ledger-service	6	keep forever
payments-api	5	90 days

Who viewed your data

Person views and drill-downs that touched you.

Nobody has viewed your data as a person.

Changes to your account

Audit rows about you and your devices, newest first.

WHEN	WHO	ACTION	TARGET
Oct 3, 02:50 PM	Priya Raman	auth.enroll	Priya Raman
Oct 3, 02:50 PM	Aisha Karim	admin.user.create	Priya Raman

Org settings, in plain words

What applies to everyone here.

Your role	member (Payments Core)
Smallest group shown	3 people; anything smaller is hidden
Per-person views	Off: nobody sees your numbers as a person
Content retention	180 days
Your usage	Arrives with usage reporting in a later release. Nothing about your usage is on this server today.

The project and its conversations are sample data; the software is not.

Adoption is the whole game on a team tier. An engineer who cannot see what is being collected does not opt in, and a Team seat that nobody opts into is worth nothing.

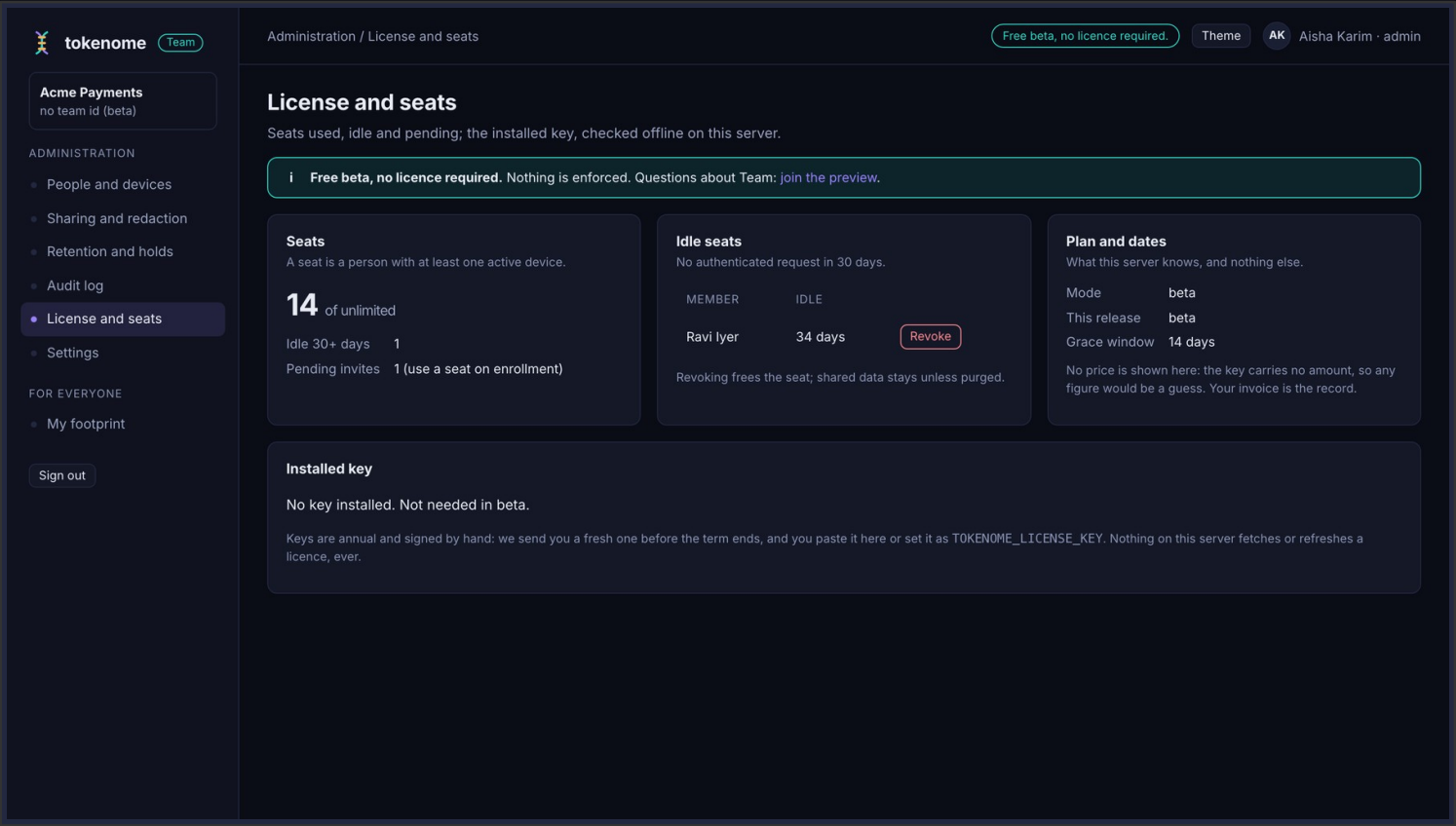
What you share, and who looked

Each person sees their shared projects, the segment counts, their enrolled devices, the audit rows about their own account, and whether anyone has viewed their data as a person.

