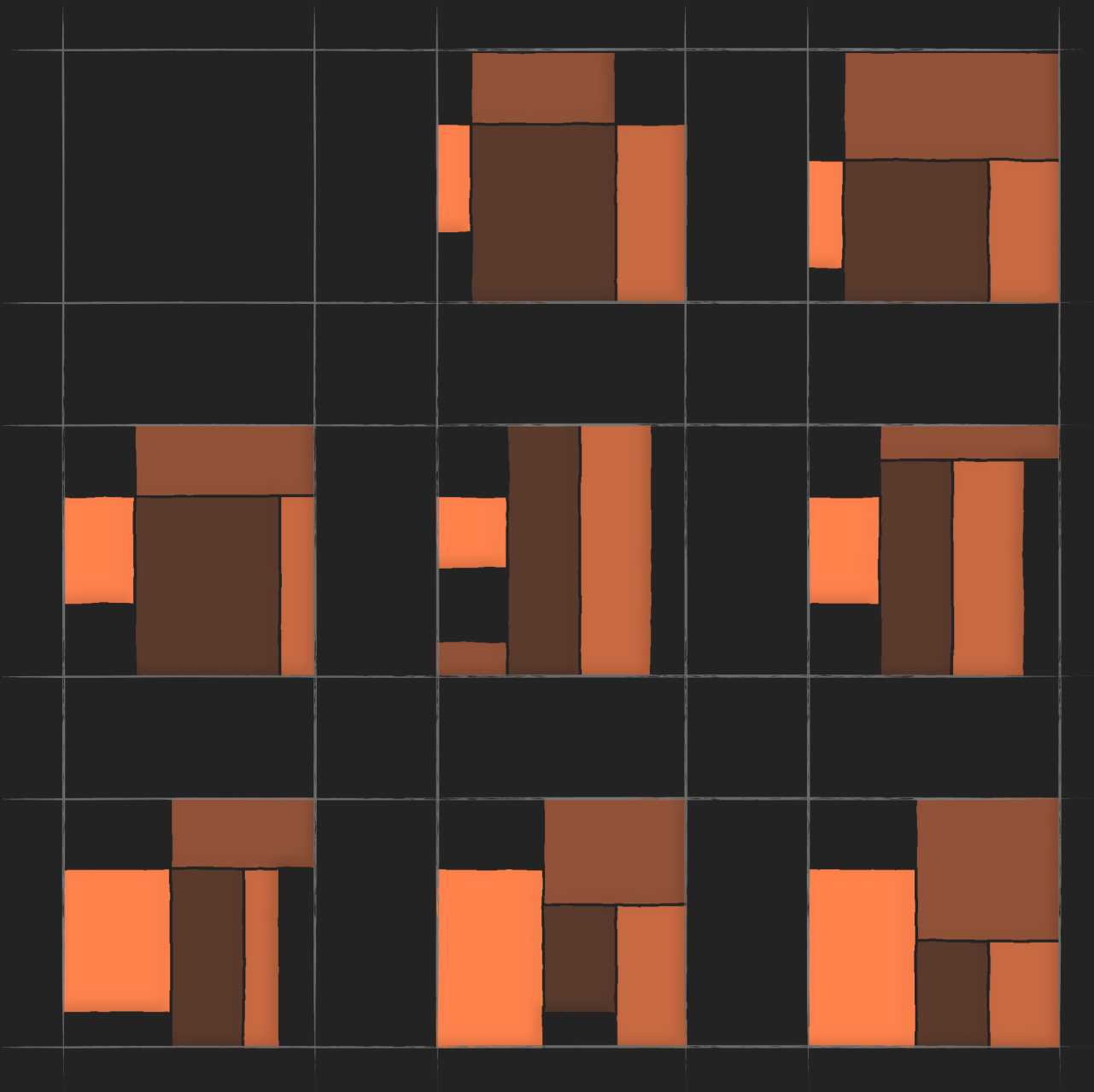


The Ontology of the Company Brain

Ebook by Slite



CONTENTS

01 **Is this... a company brain?**

On what is a Company Brain, what people think it is, and how it emerged.

02 **Why the continued hype?**

On fragmented context, the agents that choke on it, and who wants to own it.

03 **Applications of a Company Brain**

On the quick wins for humans, and the kinds of agents a brain unlocks.

04 **The Architecture of a Company Brain**

On the four components every brain shares, and the ways to build each one.

05 **Visualising real architectures from the field**

06 **Where the market is**

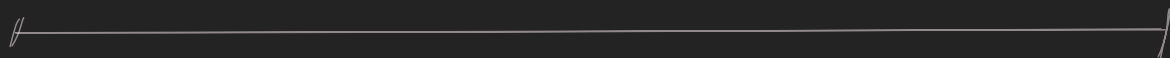
On who's building, who got burned, what everyone fears, and which brain you actually need.

07 **What the builders see coming**

On the takes every builder shares, and the questions none of them can answer.

On what is a Company Brain, what people think it is, and how it emerged.

We surveyed 149 people about how they define a company brain, traced the term's history, and mapped how awareness spread.



Is this — a
company brain?

The term "Company Brain" emerged out of thin air and spread like wildfire beyond Twitter, showing up in LinkedIn carousels, YouTube explainers, and pitch decks. People who have never opened Hacker News are now asking their ops lead whether the company has a brain yet.

Company Brain is not an entirely new concept. People have long wanted to solve the problem of finding context, and over the last year, we have seen quite a few solutions that aim to do just that. Depending on when you started following the space, you might have competing definitions of the category.

People think the Company Brain is 4 different things

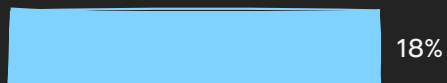
There are a few popular interpretations of the Company Brain, so we asked the respondents which one resonated with them.

What people think a company brain is

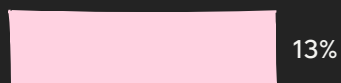
Continuously updated knowledge base



Wiki maintained from team activity



AI search across tools and knowledge



Memory layer for agents



Don't know the term



As we see from the top 2 selections, people describe the Company Brain while being agnostic to whether it's a layer for agents or humans, and primarily want a Company Brain to solve context accuracy.

71% of teams expect knowledge that keeps itself accurate



■ Maintained knowledge base (53%) ■ Generated wiki (18%)

slite Company Brain Survey 2026

It maps closely to YC's official definition, published when YC included a Company Brain in its YC '26 RFS.



“ Every company has critical know-how scattered across people's heads, old Slack threads, support tickets, and databases, and AI agents can't operate like that. We think every company in the world is going to need a new primitive: a living map of how the company works that turns its own artifacts into an executable skills file for AI. ”

Tom Blomfield
from YC's RFS 2026

Simply put,

A company brain takes everything your company knows (context in docs, CRMs, issue trackers, people's heads) and makes it usable as answers or context.

For people and agents alike.

YC's definition is one recent version of a much older idea. Tracing that history shows why the term has stuck and what people now expect a company brain to do. So let's look at its etymology.

The etymology of 'Company Brain'

Nobody woke up in April 2026 and invented the company brain from nothing. The idea assembled itself in public, over four years, out of five distinct moments.

2022

The second brain



Tiago Forte publishes *Building a Second Brain*, and the idea that your notes should think with you goes mainstream.

 BOOK

2025

Context graphs



Jaya Gupta and Ashu Garg coin Context Graphs: the argument that all of a company's scattered context should be connected into a queryable structure.

“ ESSAY

April 2026

LLM wikis



Andrej Karpathy describes LLM Wikis: living documents that an AI maintains as it works, instead of a pile of notes people have to keep current.

✂ x

April 2026

GBrain



Garry Tan says every YC company should have a GBrain: a shared, current picture of the business that the whole team can query.

