



A FIELD GUIDE FOR ENTERPRISE AI LEADERS

# Day Two: Four Decisions for Enterprise AI

The pilot worked. Now the rollout meets the whole company. Four decisions determine whether it delivers a return, and each one comes with a tool your team can put to work this week.



01  
Adoption



02  
Accuracy



03  
Efficiency



04  
Ownership

Rick Nucci, Co-founder and CEO, Guru

September 2026

# The pilot works. Then the rollout meets the whole company.

Almost every enterprise AI program starts the same way: a clean corpus, a motivated team, an impressive demo, and an approved budget. That is day one, and it almost always works. Day two is when the same system meets real permissions, contradictory documents, content nobody has touched since 2023, and a bill that grows with every action.

## Day one: the pilot

- Clean corpus
- Motivated team
- Impressive demo
- Budget approved

## Day two: the rollout

- Real permissions
- Contradictory documents
- Content untouched since 2023
- A bill that grows with every action

## HAVING AI VERSUS USING AI

**88%**

use AI regularly in at least one business function

**80%**

say AI improved their individual productivity

**37%**

can attribute any enterprise EBIT impact to AI

**6%**

qualify as high performers on company-wide profit

Source: McKinsey, The State of AI in 2026

Nearly nine in ten companies have AI. Fewer than four in ten can point to a financial result, and the 6 percent who can point to a significant one are not using better models. Everyone has the same models. Having AI is a purchase. Using AI is a function doing its actual work differently and getting a different result. Nothing forces a company to resolve these four decisions on day one, which is exactly why most companies do not.

## FOUR DECISIONS, ONE CHAIN

01

### Adoption

Will the whole company use it?

02

### Accuracy

Can it be trusted to be right?

03

### Efficiency

Can you afford it at scale?

04

### Ownership

Does the knowledge stay yours?

## A chain, not a list

People only use AI if it is right, you only scale it if you can afford it, and it only compounds if it stays yours. Every worksheet and checklist that follows is vendor-neutral and applies whether Guru is involved or not.

# One layer every AI tool in the company draws from

## DEFINITION

The collective knowledge of a company, verified and structured so that both people and AI systems can query it, each seeing only what they are permitted to see.

It sits between AI systems and the raw sources they would otherwise read directly, and resolves what is true once, so every agent, model, and chat tool inherits the same answer.

## WHERE IT SITS

### RAW SOURCES

Docs and wikis

Slack and meetings

Tickets and CRM

Decks, PDFs, sheets



### CONTEXT LAYER

**Verified. Governed.  
Resolved once.**

Permissions applied. Truth  
arbitrated. Stale retired.



### CONSUMERS

Claude, ChatGPT, Copilot,  
Gemini

Support, HR, IT, GTM agents

Coding agents and MCP  
clients

Your people

## WHAT IT IS NOT

- **Not a document repository.** Storage does not decide what is current.
- **Not a vector index over everything.** Similarity is not truth.
- **Not a single chat tool.** Every consumer, chat tools included, reads from it.

Knowledge bases, RAG pipelines, and search indexes either feed it or consume it.

## WHY THIS IS INFRASTRUCTURE, NOT ANOTHER TOOL

Organizations already run an average of **12 AI agents**, a number projected to grow **67 percent within two years**, and **half of those agents operate in isolated silos**. Every one of them either shares a context layer or builds its own private version of your company's knowledge.

Source: Salesforce, 2026 Connectivity Benchmark Report

"WE ALREADY HAVE COPILOT." KEEP IT. GIVE IT A SOURCE OF TRUTH.

### WHAT IT DOES WELL

#### Reach and permissions

Copilot, Gemini, and ChatGPT Enterprise already see your files and respect your access model. That part is solved.

### WHAT IT DOES NOT DO

#### Decide what is true

They rank by relevance. Nothing retires the old price sheet or arbitrates two conflicting documents.

### WHAT THE LAYER ADDS

#### One verified source for all of them

The same governed knowledge behind Copilot, your support agent, and your coding agents. Fix an answer once, everywhere.

# Adoption



The whole company actually uses AI to improve their work, not just the small group of people who were already good at it.

## THE DEFAULT

# 75%



Early. A few chats, no connection yet to working better.

# 20%



Up and coming. Real value, well behind the edge.

# 5%



Leading edge. Sets the bar.

Distribution observed across Guru's customer base

A blank chat box serves the 5 percent beautifully and does almost nothing for the 75 percent. To get value from an empty box you already need to know what is possible, how to phrase the question, and whether the answer can be trusted. It asks the 75 percent to behave like the 5 percent.

Training alone cannot close a gap that keeps widening, because the frontier moves faster than any enablement program. Buying everyone a license, announcing the rollout, and counting chats is what having AI looks like. Nothing changes in the business.

### HAVING AI

Buy everyone a license. Announce the rollout. Count chats. Declare adoption. Top-down mandates like this stall, and nothing changes in how the work gets done.

## WITH A CONTEXT LAYER

Purpose-built agents are designed once by the people who already know what good looks like, then rolled out to everyone. The 75 percent inherit that expertise the first time they use one: the sources, the guardrails, and the definition of a good answer are already in place. Adoption stops depending on every employee learning to prompt, and starts depending on how many workflows have an agent built for them.

### WITH GURU

Guru's answer is multiplayer knowledge agents. Instead of handing everyone an empty box, the 5% who already know what good looks like build agents with purpose designed in, and everyone else inherits that expertise the first time they use one.

Knowledge Agents are built by an owner, used by a team, and improved by every interaction. They meet people where they already work: in Slack, Teams, the browser, the Guru web app, or any AI tool and agent connected via MCP.

