



WHITEPAPER • FINANCIAL SERVICES

Building Trusted AI in Financial Services

A Guide to Knowledge Accuracy,
Governance, and Compliance

TL;DR

0.0

Modern financial services companies have outgrown the knowledge base their support CRM came with. A single customer intent, “I can’t log in,” is no longer one question with one answer. At a global digital bank, it’s 150+ different answers depending on jurisdiction, product, verification state, channel, and customer segment.

At that scale, wrong answers stop being occasional mistakes and become a statistical certainty, unless accuracy is engineered into the system itself.



In financial services, a wrong answer is not just a bad support interaction. **It is regulatory exposure.**

AI raises these stakes rather than lowering them. Bolting AI onto a stale knowledge base doesn’t fix bad knowledge, it distributes bad knowledge faster, more confidently, and to more surfaces. Traditional knowledge bases have no machinery for staying accurate: no way to detect when content goes stale, no way to find gaps before customers do, and no visibility into what your AI is actually telling people.

This paper describes the three problems that machinery has to solve, how Guru solves them with automated verification, an agentic knowledge base, and the AI Agent Center, and why together they form a loop where accuracy compounds instead of decays. It closes with how companies get there quickly, whether that means migrating the knowledge you have or assembling the knowledge that was never written down, because getting there is Guru’s project, not yours.



A note on sources. The patterns and impact data in this piece are drawn from detailed work with five financial services companies running these programs in production today. Out of respect for their commercial sensitivities, we describe their work in aggregate rather than naming them. The numbers, operational practices, and outcomes described here are real, not hypothetical, and represent what’s actually working at the leading edge of fintech support.

Section One

One Question, Thousands of Answers	04
1.0 A Login Problem Is Not a Login Problem	05
1.1 Fifty-One Versions of Every Answer	08

Section Two

Your Knowledge Now Has Two Audiences	10
2.0 AI Removed the Filter	11
2.1 Plausible Is Not Correct	12

Section Three

The Three Problems	13
3.0 Problem One: Knowledge Rots Silently	14
3.1 Problem Two: Your Customers Find the Gaps Before You Do	16
3.2 Problem Three: You Can't Improve What You Can't See	18

Section Four

The Loop	20
4.0 Where Accuracy Compounds Instead of Decays	21

Section Five

Getting There Is Our Project, Not Yours	23
5.0 Two Starting Points, One Principle	24
5.1 A Partner, Not a Toolkit	26

Section Six

In Production	27
6.0 What This Looks Like in Production	28
6.1 Three Questions for Your Organization	29



One Question, Thousands of Answers

One Question, Thousands of Answers

1.0

A Login Problem Is Not a Login Problem

A customer at a modern digital bank types five words into a chat window: “I can’t log in.”

To the customer, it’s one question. To the support team, it’s a fork in a decision tree that branches at least five times before anyone can answer.

- **Which jurisdiction is this customer in?** Germany under BaFin, Brazil under the Central Bank, the UK under the FCA, Singapore under MAS, the EU under PSD2? Each has its own rules for authentication, identity verification, and recovery.
- **Which product are they using?** Personal account, business account, asset account, currency wallet? Each product has its own auth flow, its own permission model, its own escalation path.
- **What’s their verification state?** Fully KYC’d, partially verified, pending review, flagged for enhanced due diligence?
- **Which channel are they on?** Mobile app, web, the new biometric flow that just shipped last week?

By the time the agent has enough context to answer, the “one” login question has become 150 possible answers. The right answer for a German personal customer with partial verification is completely different from the right answer for a Brazilian business customer with full verification, even though both customers typed the same words.