🏠 YC

April 2026

The name



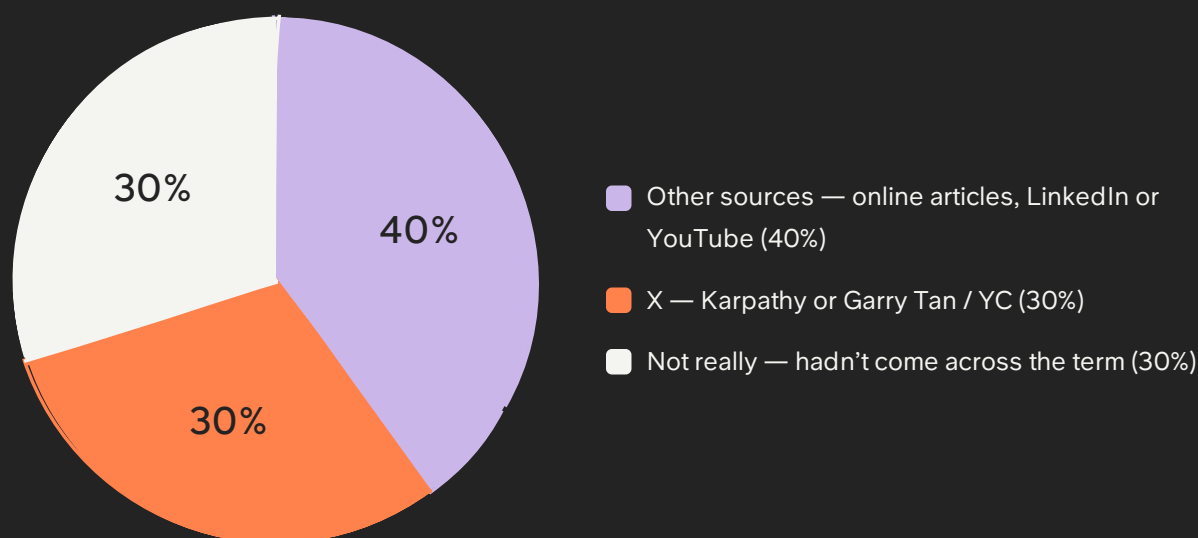
YC's Request for Startups names the Company Brain as a category: the company's memory, made usable.

🔗 RFS

4 months after YC's definition, 56% of people know the term from sources beyond X

A term coined by Tech Twitter's most-followed accounts is now most often encountered in places Tech Twitter doesn't reach. Four months after YC named the category, 56% of respondents had encountered the term somewhere other than X. The idea had moved beyond its original audience and into broader conversations about how companies manage context.

How people discovered the term



slite Company Brain Survey 2026

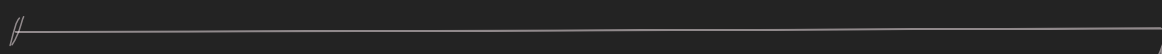
Beyond the purpose of a Company Brain, there are many follow-up questions to answer before we can fully understand it, such as:

- Why does it have continued hype?
- What are its core architectural modules?
- What are its core applications that everybody is excited about?
- What are the open questions we have around the company brain?

We put these questions to survey respondents and vendors actively building Company Brains. Their current hypotheses are detailed in the next section.

On fragmented context, the agents that choke on it, and who wants to own it.

Company Brains have seen enduring hype because they solve the growing problem of context fragmentation which reached an inflection point after we started working with AI Agents.



Why the continued hype?

Companies have been managing land, machines, and money for centuries.

But managing what a company collectively knows is a problem we've been working on for a few decades.

The problem grows with the company. More people, more tools, and more context created every day have increased the urgency of finding a tangible answer to:

How does everyone stay on the same page?

For the last twenty years, our answer was to imagine a single source of truth for information. Every wiki, intranet, and shared drive was built on that insight, and entire companies were built around selling it. No wiki so far, including ours, has ever come close to solving the problem because modern work outgrew the model.

A modern business runs on structured records, unstructured conversations, databases, tickets, meeting recordings, and design files, and each of these has earned its own dedicated tool.

So the stack kept growing, and the distance between knowledge workers and the insight they needed kept widening.

Your context is fragmented, and will stay that way.

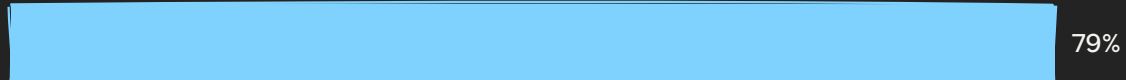
The average company now splits its knowledge across several systems.

Where company knowledge lives

Dedicated wiki or knowledge base



Two or more systems at once



Shared drives and folders



Slack or Teams threads



Email chains



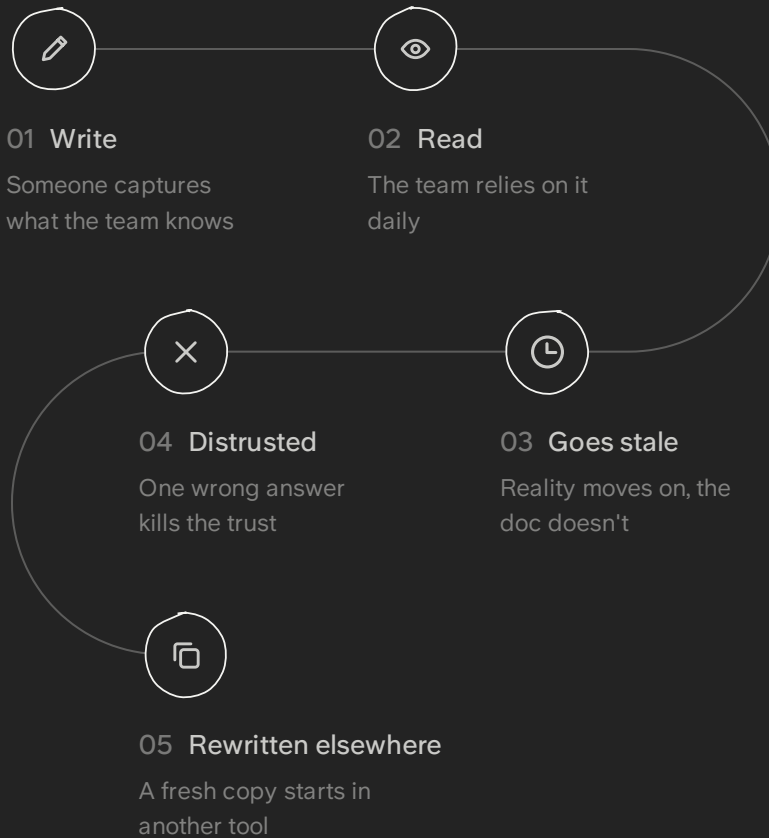
slite Company Brain Survey 2026

At the same time, we can only expect this stack to grow as vertical AI SaaS solutions come into play. Expecting the wiki to be the single source of truth may have been misguided then, but today it is positively wrong.

So the emergence of the Company Brain is, in part, an acknowledgement of the reality of our current software stack. A Company Brain has to work across sources and meet data where it lives, rather than mirror everything into one canonical repository.

Documentation's broken promise

Since context gets fragmented, docs get neglected. Neglected docs go stale, and [stale documentation](#) actively misinforms the team that trusts it.



“It's always the documentation, and not the AI, that is the culprit. In any organization, there are many documents already available. Having led engineering teams of more than 100 members and founded Motion Design School, I am familiar with this trend. The knowledge is spread across Slack, Notion, tickets, emails, code comments, and people's minds. It keeps happening because people do not trust where the knowledge is kept.”



Vitaliy Kononov
Co-Founder and CTO, Atty

This is why most survey respondents wanted a company brain for people first.

Stale documentation creates a trust problem for the knowledge worker who needs it for their everyday work. That loss of trust explains why teams want a company brain for humans first, even though agents make the problem more urgent.

Why teams want a company brain

Answers for me



Me and my team



Power my agents



All of the above



This was one of the most surprising findings for us because only 7% of respondents wanted a company brain solely for their agents.

92% of teams expect a company brain to serve people—not agents alone.



Human or team use case (92%) Agents only (7%)

slite Company Brain Survey 2026

Agents amplify every problem humans faced with stale documentation

Every problem above has been survivable because humans supply the judgement the docs lack. When a person finds an obviously outdated document, they can check it with a colleague.

Agents can't. Agents need that same context to do any work at all, and they arrive with none of the judgement.



“ Let's say a junior who joined four weeks ago decides to write a lengthy report on a subject matter he doesn't know anything about. He saves it in a Confluence doc, and this is all fine because no one finds this page as long as there is no AI. But out of nowhere, AI finds this single page and draws very strange conclusions from it because it assumes that this page holds authority because it's in a real Confluence doc. I wonder how you funnel all this in a way that you're not blocked all the time. ”

Florian Redeker
Product Manager, AllUnity

Because an AI Agent has no judgement about accuracy or surrounding context, autonomous Agents can be dangerous in production: they risk producing false output, creating duplicate work, and causing reputational damage.