Org settings in plain words

The retention window, the smallest group the console will show, and whether per-person views are on at all. The same page for everyone, admin or not.

Seats, and a beta that is free with no end date on it



The project and its conversations are sample data; the software is not.

Free beta, and the product says so

Nothing is enforced while the beta runs: no seat limit, no key to install, and no date on the page telling you when that stops. When it does change, the price on the next slides is the price.

A seat is a person, not an install

Someone with at least one active device. The page shows the seats in use, who has gone quiet for thirty days, and the invites that will take a seat when they are accepted.

No price anywhere on the page

The key carries no amount, so the console shows none rather than inventing one. Your order is the record.

Your infrastructure, your keys, your data

The shared server is software you run, not a service we host. We never hold your engineers' work.

How it is built

- Device keys, not passwords: the private half is generated on the laptop and never leaves it; https is enforced on enroll.
- A login signs a server nonce and gets a short-lived token. No password to phish and no identity provider to depend on.
- Clients never touch the search engine directly; the server injects the ownership filters on every query.
- Offboarding revokes the person and purges what they shared, and the purge is in the audit log.

What we will not pretend

The server holds shared text and vectors in the clear. That is what makes keyword and semantic search possible, and any vendor claiming otherwise is not describing a working search product.

The protection is organizational: your machine, your region, your disk encryption, your backups, behind your TLS. The share gate is a safety net, not a licence to put something on the server that the organization should not hold.

Not in the beta, and not described as if it were: SSO and SCIM, a provenance API, a pull request bot, a team FAQ across everyone's journals.



A server you run, on Docker or on Railway

No agent fleet, no data pipeline, and no change to how your engineers work.

THE SERVER

One image, pulled by compose

ghcr.io/tokenome/tokenome-server, smoke-tested on x86_64 and aarch64 every release. Compose pins the version and starts the index, the server and TLS; Railway runs the same image.

THE LAPTOPS

One install, one enroll

The same build as the free tier. Enrollment is one command with a token from an admin, and the person keeps their private local index either way.

THE LICENCE

One signed key, checked offline

Annual, signed by hand and pasted into the console or set as an environment variable. The server checks it locally; it never calls out for licensing. The beta needs no key at all.

The image is published with the launch release; if a pull is refused, hello@tokenome.ai gets you access. We do not host servers for customers yet, so self-hosted is the only way to run Team today. Hosting is in build; we will not sell you a door that does not open.



WHAT IT COSTS

Priced per seat, not per query

Embeddings run on your hardware, so there is no usage meter and no bill that grows with how much you search.

INDIVIDUAL

Free

forever, on one machine

- Full local capture, every assistant
- Hybrid search: CLI, web UI, MCP
- Code provenance, journal and FAQ
- No account, no card, no clock

TEAM • BETA

\$24

per seat, per month

- Everything in Individual
- One search across the team, attributed
- Opt-in sharing; the share gate
- Admin console: people, sharing, audit
- \$240 a seat a year, two months free

ENTERPRISE

Talk to us

annual, invoiced

- Everything in Team
- Quoted to your deployment
- Scoped with you before anything is priced
- hello@tokenome.ai

Three seats is the floor, because a team-wide search does nothing for one person. Free during the beta, with no end date set. Founding customers: 50% off the first year, the first ten teams, with the renewal at \$24 a seat written into the same order.

Every tier runs on your own hardware • no usage meter, no per-token billing • at \$24 a seat, Team pays for itself at three and a half minutes an engineer a week; the calculator is on tokenome.ai

Three steps, and the first one needs nothing from us

You keep the free tier either way. The only question Team has to answer is whether shared recall earns its seat price.

STEP ONE

Install the free tier and index what you have

One engineer, one laptop, the transcripts already on it. By the end of the first run there is a searchable record and a journal for work that is already done.

STEP TWO

Join the preview and stand up the server

Tell us roughly how many engineers. We help you bring the server up on your own infrastructure, enroll the team, and opt two projects in. Free during the beta.

STEP THREE

Ask the questions you could not answer

Run the real ones: the reversal nobody can explain, the module whose author left, the decision that keeps getting re-litigated. Decide on what comes back.





Your code says what changed. Tokenome says why.

Free on your machine. Paid when the answer has to leave it.

John O'Neil

Founder

oneil@tokenome.ai

Sid Probstein

Growth

sid@tokenome.ai