One Intent, 150+ Answers

The same five words, branching before anyone can answer

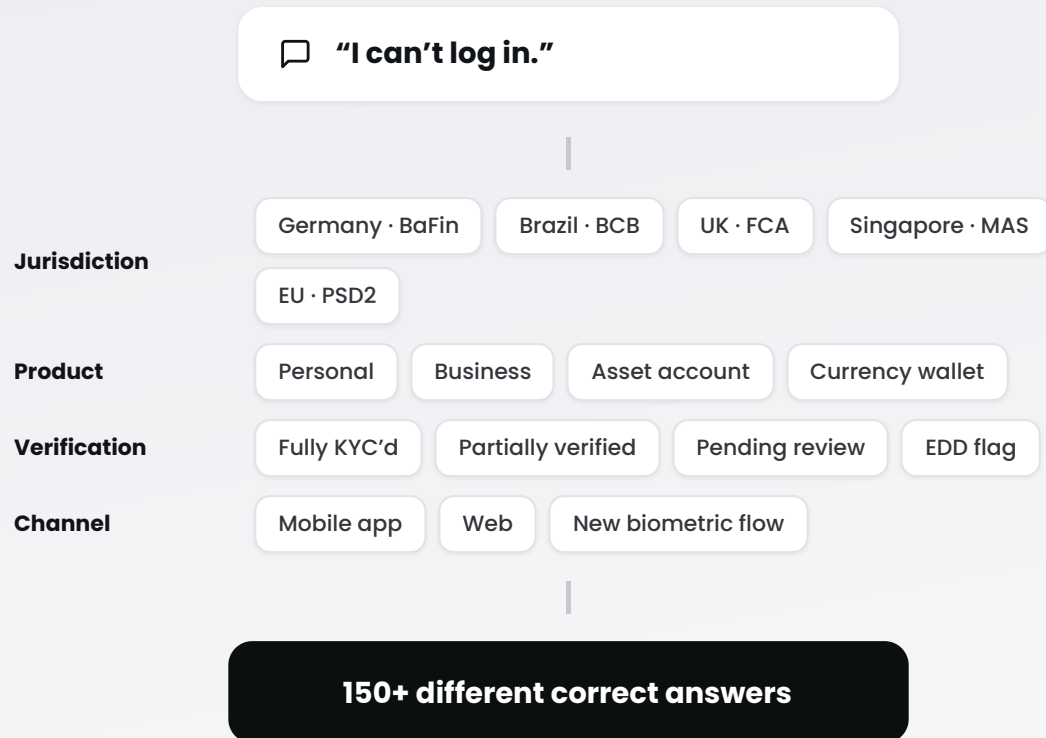


Figure 1. One intent, 150+ answers. The same five words require a different correct answer for each combination of jurisdiction, product, verification state, and channel.

This is not a one-company problem. It is the structural condition of every financial services company that has scaled across products, regions, and regulatory regimes faster than the support tooling around them could keep up. At one global fintech we work with, more than 100 autonomous product teams ship updates every week. The customer support knowledge base sees 200 to 300 changes per week from CS alone.



A CS leader at a global fintech, describing the result:

Ungovernable knowledge debt.

Now consider what that means for accuracy. When 150 possible answers each depend on a regulation, a product detail, and a policy that all change independently, some fraction of your knowledge base is wrong at any given moment. The only question is how large that fraction is, how fast it grows, and whether anything in your system is built to catch it.

In most support organizations, the honest answer is that nothing is. Across organizations generally, only 8 to 12 percent of content ever gets reviewed by a human. The other 90 percent sits ungoverned, aging quietly.

150+

possible answers behind
one login question at a
global digital bank

200–300

knowledge base changes
per week from CS alone at
one global fintech

8–12%

of content ever gets
reviewed by a human; the
rest ages ungoverned

In most industries, a stale article produces a frustrating ticket. In this industry, a wrong answer about a BaFin authentication rule, a KYC requirement, or a fee schedule is a compliance event, with remediation, reporting, and regulator conversations attached.



Accuracy is not a quality metric for financial services support. **It is the whole point.**

Fifty-One Versions of Every Answer

Insurance runs the same math, with a different multiplier. A policyholder types a question that sounds even simpler: “When will my claim be paid?”

In the United States, insurance is regulated state by state, which means an insurer operating nationally is not answering under one set of rules. It is answering under fifty-one of them, each with its own commissioner, its own statutes, and its own clocks. How quickly a claim must be acknowledged, how often the policyholder must receive a status update, how long the insurer has to pay once a claim is settled, what notice is required before a policy can be cancelled or non-renewed: every state sets its own numbers, and few of them match.