One context problem, two consequences

	 For humans	 For agents
Tribal knowledge	 Costly You ask the person and wait. When they leave, the knowledge leaves.	 Severe The knowledge does not exist. The agent builds answers around a hole it cannot see.
Missing context	 Survivable You misapply a rule once, then ask around for the backstory.	 Severe The agent applies the rule universally. A missing caveat becomes a policy violation at scale.
Channels conflict	 Survivable You cross-check versions and choose with judgement.	 Costly The agent retrieves whichever version ranks highest and presents it as truth.
Outdated docs	 Survivable You notice the page feels stale and verify before acting.	 Severe A page from 2024 reads as confidently as yesterday's update.
Half never write	 Survivable You know whom to ask instead.	 Costly Whole teams are invisible, and the agent does not know what it is missing.
Cannot keep up	 Costly Doc debt grows quietly and new hires drown in it.	 Severe Every stale page becomes a new landmine exactly when the system is needed most.

“ A stale doc no longer just misleads a human who might double-check; it silently propagates into proposals, client deliverables, and automations. The cost has shifted from 'someone wastes 20 minutes' to 'the system confidently executes on wrong facts'. ”

Survey respondent
on what stale documentation costs their team

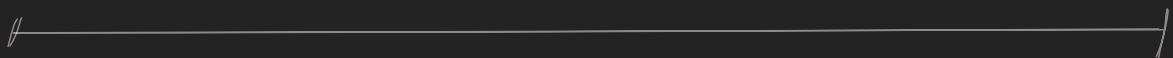
The takeaway from this section can be distilled into three interconnected truisms:

- Documentation promised to be the single source of truth, but context fragmentation has proved that it can never fulfil that promise.
- Documentation's false promise creates a problem of stale context, which fractures trust in the reliability of async documentation.
- While humans have worked around stale documentation to minimise damage, Agents have no ability to do so.

That's why Company Brains are a response to three core problems at the heart of knowledge work. In the next chapter, we'll explore how Company Brains solve these problems and go beyond them to unlock completely novel ways of working, including autonomous Agents.

On the quick wins for humans, and the kinds of agents a brain unlocks.

A Company Brain unlocks clean, accurate context which dramatically changes how we work with each other, or the kind of work we can trust our AI Agents to do.



Applications of a Company Brain

A company brain does two things.

It gives humans answers faster, and it gives agents answers when they need them, even when they're running a cron job at 3am with no human to ask.

Most people start with the first and discover the second as they scale agentic work, unlocking novel workflows and handing projects off to vertical Agents without context assembly. Once you have your first Company Brain-backed Agent in production, you don't go back.

For humans, the brain replaces the longest part of every task: assembling context.

Respondents told us documentation is actively used by 4+ teams per company on average. So when a Company Brain works, the wins immediately extend to everyone else who doesn't document, because they probably ask colleagues who do.

- A new hire asks the brain instead of interrupting five people.
- Marketing reads what actually shipped without pinging engineering.
- Someone needs to know why you killed that pricing plan in 2024, and gets the answer with sources, even after everyone involved has left.
- The Slack ping that costs a teammate twenty minutes of focus becomes a query the brain answers in seconds.

None of these are revolutionary on their own. Together, they solve the sprawl of [tribal knowledge](#) and compress the time between "I need to know something" and "I know it" from hours to seconds, across every team that touches the brain.



“ I'm often too lazy to construct a proper context. And I think this is the way most people work at the moment; they don't even know how much better the results could get. ”

Florian Redeker
Product Manager, AllUnity

For agents, the brain is the difference between useful and dangerous

The people we surveyed are past the experimentation phase with AI and use it every day at an intermediate to advanced level, with their own set of skills, agents, and hooks.

How fluent teams already are with agents

One pick each. Agent-native sits inside the 57% who already deploy agents.

Deploy their own agents and skill files



Agent-native, included in the 57%



Still use AI mainly as chat



slite Company Brain Survey 2026

Every one of those agents needs company context to do real work, and each kind needs a different slice of it. Each type below sets out what it pulls from the brain and what goes wrong when it can't.

Eight agent types, one shared brain

Horizontal assistants

The chat window that knows the company

Needs from the brain

Retrieval across all company context, permissions-aware

Breaks without it

Confident answers from stale or partial docs. The classic “AI said our refund policy is X” incident.

Support agents

The first answer the customer hears

Needs from the brain

Product docs, known issues, policy edge cases, tone guidelines

Breaks without it

Wrong answers to customers at scale, escalation rates that erase the cost savings.

Marketing/content agents

Copy that knows what actually shipped

Needs from the brain

Brand voice, customer stories, what’s shipped vs roadmap

Breaks without it

Generic slop, or worse, announcing features that don’t exist.

Coding agents

Code that remembers why you built it that way

Needs from the brain

Architecture decisions, past incidents, internal conventions, “why we did it this way”

Breaks without it

Technically correct code that violates every unwritten team convention, re-litigating settled decisions.

Sales agents

Deals that know the last six quarters

Needs from the brain

Pricing rules, past deals, positioning, competitive intel

Breaks without it

Off-brand proposals, discounts nobody approved, positioning from 2 quarters ago.

Ops/HR agents

The onboarding answer nobody has to give twice

Needs from the brain

Policies, org structure, approval chains, tribal knowledge

Breaks without it

New hires get confidently wrong answers in week one, trust never recovers.

 Self-maintaining knowledge agents

The agent that keeps the brain itself current

Needs from the brain

The brain itself, plus signals about what's stale

Breaks without it

The brain rots, and every agent above degrades with it.

 Background/ambient agents

Work that runs when nobody is watching

Needs from the brain

Everything above, with zero human in the loop to catch errors

Breaks without it

These simply can't exist. This is the category the company brain unlocks outright, not improves.

Seven of the eight agent types above get better with a company brain. The eighth, background agents, can't exist without one.

Background agents run triggered workflows, scheduled jobs, and multi-step automations with zero human in the loop. They're the category everyone wants next. They're also the category that breaks hardest on stale context, because no one is there to catch the error.

By now, if you're convinced you want a Company Brain, you might be wondering: how do you actually build one?

We talked to founders of Company Brain solutions and people who have built one for their own businesses, only to find that the architecture does not follow a clean step-by-step guide. It depends on the technology and data you're working with. We'll share all our findings in the next section.

On the four components every brain shares, and the ways to build each one.

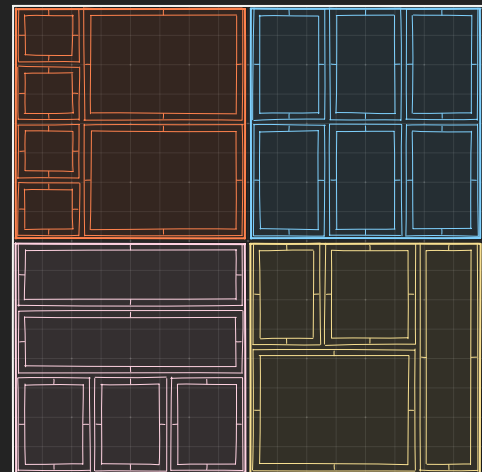
Under the hood, Company Brains differ in their architecture by orders of magnitude. However, each one decomposes into the same four components.

The Architecture of a Company Brain

Every company brain has the same four components.

We looked inside a dozen company brains for this book, open source and commercial, personal and company-wide, DIY and bought. They differ everywhere except in one place. Each one decomposes into the same four components.

The brain gets signals, it remembers, it dreams and prunes, and it speaks when spoken to.



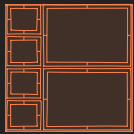
The four parts of a company brain

- | | |
|--|--|
| ■ Getting signals | ■ Remembering |
| ■ Dreaming and pruning | ■ Speaking and searching |

After interviewing builders and test-driving OSS Company Brains, we realised that every implementation had different components, each trying to solve one of the following four problems:

- Getting Signals → How do you pass data from sources such as Slack and your CRM to your Company Brain?
- Remembering → How does the Brain remember what's important?

- Dreaming and pruning → How does the Brain form new connections and prune older information?
 - Speaking and searching → What's the interaction layer between the Brain and its end users?
-



Getting signals

Getting signals is how the brain learns what happened in the company today. Without a steady feed, every answer it gives describes an older version of your company.