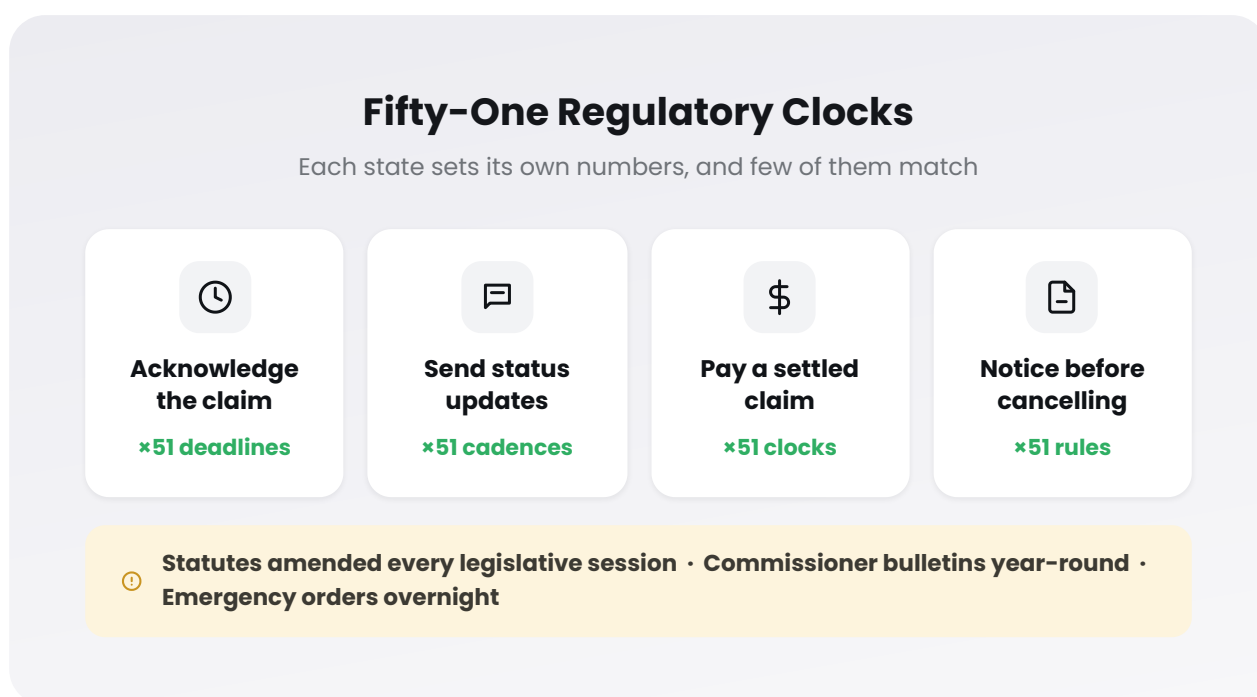


Figure 2. Fifty-one regulatory clocks. Every legislative session amends the statutes somewhere, commissioners issue bulletins year-round, and the support team is expected to have the new answer the same day.

Then the branching continues. Which line of business, auto, home, renters, life? Auto alone splits between no-fault states with personal injury protection requirements and tort states, where the correct guidance after an accident is fundamentally different. Is this a routine claim or a catastrophe claim? Coastal states layer hurricane deductibles on top of standard ones, and when a governor declares a state of emergency, the insurance commissioner typically issues bulletins that suspend claims deadlines, freeze cancellations and non-renewals, and extend grace periods, often overnight, often for a specific list of counties, and often for a defined window that later gets extended.

So the “one” question about a claim payment becomes thousands of jurisdiction-dependent answers, and the set is never stable. Every legislative session amends the statutes somewhere. Commissioners issue bulletins year-round. A single hurricane can change the correct answer in three states in one afternoon, and the support team is expected to have the new answer that same day, because the customers asking are having the worst week of their year: a flooded kitchen, a totaled car, a house that isn’t livable.

The stakes carry the same regulatory weight as banking, with a sharper edge. An answer that misstates a claims deadline or handles a policyholder on last quarter’s rules isn’t just a bad experience. It can constitute an unfair claims practice, the kind of thing that surfaces in market conduct exams and bad-faith litigation.

□ Whether it’s a neobank’s login flow or an insurer’s claim clock, the shape of the problem is identical: **one simple question, an explosion of jurisdiction-dependent answers, and a body of knowledge that changes faster than any team can review by hand.**

Your Knowledge Now Has Two Audiences



Your Knowledge Now Has Two Audiences

2.0

AI Removed the Filter

Until recently, company knowledge had one audience: people. And people always had a backdoor around bad knowledge. When the article looked stale, they shoulder-tapped the person who actually knew. A wrong document was an annoyance, because a human filter stood between it and the customer.

AI removed the filter. Agents read whatever is in front of them and say it with total confidence, then act on it. Twenty-seven AI agents quoting a retired fee schedule is a different kind of problem than one confused new hire, and it is the problem every fintech deploying AI in support is now exposed to.

The Filter Is Gone

One audience: people

A stale article looks wrong, so someone **shoulder-taps the person who actually knew**

A wrong document is **an annoyance**

Cost of being wrong: **one confused new hire**

Two audiences: people and AI

Agents read whatever is in front of them and **say it with total confidence, then act on it**

A wrong document is **distributed across every AI surface**

Cost of being wrong: **27 AI agents quoting a retired fee schedule**

Figure 3. The human filter stood between bad knowledge and the customer. AI reads whatever is in front of it, says it with total confidence, and acts on it.

The most common first move when support complexity hits the wall is to throw an LLM at the existing knowledge base. Take the KB as it stands, put a model on top, expose it through agent assist or chat, and hope the AI synthesizes its way out of the mess.

What actually happens is that the AI inherits the mess. If the underlying knowledge is fragmented, contradictory, or stale, the model doesn't repair it. It delivers the stale answer faster, in more channels, with more confidence.

In a blind test one global fintech ran during their AI tooling evaluation, a competitor product returned critically inaccurate answers on 66 percent of test questions, compared to 16 percent for the approach described in this paper. The model wasn't the bottleneck. The knowledge underneath it was.

66%

of test questions returned critically inaccurate answers with a competitor product, in a global fintech's blind evaluation

16%

with the approach described in this paper. The model wasn't the bottleneck. The knowledge underneath it was.