Tools produce activity at different speeds and volumes, so capture methods trade freshness against cost and noise. Four approaches cover the field.



Scheduled pulls. A connector syncs each tool at a fixed interval. Useful for slow-moving sources like docs and CRM records.



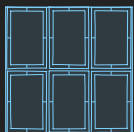
Event capture. A webhook pushes each event to the brain the moment it happens. Useful for fast-moving sources like chat and tickets.



Agent ingestion. An agent reads a source, keeps what matters, and writes a structured summary into the brain. Useful for noisy sources like meeting transcripts.



Inference-time retrieval. Nothing gets stored. The brain queries live sources the moment a question arrives. Useful when the source system should stay the only copy.



Remembering

Remembering is the form in which the brain stores knowledge so it can be found again. That form decides which questions the brain will be able to answer later.

Richer forms answer more questions but cost more to build and maintain, which is why four coexist.



Plain files. Readable by any human and any agent. Useful for small technical teams that want zero infrastructure.



Documents. A knowledge base with owners and permissions. Useful when humans are the primary readers.

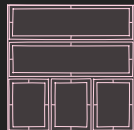


Vector index. Searched by meaning instead of keywords. Useful for large piles of unstructured content.



Knowledge graph. Entities and relationships stored as typed edges. Useful when questions involve relationships or history.

Some brains combine two of these on purpose: readable docs for humans plus a dense, structured tier that only agents open.



Dreaming and pruning

Dreaming and pruning are how the brain keeps its own knowledge current without being told to. Knowledge rots, and a brain that never prunes becomes the stale wiki it replaced. Together, dreaming and pruning help create a [Knowledge Base that maintains itself](#).

Staleness shows up in different ways: through age, disuse, or contradiction with reality. Each needs its own routine.



Consolidation runs. A scheduled job reorganizes and re-links everything that came in that day. Useful for keeping memory coherent as it grows.



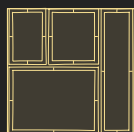
Time decay. Old and unused facts lose retrieval weight, or expire outright. Useful against silent aging.



Drift detection. The brain compares what docs claim against live activity in connected tools and flags mismatches. Useful for catching pages that reality has left behind.



Review queues. Every change the brain proposes waits in an inbox for human approval. Useful wherever a wrong update costs more than a slow one.



Speaking and searching

Speaking and searching are how people and agents get answers out of the brain. This is the only component users ever touch and the payoff for the other three.

Different askers need different doors: humans want a conversation, while machines want an endpoint.



Chat. Plain-language questions, answers with citations back to sources. Useful for everyday team questions.



Hybrid search. Vector similarity plus keyword matching plus a reranking pass. Useful as the machinery underneath every other mode.



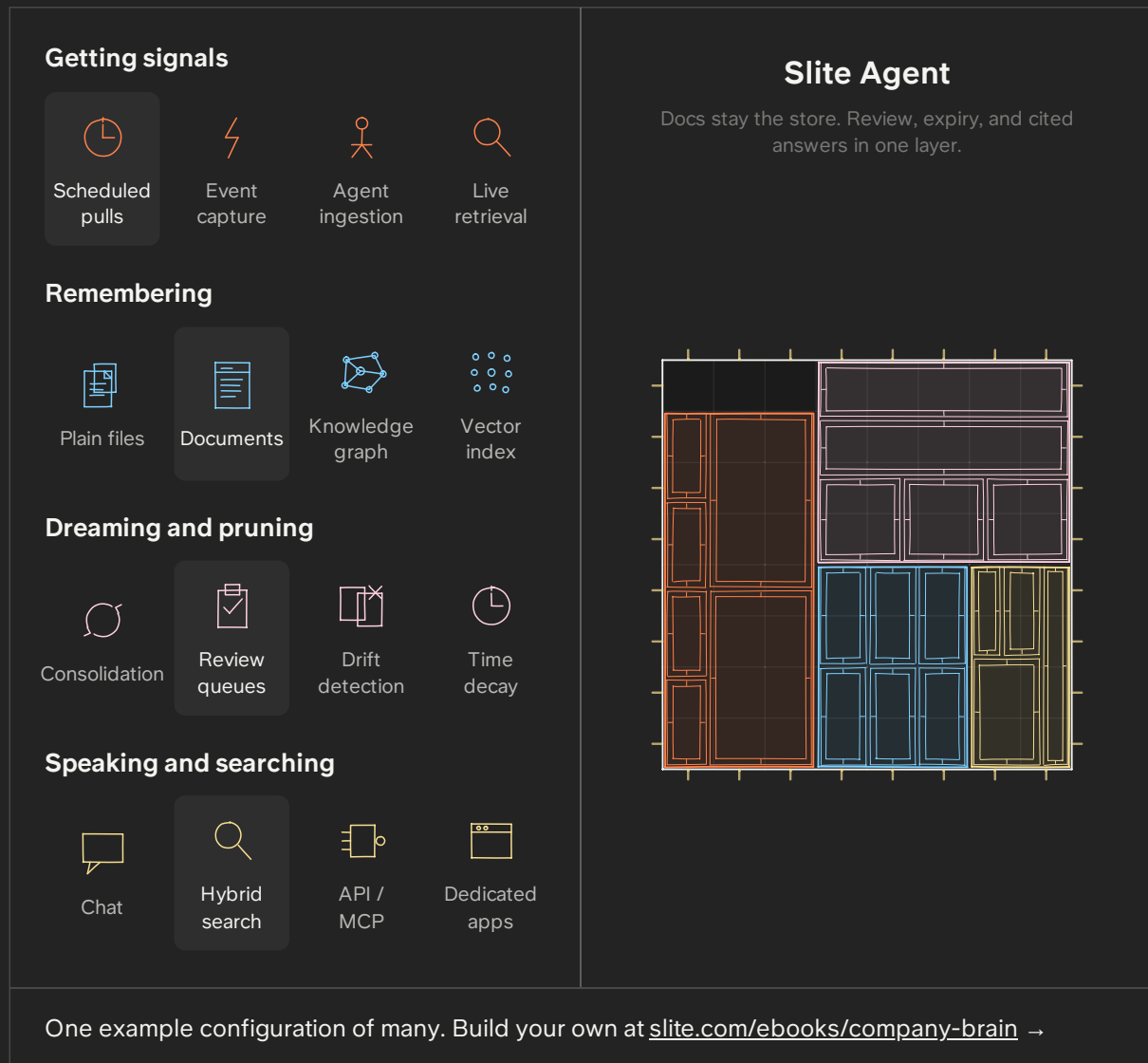
Headless API or MCP. The brain exposed as an endpoint that other systems query directly. Useful for plugging the brain into coding agents, support bots and workflows.



Dedicated apps. Purpose-built interfaces for a specific team or workflow. Useful when one use case deserves its own surface.

One brain, many ways to build it

Depending on how you mix and match the components laid out above, you get different versions of a Company Brain, each with its own caveats and nuances.



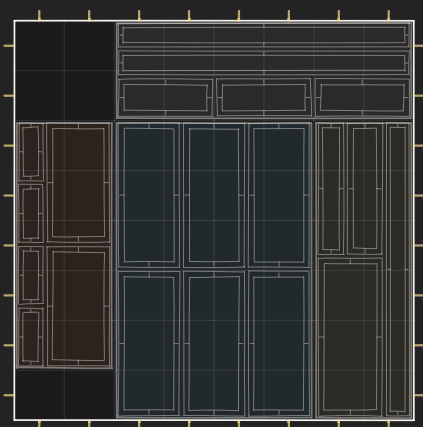
One example configuration of many. Build your own at slite.com/ebooks/company-brain →

Visualising real architectures from the field

Breaking down the architectures of the most popular open-source and closed-source brains that are being used by real users for their own teams today.

What are the most popular technical architectures in the market?

Each company brain below shows how it handles ingestion, context storage, cleanup, and access for humans or agents.



GBrain

Git keeps the durable record. A scheduled dream cycle maintains the graph and consolidates memory overnight.



Getting signals

Email and calendar come in through [scheduled collectors](#). General webhooks, voice notes, an inbox folder, and custom skillpacks give you other ways to feed the brain.



Remembering

Your Markdown repo stays the system of record. GBrain indexes it with [PGLite by default, or Postgres with pgvector for shared deployments](#), and turns local wikilinks into typed edges without an LLM.



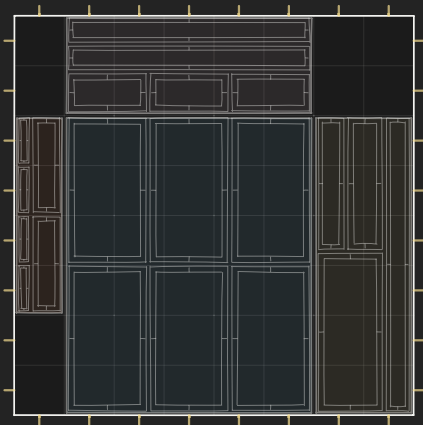
Dreaming and pruning

Schedule the [dream cycle](#) overnight and GBrain maintains links, repairs citations, and consolidates memory. gbrain think also calls out stale context, uncited claims, and contradictions.



Speaking and searching

Search blends [vector and keyword retrieval with rank fusion](#). Three search modes trade cost for quality, while gbrain think turns the results into a cited answer.



mem0

A developer memory layer that turns conversations into useful facts, then brings the right ones back when an app needs them.



Getting signals

Developers choose when to send a conversation through `add()`. mem0 uses an LLM to pull out the facts worth keeping, so ingestion starts inside the app rather than on a schedule.



Remembering

Memories live in a vector store and can be scoped to a user, agent, session, or organization. Entity extraction ships in open source, while the [interactive graph](#) lives on the hosted platform.



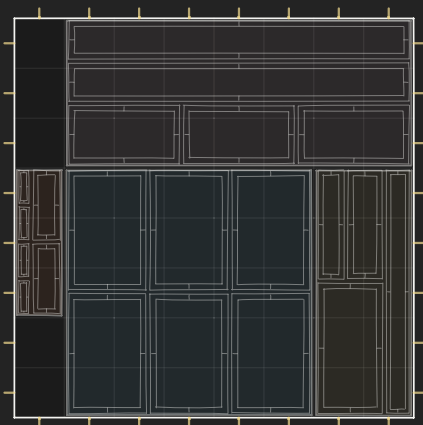
Dreaming and pruning

v3 keeps every write instead of updating or deleting old memories. At search time, [memory decay](#) gently favors recent, frequently used memories without hiding the rest.



Speaking and searching

When the app asks for context, mem0 combines semantic search with keyword and entity signals. Developers can use its SDKs or connect through the [hosted MCP server](#).



Letta

Letta lets the agent decide what belongs in memory, while a sleep-time agent works on the shared context in the background.



Getting signals

The agent can save and update memories through its own tool calls, while developers can write through the API. Native [Slack](#), [Telegram](#), and [GitHub integrations](#) can bring in outside context.



Remembering

Memory is split into [three layers](#): small labeled blocks kept in context, a searchable archive outside the context window, and the full message history. Letta stores them in Postgres.



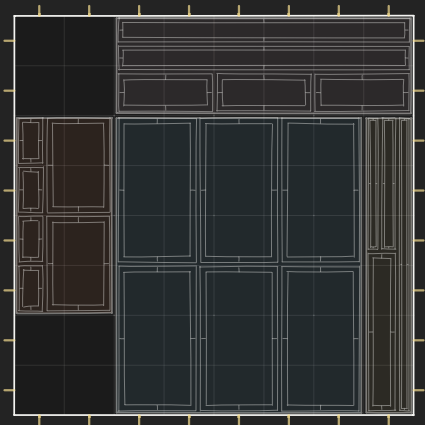
Dreaming and pruning

A sleep-time agent rewrites shared memory in the background. When the context fills up, [compaction](#) summarizes about 30% of the messages and keeps the rest.



Speaking and searching

The agent searches and updates memory through tool calls. People can inspect and edit it in the [Agent Development Environment](#), while developers can work through REST APIs and SDKs.



Zep / Graphiti

Zep keeps a temporal graph of people, events, and facts, marking old information as expired when something newer makes it outdated.



Getting signals

Conversations, JSON, and documents arrive as [episodes](#) through `add_episode()` or Zep's managed ingestion pipeline.



Remembering

Each episode feeds a [three-layer knowledge graph](#): raw events, extracted entities and relationships, and broader communities. Facts are embedded and time-stamped, so the graph remembers both what happened and when it was true.



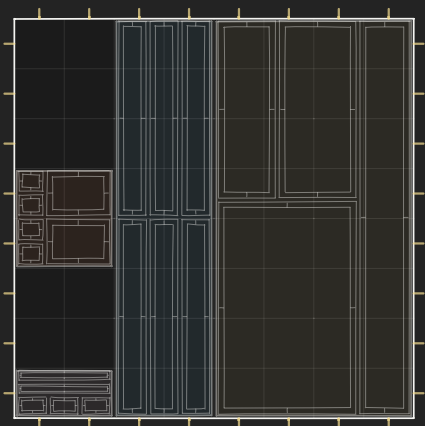
Dreaming and pruning

When new information contradicts an older fact, an LLM marks the old relationship as expired instead of deleting it. [Entropy-gated fuzzy matching](#) helps resolve likely duplicates without collapsing ambiguous names.



Speaking and searching

At query time, Zep combines [semantic search, keyword matching, and graph traversal](#). Because retrieval does not need an LLM, results can come back in well under a second.



Sylph

Sylph keeps the company brain in a Git repo you can read and shape yourself. Agents can draft improvements, but people decide what lands.



Getting signals

Sylph does not bundle outside data sources. You choose and configure the [MCP connectors](#) it can use, so prompts can pull live context from tools such as Slack or Gmail.



Remembering

The brain is the [Git tree itself](#): a central `CONTEXT.md`, domain folders for teams such as brand, sales, product, HR, and legal, and Markdown files the agents can read.



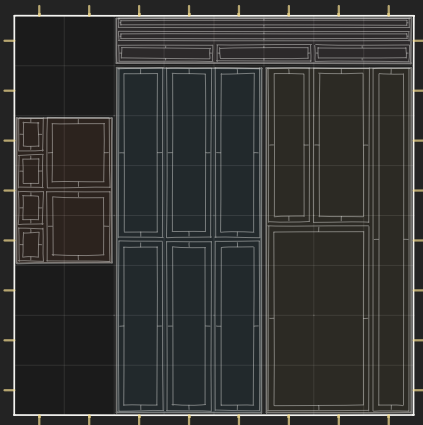
Dreaming and pruning

Agents place proposed improvements in `_drafts/` for human review. Once someone approves an output, [Sylph compares the draft with what was kept](#) and updates its own rules from the diff.



Speaking and searching

Teammates work through slash commands in Claude Code, Codex, or Cursor. Because the memory is plain Markdown in Git, people can inspect changes with familiar tools such as [git diff and grep](#).



DIY (Claude Code + git)

A company brain made of Markdown, pull requests, and plain-file search. It stays flexible because an engineer keeps it healthy.



Getting signals

Engineers add and update knowledge through pull requests. Subagents can research in parallel, then return compact summaries for the parent agent to save.



Remembering

The memory lives in the Git repo: CLAUDE.md, rules, skills, runbooks, and other Markdown files, all versioned with the rest of the system.



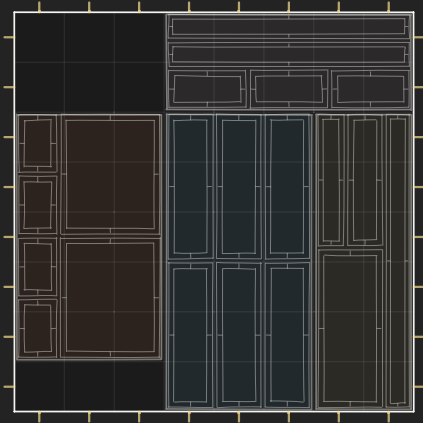
Dreaming and pruning

Session compaction is a command you run when context gets crowded. Longer-term pruning happens through human review, usually in pull requests.



Speaking and searching

Agents use read, glob, and grep to find context in plain files, loading skills only when needed. There is no semantic index.



Pletor

Pletor turns brand rules, creative references, and performance data into a shared memory its creative agents can use.



Getting signals

Brand nodes and data feeds bring in guidelines, references, performance data, and competitor signals. The system works across text, images, video, and audio.



Remembering

Pletor keeps those inputs in one living Brand Brain, connecting what the brand says, makes, and learns from performance.