2.1

Plausible Is Not Correct

The deeper issue is that retrieval was never designed to care whether an answer is still true. RAG, enterprise search, and MCP connectors were each real steps forward, and each one ignores state. Semantic search returns something plausible. Your business needs something correct.

"Find yesterday's meeting transcript" works fine, because the thing exists and retrieval finds it. "How do we handle this issue for a UK customer on our business edition" falls apart, because the right answer changed last week and nobody updated the knowledge base, so retrieval returns the old answer, confidently.



Every dollar you spend on agent assist, chat deflection, self-service, and voice AI **inherits the accuracy of the knowledge layer underneath it.**

Which raises the real question: what would a knowledge system built to stay accurate actually have to do? Three problems have to be solved, and traditional knowledge bases solve none of them.



The Three Problems

What would a knowledge system built to stay accurate actually have to do? Three problems have to be solved, and traditional knowledge bases solve none of them.

The Three Problems

3.0

Problem One: Knowledge Rots Silently

No article knows it's wrong. A regulation changes in one market, a product team renames a flow, a fee schedule gets updated, and the twelve articles that reference the old reality keep sitting there looking exactly as authoritative as they did the day they were written.

The teams we work with have a word for this: **drift**. Your team operates in one reality, and your knowledge describes another. Humans move on and tell each other. Your AI is still reading what got written down last quarter, and acting on it.

Traditional knowledge bases have no mechanism for detecting drift. Review is manual, so it depends on someone finding time, and the volume guarantees no one ever will. That is how you end up with the 90 percent ungoverned figure: not negligence, just arithmetic. The knowledge decays by default, and the first person to notice is usually a customer, an auditor, or an AI agent confidently serving the wrong answer.

How Guru solves it

Guru makes accuracy a property of the system instead of a hope. Every piece of knowledge carries a **verification state, a named owner, and a verification date**, and this extends beyond Guru itself to any connected source, including the document drives, wikis, and shared folders where a fintech's policy documents actually live.

Knowledge Agents then automate the review work itself, verifying and unverified content continuously based on rules you configure: **behavioral rules** built on how content actually performs (feedback, usage, engagement) and **content rules** built on what it says (flag anything referencing a deprecated flow, anything touching a specific regulation, anything untouched in 90 days). The agents run continuously, report every action they take, and record every decision in a quality log your team can inspect and override.

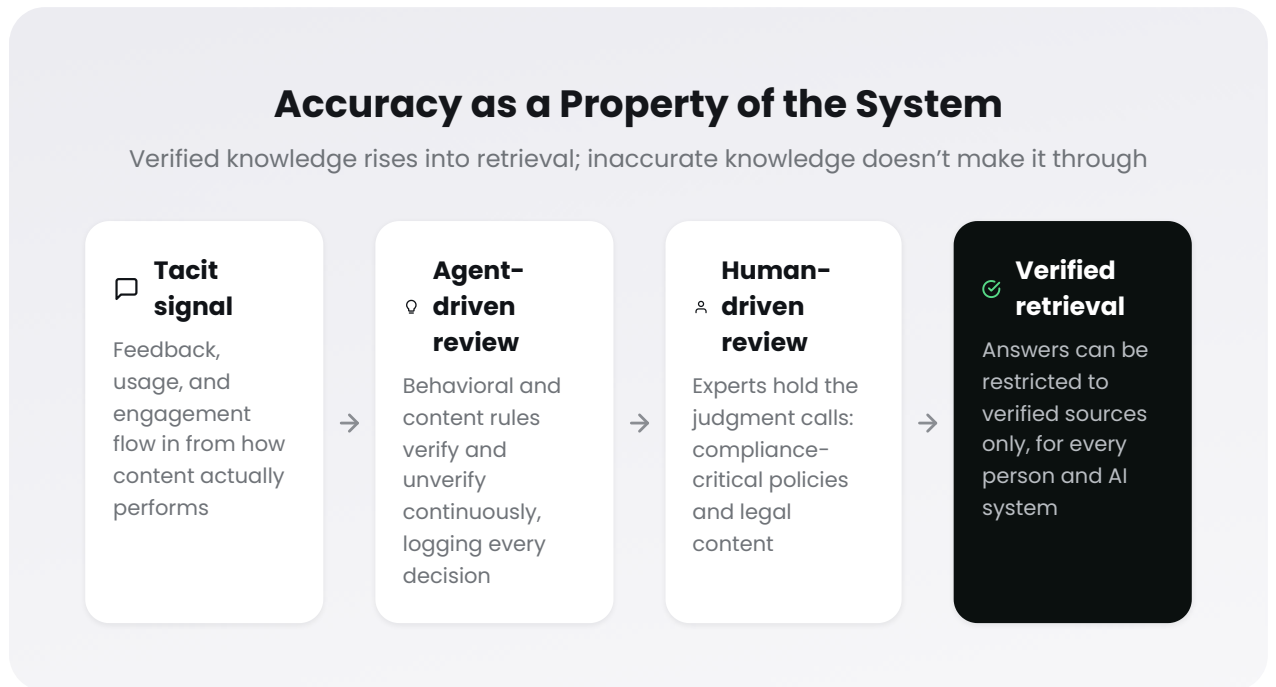


Figure 4. Accuracy as a property of the system. Three mechanisms keep the loop running: tacit signal, agent-driven review, and human-driven review.