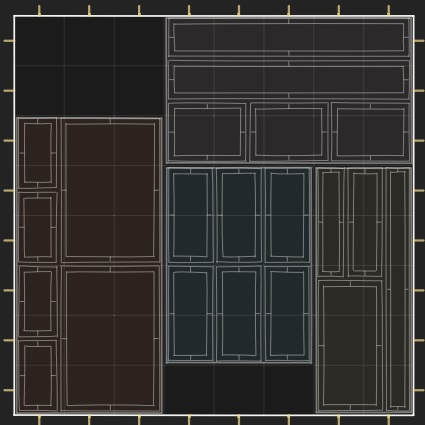
Dreaming and pruning

People stay responsible for taste and performance decisions, while feedback and results feed the next round of creative work.



Speaking and searching

People work with the brain through chat and voice agents, or build creative workflows on Pletor's canvas.



Gorgias Cortex

Gorgias built Cortex from the systems its teams already use, turning company knowledge into a graph its agents can navigate.



Getting signals

Airbyte and a custom ingestion tool bring in Postgres, Notion, Gong, GitHub, Linear, and HubSpot. Long transcripts are chunked and embedded so they stay searchable.



Remembering

The result is a knowledge graph of roughly 12,000 typed Markdown nodes in GitHub, connected by relationships the agent can follow.



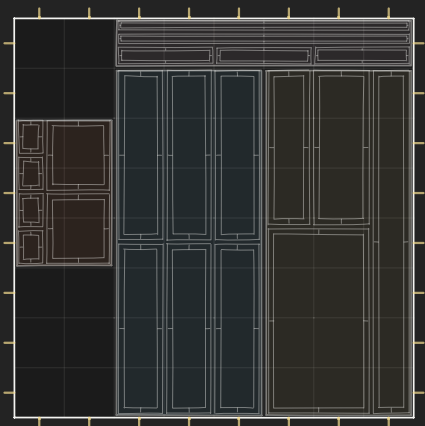
Dreaming and pruning

The team studies conversation traces to spot missing context. A scheduled maintenance loop groups recurring gaps into pull requests that improve the graph.



Speaking and searching

The agent starts with task-specific skills, then follows relationships through the graph until it has enough context to answer.



CA

Carrara

A cloud brain replicated into a native Mac app, with every memory approved by a person.



Getting signals

Every hour, Carrara pulls from Granola, Gmail, Calendar, and Slack, resolving each teammate across multiple email addresses. Composio makes it easier to add new sources.



Remembering

The shared memory lives in Neon Postgres, while a native macOS app replicates it locally for each teammate.



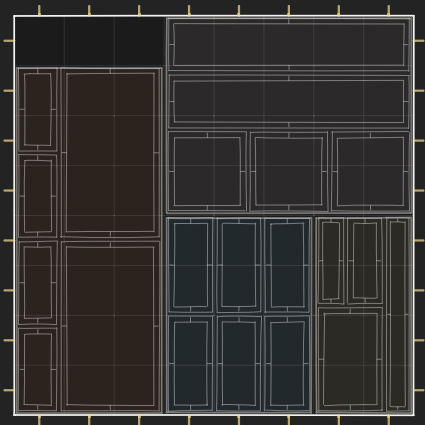
Dreaming and pruning

Nothing enters the brain without review. A dedicated acceptance interface keeps every memory human-approved; automated pruning is still to come.



Speaking and searching

Agents connect through a custom MCP server, while teammates use the native macOS app.



Slite Agent

Slite keeps the docs people already use, then adds review, expiry, maintenance, and cited answers around them.



Getting signals

Slite Agent live-syncs across [20+ sources](#), so it can bring the most current company context into every inference.



Remembering

People and agents can push context into Slite's knowledge base with [verification tags](#), owners, and expiry windows that make its status clear.



Dreaming and pruning

With [auto-maintenance](#), Slite regularly fact-checks your docs against the rest of your company data. Suggested updates appear as side-by-side diffs in a triage view, where you decide what changes.

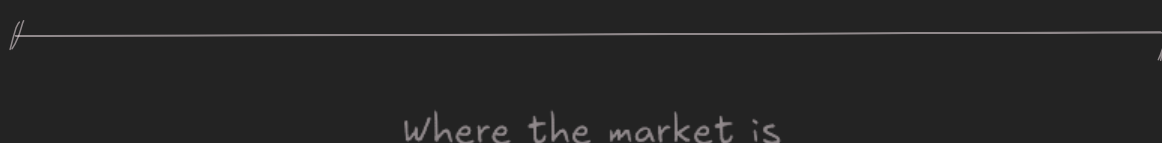


Speaking and searching

Slite Agent answers with citations in the app and API. Its [MCP server](#) lets other agents search, create, and maintain knowledge too.

On who's building, who got burned, what everyone fears, and which brain you actually need.

Beyond the hype, this is a lay of the land of how many people have tried a Company Brain, or actively evaluating one, and exactly what they're looking for.



Very few teams have a brain that actually runs. First, building a company brain that can serve your entire team is genuinely hard because you have to account for nuances that do not arise with a personal second brain. Second, teams are still deciding whether to build one in-house or buy one from the market.

Only 17% of people have a company brain that works.

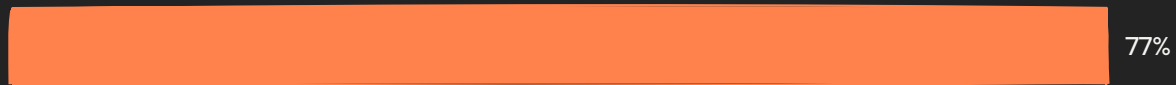
Among those who tried, half built custom setups on Claude or Codex wired to their knowledge base, and another one-fifth use [GBrain](#). But all the survey respondents who haven't adopted a company brain yet, or who failed to make one work, have tangible concerns.

The top 5 concerns about adopting a Company Brain

Whatever tool you pick, you are really picking answers to these worries.

If you're planning to build a Company Brain, or are already in the midst of buying one, it's worth tackling these concerns before deployment because they will naturally surface afterward.

Accuracy — how accurate are the answers



Privacy — how to manage permissions



Maintenance burden



Cost of daily ingestion and maintenance



Setup effort



slite Company Brain Survey 2026

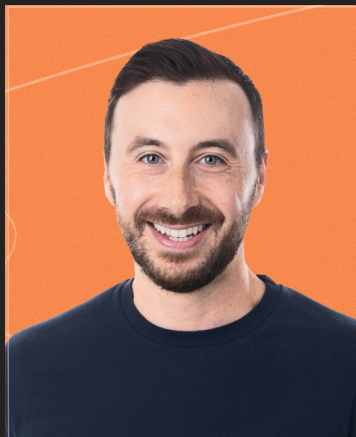
One overlooked factor - Context sovereignty

A company brain has to read across every source.

That means whoever runs it holds the most complete, portable version of your company that has ever existed in one place.

This potentially creates the largest context lock-in vulnerability in your tool stack.

SaaS platforms are building their own brains on the slice of context they already hold and squeezing everyone else's access to it. As of mid-2025, third-party tools can no longer bulk-index Slack messages, and non-Marketplace apps are rate-limited and can't store historical data long-term.