Two details matter most for this vertical.

First, you can restrict answers to verified sources only, so nothing your agents or your AI say to a customer comes from ungoverned content.

Second, the audit trail is not a document you assemble after the fact when a regulator asks. It is the system itself: every article, its owner, its verification date, and every automated decision, inspectable at any time.

Your experts stay focused on the high-stakes judgment calls, compliance-critical policies and legal content, while the automation handles the routine review volume no team could ever staff for.

Problem Two: Your Customers Find the Gaps Before You Do

The second problem is what never got written down at all. The real answer to a recurring question lives in a Slack thread where a decision got made, a support ticket that keeps coming back, a call where someone explained how the process actually works.

The person who has that knowledge is busy, nobody owns capturing it, and knowledge teams are reactive by construction: the gap gets discovered at the worst possible moment, mid-ticket, when a customer is waiting and the answer isn't there. Meanwhile the backlog of missing and outdated content grows faster than any team can write, because it used to feel free to ask people to document things by hand, and it never was.

How Guru solves it

Guru's Knowledge Agents close this gap by doing the discovery themselves. Think of the colleague who somehow sat in every meeting, read every ticket, and followed every thread, and every so often said "we never wrote that down, but we need to."

The agent reads the places knowledge actually lives, work chats, meeting recordings, support tickets, project tools, and surfaces what should be captured or updated. Then it does the part nobody has time for: **it writes the draft, picks the right expert, and routes it to them for review.** It also notices contradictions, when a decision in a thread conflicts with a published article, and drafts the correction with the change clearly shown.



Your expert reviews a finished draft or a diff, never a blank page. **An afternoon of documentation work becomes a two-minute review.**

Discovery Without Surveillance

The agent proposes and a human approves. Nothing publishes on its own unless you choose that.

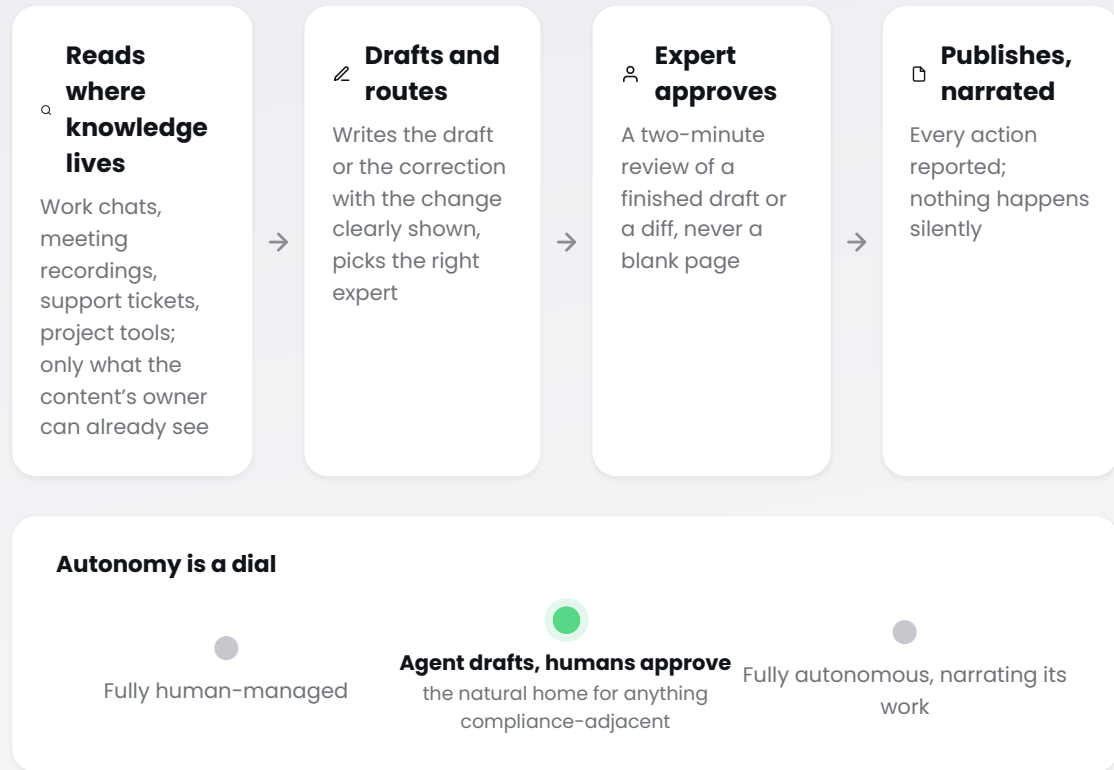


Figure 5. Discovery without surveillance. The agent proposes and a human approves, it is permission-aware, and it narrates everything it does.