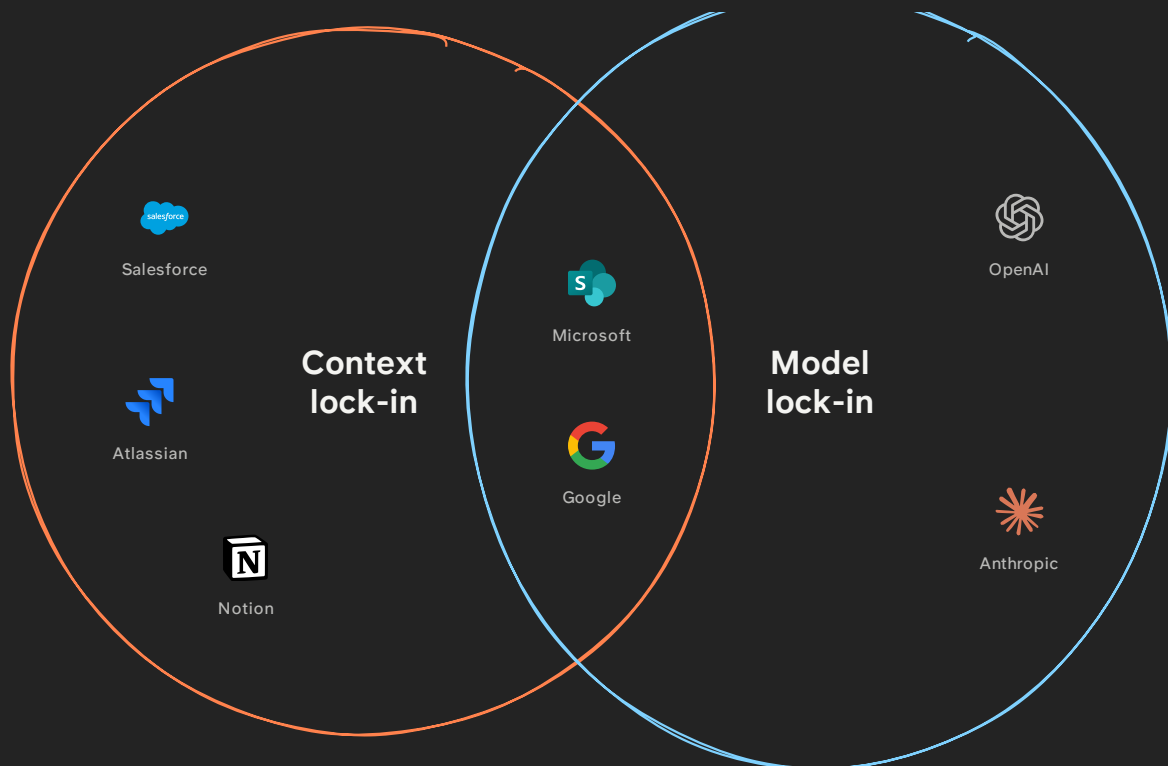


“They'll squeeze the ecosystem with rate limits, clunky sandboxes, higher access fees, opaque review processes, and stricter gating behind marketplace policies. This is API war by attrition, not revocation.”

Marc Andrusko

Partner, a16z, The API Battleground, August 2025

The frontier labs want the same prize from the other side, and each player locks you in from the side it already controls.



That's why, whether you're buying or building a Company Brain, one requirement should be non-negotiable:

Treat context sovereignty as a requirement, whoever you buy from and whatever you build.

Context lock-in compounds quietly. You pay it in every answer, every agent, and every workflow built on top of it.

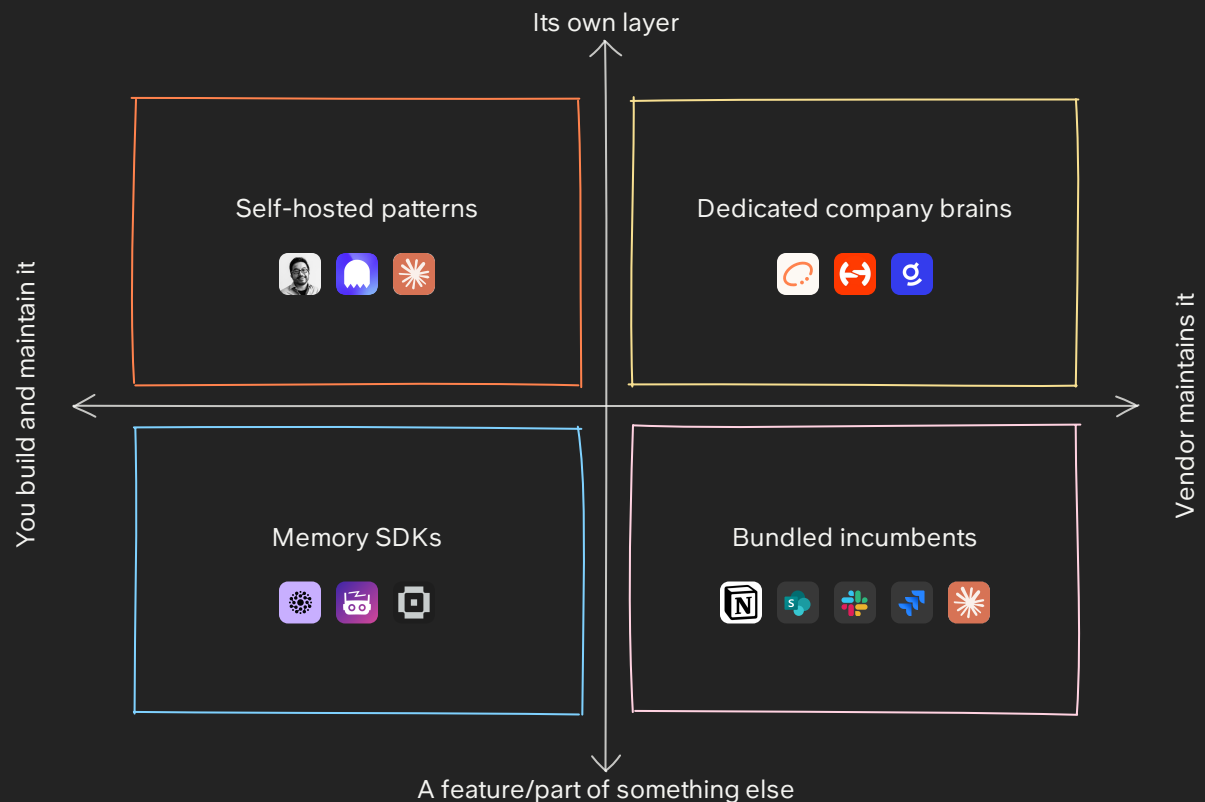
That is why the company brain has to be its own layer. It should sit above the platforms and the models, so it owes loyalty to neither.

The map of current Company Brain solutions

Every harness on the market makes different trade-offs across the five biggest user concerns. The cleanest way to see them is to ask who runs the upkeep and whether the brain is its own layer or a feature of something else.

The map splits into four families.

Where the market is



Self-hosted patterns

They give you full control and modality to build around your data infrastructure. The trade-off is that your engineering team owns the system maintenance for accuracy, model routing, and cost improvements.

Dedicated company brains

These vendors are building the company brain as a standalone product. They connect to your existing sources, handle architecture maintenance, improvements, and serve context as a graph or instant answers via UI/MCP.

Memory SDKs

They give developers the building blocks to add memory to the product or agent they are building. That memory stays tied to the product's users, sessions, and workflows, rather than becoming shared context for the whole company.

Bundled incumbents

They add a brain-like layer to a suite your team already uses, which makes them easy to switch on. But they are designed to make that suite smarter, not to create an independent company brain across your stack.

Which one do you need?

How big is your team?

1 – 50

51 – 200

200+

What worries you most about adopting one?

Wrong answers

Who gets to see what

Setup and upkeep

What it costs to run

Who would own it internally?

Technical

Not technical

Dedicated company brains



Memory SDKs



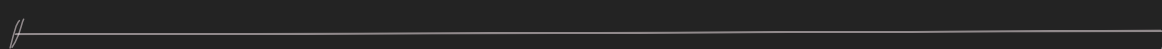
Self-hosted



One set of answers of many. Match your own at slite.com/ebooks/company-brain →

On the takes every builder shares, and the questions none of them can answer.

A consolidation of which parts the vendors have consensus on, and what they don't.



what the builders see coming

We talked to several builders for this book. They disagree on almost everything: retrieval strategies, graph vs. no graph, and how much to automate. But on four things, every single one of them landed in the same place.

You'll maintain the Company Brain forever

Half the builders in our survey custom-built a Company Brain on Claude or Codex, and roughly nine in ten of Slite's enterprise pilots had already built something in-house before coming to Slite.

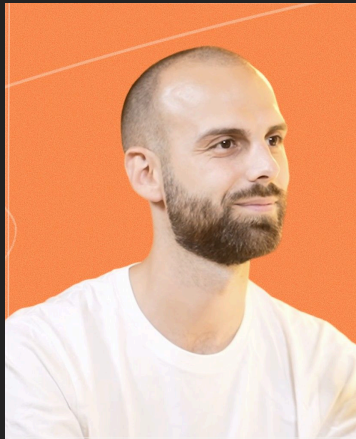
What kills these brains is maintenance: most DIYers walk away from working brains because of the upkeep.

A big chunk of the upkeep comes down to the fact that the company brain field refuses to settle.