You control how much of this runs on its own. The autonomy is a dial, from fully human-managed, to agent-drafts-and-humans-approve, to fully autonomous with the agent narrating its work. For a financial services team, the middle setting is the natural home for anything compliance-adjacent: the agent does the noticing, drafting, and chasing, and your domain experts hold the approval.

Three things make this safe for a regulated environment, and they are worth stating plainly. **The agent proposes and a human approves**, so nothing publishes on its own unless you choose that. **It is permission-aware**, so it only reads what the content's owner can already see. And **it narrates everything it does**, so nothing happens silently. Discovery here means your knowledge base notices what changed before you do. It does not mean surveillance.

Problem Three: You Can't Improve What You Can't See

Once AI starts answering questions, for your customers, your agents, and increasingly for other AI systems, most organizations have zero visibility into what was asked, what was answered, and where the answers were wrong. Every one of those answers is your company speaking, and in this industry that makes invisibility two problems at once.

It is **an improvement problem**, because a system that can't see its own mistakes can only get worse over time. And it is **an auditability problem**, because “what did your AI tell customers about this policy” is a question a financial services company will eventually be asked, by a regulator, an auditor, or its own legal team, and “we don't know” is not an acceptable answer.

How Guru solves it

Guru's AI Agent Center captures every question asked of a knowledge agent, whether it came from a person or another AI, along with the answer given and its status: answered, flagged inaccurate, or unanswered. That record is your audit surface. It is also fuel.

The knowledge agent reads its own record back and acts on it: unanswered questions spotlight knowledge gaps, so the agent drafts the missing content; flagged answers point to stale knowledge, so the agent drafts the fix. Your experts approve every change, and every change is versioned and reversible.

Every Answer, on the Record

Answered, flagged inaccurate, or unanswered — and every status acted on

"What's the recovery flow for a partially verified UK customer?"

Answered

Knowledge confirmed by use

"Does the fee schedule apply to business accounts?"

Flagged inaccurate

→ agent drafts the fix from stale knowledge

"How do we handle EDD-flagged asset accounts?"

Unanswered

→ gap spotlighted, agent drafts the missing content

Your experts approve every change · every change is versioned and reversible

Figure 6. Visibility as audit surface and fuel. Improvement here is inspectable, versioned, and approved by your experts.

To be precise about what improves: the agent is not retraining a model. It is improving the knowledge the model runs on, and that improvement is inspectable, versioned, and approved by your people. The practical consequence is a system that inverts the usual trajectory. Most AI deployments are most accurate on launch day and degrade as the business drifts away from the knowledge underneath them. **This one gets more accurate the more your people and your agents use it**, because every interaction either confirms the knowledge or improves it.

The Loop

These are not three features that happen to ship in the same product. They are one continuous loop that a knowledge agent runs end to end.



The Loop

4.0

Where Accuracy Compounds Instead of Decays

A knowledge agent runs the loop end to end: **discover** what should be captured, **draft** it and route it to the right expert, **organize** collections as products and teams change, **verify** continuously that published knowledge is still true, and **retire** what has gone stale. Humans direct the loop and approve what matters; agents do the work.

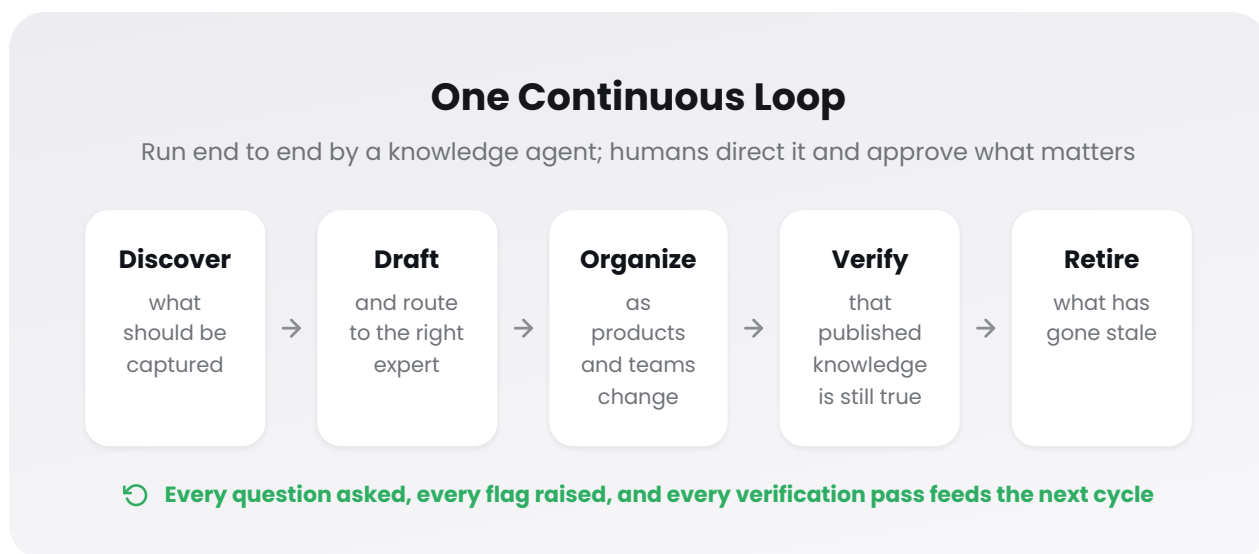


Figure 7. One continuous loop. Every downstream consumer, human or AI, inherits the improvement at once.

The loop is the answer to the arithmetic from the beginning of this paper. A knowledge base with no loop can only decay, because change accumulates and review capacity doesn't. A knowledge base with the loop compounds, because every question asked, every flag raised, and every verification pass makes the underlying knowledge better, and every downstream consumer, human or AI, inherits the improvement at once.

One challenger bank we work with runs this loop deliberately, feeding 20 to 30 targeted corrections per week from frontline agents back into the knowledge.

20–30

targeted corrections per week, fed back into the knowledge by frontline agents

98 → 100%

chat QA success rate over the course of the program

0

model changes. The models never changed. The knowledge underneath them did.



Over the course of the program, chat QA success rates moved from 98 percent to 100 percent. **The models never changed. The knowledge underneath them did.**

Getting There Is Our Project, Not Yours

Getting There Is Our Project, Not Yours

5.0

Two Starting Points, One Principle

Every CX leader in this vertical has scar tissue from a knowledge migration. Thousands of articles in the help center that came bundled with the ticketing system, a wiki graveyard nobody has opened in a year, content that needs compliance review before anyone touches it, and no one on the team with a quarter to spare. The fear of that lift is the single biggest reason support teams stay on a knowledge base they already know is failing them. **The migration feels like the risk, so the decaying status quo wins by default.**

Plenty of teams face the opposite starting condition, and it's just as paralyzing. The knowledge was never really documented in the first place, or what exists is so outdated and fragmented that nobody trusts it, let alone wants to move it. The real answers live in a Slack channel where senior agents field the same questions every week, in ten thousand resolved tickets, and in the heads of a few tenured people. For these teams, the barrier isn't "we'd have to migrate it all." It's "we'd have to write it all first," which feels even worse.

□ Both starting points run on the same principle as everything above: **AI does the heavy lifting, and your people direct it.**

If you have content worth moving

Our migration tooling uses AI to ingest your existing knowledge, restructure it, and flag what is stale, duplicated, or contradictory on the way in, so you are not paying to move the rot, and your experts review and approve rather than copy and paste.

If you're starting closer to zero

The knowledge agent builds the knowledge base rather than moving one. Point it at the places your answers already live: the internal Q&A channel, your ticketing system, meeting recordings. A scheduled agent reads the shoulder-tap channel every morning, drafts articles from the answers your experts are already giving, organizes them by product as it goes, and routes each draft back to the person who gave the answer for a quick approval.

Within weeks, a usable, verified knowledge base exists that nobody had to sit down and write, assembled from how your team actually works today rather than from documentation someone hoped to get to. And the unusable legacy content doesn't need to be rehabilitated first. The agent treats it as one more source, keeps what's still true, and lets the rest retire.