- Retrieval is still an open argument. The tools we studied run five different rerankers and three different search philosophies, with no shared benchmark.
- There isn't even consensus on whether a brain needs a context graph. Zep bet its whole architecture on temporal graphs, Gorgias hand-built one with 12,000 nodes, and Slite and Letta ship no graph at all.

Every component of the architecture chapter is still being tinkered with in public, month after month.

We spoke to Yochan Khoi, who helped build Gorgias's internal Company Brain, Cortex. The team's architecture article went out of date between its publication on [Medium](#) and our interview.



“ The thing that has the most value... is really how you maintain a knowledge base, how you structure it, how you debug it. ”

Yochan Khoi
Staff Context Engineer, Gorgias

Gorgias can afford to build and maintain Cortex with an internal AI team. For a small or mid-sized team, ongoing maintenance is the job nobody budgets for. It turns the DIY brain from a weekend of Markdown files into an ongoing product commitment in a market that hasn't picked its standards yet.

Building in-house can make sense for very large companies with unusual requirements and an internal team ready to own the system as a permanent product. Does your company want to maintain ingestion, retrieval, permissions, evaluation, and governance indefinitely? Gorgias can. Most teams, including many large ones, would rather offload that responsibility.

AI is not ready to write autonomously to your Knowledge Base

AI Agents can't autonomously declare what's true about your knowledge. Auto-created knowledge works well in personal brains because you have no conflicts with yourself.

However, company context is constantly moving, and Agents struggle to distinguish between a scoped plan and a long, authoritative Slack conversation about a potential change.

That is why most builders are defaulting to human triage before adding new context.

- Pletor is building an inbox for brain amendments.
- Gorgias routes every graph fix through a reviewed PR.
- Sylph's agents can't promote their own drafts.
- At Slite, we built an Agent Triage feature because we knew shipping the Slite Agent without it could be dangerous.



“ The promise of having an open Claude that learns from your Slack and just runs things is sexy. But it's quite hard to execute. ”

Maxime
Founder, Pletor

Hopefully, none of this is a permanent verdict.

Models will keep improving, and someone may yet land a framework that reliably handles tacit, unstructured knowledge. Every gate the builders use today is built to open as that happens. Until models can judge company context reliably, human review remains the simplest way to stop bad information from compounding.

Capture has to happen at the place of work

We spoke to enthusiastic knowledge base users, the people most committed to their company's brain. We repeatedly heard that they couldn't keep feeding it either.



“ You have to be very purposeful about feeding the machine, and even with the best of intentions from me trying to drive that thing, I wasn't doing it either. ”

Glen Sykes
Chief Product Officer, Spoor

That is why builders are no longer trying to fix human discipline. They're moving capture into the work itself.

- Claire Gouze, co-founder at [nao Labs](#), ends every agent skill with a logging step, so her brain updates as a side effect of getting things done.
- Yochan Khoi's team at [Gorgias](#) mines its own conversation traces nightly for what the knowledge graph is missing.
- [Slite Agent](#) watches activity in your connected tools rather than waiting for anyone to write things down.

Capture only survives when it rides along with work that is happening anyway, invisibly and at the moment the knowledge is created. Any brain that depends on someone remembering to feed it starves on schedule.

People will keep asking the brain questions. Agents will ask more.

Your search bar and the chat window aren't going anywhere. Someone will always need the refund policy before a call.

But see who else has started asking.

mem0, Guru, and Slite all shipped MCP servers this year, while Gorgias is exposing one for teams that want to stay code-first.

Builders we interviewed already wire the brain into their agent routines as a context step rather than visiting it as a destination. An agent checks context before every task at machine frequency. The moment a company runs real agent workflows, agent queries outnumber human ones by orders of magnitude.

What nobody can answer yet

The same builders who agreed on everything above didn't reach a clear consensus on some open questions for which the way forward isn't clear-cut.

How does context from personal AI chats reach the brain?

More work is moving into Claude and ChatGPT sessions, but very little of that context reaches the company brain. How much of that work should go back into it?

“The only place where some kind of work is done but is not connected remains ChatGPT or Claude. All these tools are being connected, except these ones. A lot of the work is now being done there, versus in Slack or in Notion. It actually takes market share from messaging and knowledge bases.”



Maxime
Founder, Pletor

Your colleagues shouldn't know you were arguing with Opus at 2am. But prompts are the most distilled version of your thinking. Having them in the company brain is the clearest way to show AI how you actually reason about strategic and creative problems.

When two versions conflict, whose truth does the brain keep?

AI collapsed the cost of producing plausible-looking knowledge, so the brain now needs an opinion about whether the CMO's positioning or the intern's summary wins. Every mechanism the builders use today reduces to one trusted human, who becomes both the bottleneck and the single point of failure.



“ The junior who joined four weeks ago decides to write a lengthy report on a subject matter he doesn't know anything about. He saves it, and this is all fine because no one finds this page as long as there is no AI. But all of a sudden, AI finds this single dump page and draws very strange conclusions from it. Two years ago, he generated those reports once a week. These days, he generates two a day. ”

Florian Redeker
Product Manager, AllUnity

When does the brain get to write itself?

Every builder we talked to has a human in the loop before anything reaches the brain. Gorgias routes every graph fix through a reviewed PR. But none of them think this is permanent.

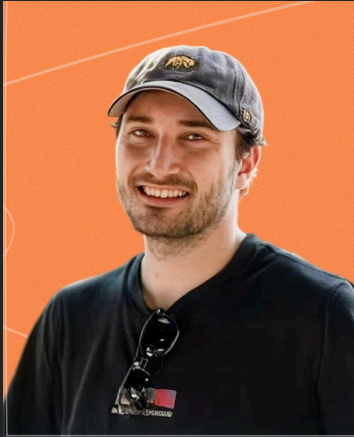
The question they're all circling is the same: at what point do you trust the model enough to let it write without asking?

What makes this harder than it sounds is that company context is political. Two people can hold contradictory versions of a decision and both be right about their slice of it. A model that auto-merges the louder version is quietly taking sides, and the brain ends up with one confident, incomplete story where there used to be a live disagreement.

How do you get the 80% that was never written down?

Every brain we looked at starts from the same assumption: the knowledge already exists somewhere in your stack, and the job is to connect it.

Leon Iwanowitsch, co-founder of Ontora, thinks that assumption misses most of the picture.



“80% of operational knowledge is still tacit, living inside people's heads.”

Leon Iwanowitsch
Co-founder, Ontora

Docs are the residue a decision leaves behind, but they do not always capture the reasoning behind it. The senior engineer knows something about the codebase that they've never had a reason to write down, and no connector will ever reach it because it was never in any tool to begin with.

The builders are starting to treat this as an input problem. Ontora's answer is to use structured, AI-led interviews that pull tacit knowledge out of employees' heads and feed it into the brain. Nobody has cracked how to do that at scale yet.

A note from Chris

When we started Slite ten years ago, I resisted the urge to call it a knowledge base with all my might.

The phrase felt like a lie. As if the problem of sharing, maintaining, and finding knowledge as a team was solved. It was clear to us that keeping knowledge useful, trustworthy, and current would be a challenge for the next decade.

We solved part of it by making the product so fast and natural to use that people were far more likely to add and fix knowledge themselves. We made search radically faster and more accurate. We also built on that with strong verification signals and improved review experiences.

But no matter how close we got, the root causes of context sprawl and humans being too busy to babysit documentation kept putting distance between people and the insights they needed during everyday work.

When ChatGPT launched back in 2022, everything changed. Finally, a proper single source of truth for teams was within arm's reach.

We shipped the world's first AI search for a knowledge base two months later, and we've been building a Company Brain ever since. We've already solved some of the hardest problems around retrieval and knowledge maintenance, but as you saw above, there are still so many questions to answer.

I went into these conversations with builders hoping that some of them had answers. Instead, I found them wrestling with the same questions. And that felt oddly reassuring because their uncertainty was matched by an enormous amount of care. They spoke openly about what had failed, borrowed ideas from one another, and treated disagreement as a way to improve the work.

Those exchanges left me more excited about the space than I was when we began.

After ten years in this category, that sense of shared effort means a lot to me. The hard questions remain hard, but none of us has to answer them alone.

If we get this right, a company brain will give people back the time and attention they now spend searching for and checking context. Work should come with a little less frustration and a little more joy.



Chris

Co-founder and CEO, Slite

slite

This is the print companion to the interactive edition — the live market map, vendor deep dives, and survey explorations are at

slite.com/ebooks/company-brain