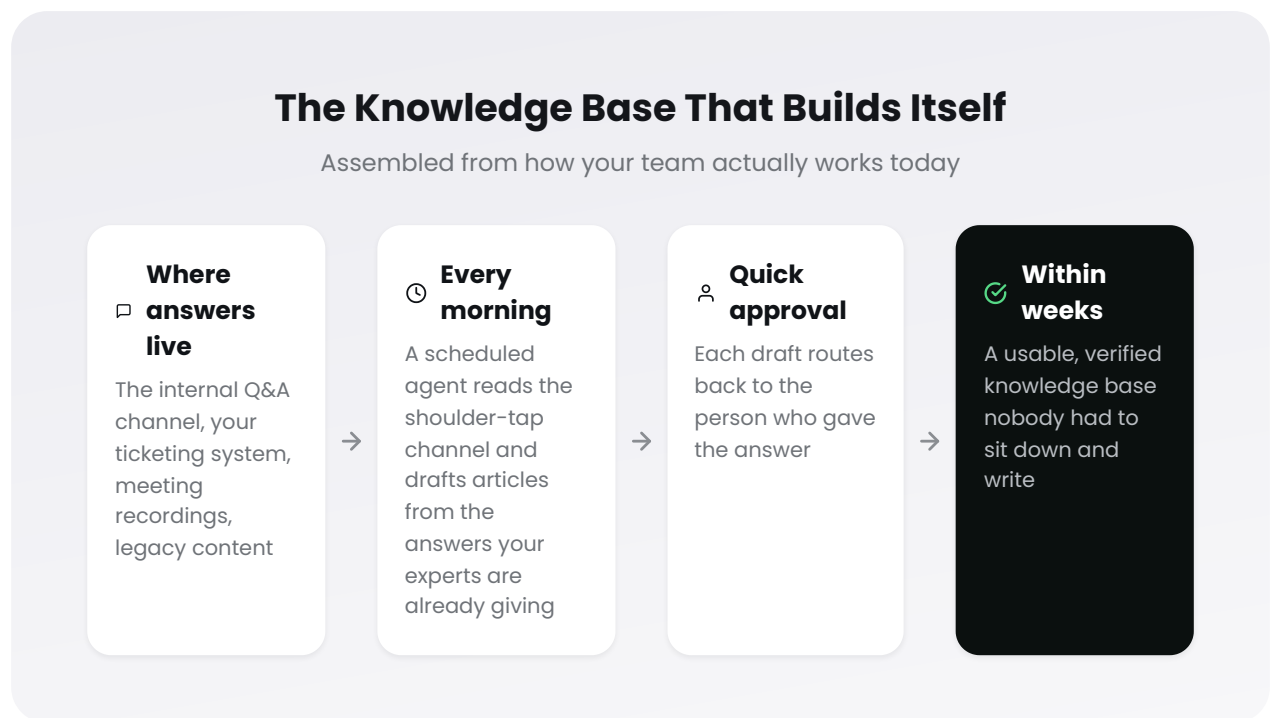


Figure 8. The knowledge base that builds itself. Point the agent at the places your answers already live and a verified knowledge base assembles from how your team actually works today.

A Partner, Not a Toolkit

In either case, you are not doing this alone or handed a toolkit and a wave goodbye. A dedicated Guru customer solutions engineer partners with your team through the buildout and stays as an ongoing partner afterward, tuning verification rules, designing the review workflows, and adapting the system as your products and regulations change.



A CS leader, describing what this partnership meant during a bank merger that required consolidating two knowledge bases on a hard deadline:

Our salvation.

The result should feel like this: **getting to an accurate knowledge base is Guru's project, and running your support organization on top of it is yours.**

In Production

More than 2,000 companies run Guru in production today. Among the financial services companies whose work informs this paper, the outcomes cluster around accuracy.

In Production

6.0

What This Looks Like in Production

66 → 16%

In a blind evaluation at a global fintech, critically inaccurate answers dropped from 66 percent with a competitor product to 16 percent.

840 → 90

A challenger bank rebuilt its core support knowledge from roughly 700 to 840 articles into 80 to 90 rigorously maintained ones, the same coverage with a fraction of the surface area to keep accurate.

3 : 6,000

That knowledge is maintained by a team of three supporting more than 6,000 customer operations users.

98 → 100%

That bank's continuous feedback loop, 20 to 30 corrections per week from frontline agents, moved chat QA success from 98 percent to 100 percent.



An audit that leadership called unsurvivable without it

A major Latin American bank passed a regulatory audit that leadership described as unsurvivable without traceable, verifiable knowledge sources.



Exactly the slice their contract covers, and nothing else

A US digital bank uses scoped, permissioned collections to give BPO contractors exactly the slice of knowledge their contract covers, and nothing else.



More than 2,000 companies run Guru in production today. The outcomes above come from the financial services companies whose work informs this paper.

Three Questions for Your Organization

01 What percentage of your knowledge base has been verified in the last 90 days?

Not written, not viewed. Verified by someone accountable for its accuracy. If you can't answer, the answer is close to zero, and your AI is answering customers from it anyway.

02 How do you learn about a knowledge gap today?

If the honest answer is "a customer complaint" or "an escalation," your knowledge system is being QA'd by your customers.

03 If a regulator asked what your AI told a customer about a specific policy last Tuesday, could you answer?

In this industry, that question is not hypothetical. It is a matter of when.

If those questions are uncomfortable, the knowledge layer is the bottleneck, and it is a solvable one. The financial services companies that engineer accuracy into their knowledge will find that every AI investment they make compounds. The rest will spend more on AI tooling each year and get less from it, because the layer underneath was never built to stay true.



ABOUT GURU

Guru is an enterprise AI search and knowledge platform that delivers trusted information from your company's scattered apps, chats, and docs — directly in your workflow.

If the three questions in this paper are uncomfortable, the knowledge layer is the bottleneck, and it is a solvable one. See what verified, governed knowledge looks like running under your support organization's AI.

[Learn more at getguru.com](https://getguru.com)