### USING AI

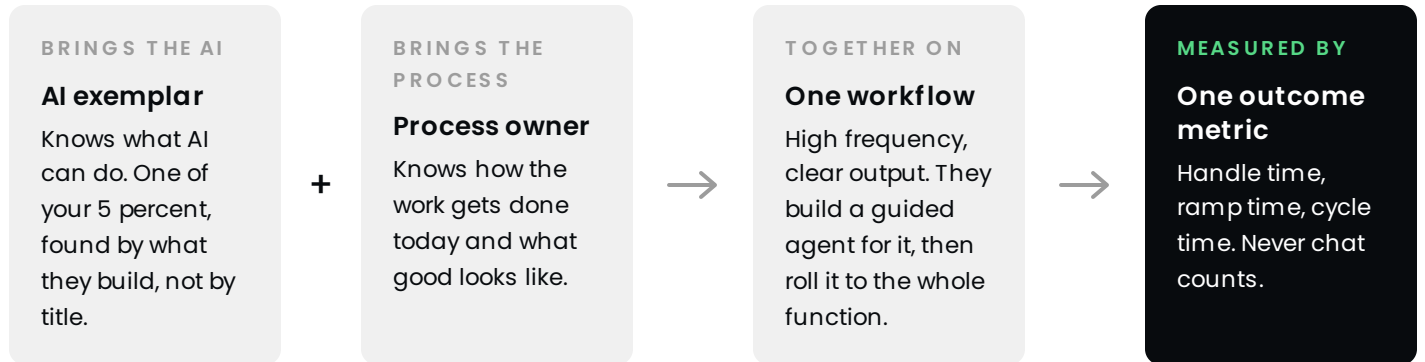
A function doing its actual work differently and getting a different result. Built team by team, problem by problem, and measured by the outcome of the work rather than by usage of the tool.

### Depends on accuracy

The 75 percent are the least equipped to catch a wrong answer, so they are the group that stops using AI fastest when it burns them. Picture the rep who asks about pricing, gets a confident wrong answer in front of a customer, and never asks again.

# The AI pod: one per function

Using AI happens when a function combines AI expertise with business process expertise. Pair the two, give them one workflow, measure one outcome, then roll it to the whole function. It is an org design move, not a software purchase.



## POD CHARTER: FILL THIS IN FOR YOUR FIRST FUNCTION

| The pod                         | The work                     |
|---------------------------------|------------------------------|
| Function                        | Workflow                     |
| AI exemplar                     | Who runs it today, how often |
| Process owner                   | Outcome metric               |
| Executive sponsor               | Baseline today               |
| Agent live by                   | Target in 30 days            |
| Rolled to the whole function by | Target in 90 days            |

### Picking the workflow

- Runs many times a day, not once a quarter
- Has a clear, checkable output
- Depends on knowledge that already exists somewhere
- Hurts today: slow, inconsistent, or done by too few people

### Finding the exemplar

- Look at what people have built, not what they say
- Already uses skills, MCP, or agents in their own work
- Sits inside the function, not in IT or a center of excellence
- Wants to make the process owner look good

### Rules of the pod

- One workflow at a time. Ship it before starting the next.
- Measure the outcome of the work, never usage of the tool.
- Roll to everyone in the function, then hand off.
- Team by team compounds. Top-down mandates stall.

Why one metric: chat counts are how companies convince themselves they have adopted. A pod that cuts handle time in one queue proves more than a thousand logins ever will.

# Accuracy



The knowledge your AI draws from is verified, current, and correct. Not just findable, but trustworthy.

## THE DEFAULT: RELEVANT IS NOT RIGHT

### THE QUESTION

**"What is our current pricing?"**

### WHAT THE CONNECTORS RETURN

- New price sheet (Sept 2026)
- Old price sheet (still there)
- Slack thread debating the change
- Last quarter's sales deck

**All four match the query equally well. One of them is true.**

Connectors are built for relevance. They have no concept of whether something is still true, so the agent confidently serves whichever result ranks highest. Every agent and chat tool in the company makes this mistake independently, so there is no single place to fix it. In regulated industries a wrong answer is not an inconvenience, it is a liability.

**Inaccuracy** is the negative consequence of gen AI that organizations most often report experiencing, ahead of cybersecurity and explainability. **44 percent** have experienced at least one negative consequence.

Source: McKinsey, The state of AI in early 2024

## WITH A CONTEXT LAYER

**Decide what is true once. Every agent inherits it.**

- Retrieval draws only from what passes verification
- The people closest to the knowledge own the standard
- Verification is continuous, not a quarterly audit
- A correction made once reaches every consumer

### WITH GURU

Guru puts a verification layer between your raw knowledge and every consumer of it, human or AI. The work of deciding what's true happens once, by your team, and then every employee and every agent draws from that verified set. Three systems keep that layer accurate over time.

**Verification.** A subject matter expert approves that knowledge is accurate and worth trusting. Drafts never go live on their own.

**Agentic Knowledge Base.** The agent mines Slack threads, support tickets, and meeting recordings, drafts and organizes knowledge, and routes it to experts for review, so the newer version replaces the old one instead of drifting alongside it.

**Quality Rules.** Daily, automated audits of the knowledge itself. Untrusted content is hidden from the agent's context, and every decision is logged with its reasoning.

Every answer lands in the AI Agent Center with its full trace, and permissions are inherited from the source, so verification never widens access.

### Depends on efficiency

Once the layer is accurate enough to depend on, you have made it load-bearing. Every agent, team, and workflow runs through it, and the cost of running it becomes a cost of running the company.

# Do you have an accuracy strategy? Seven questions.

Score these with the people who own the knowledge, not the people who bought the tools. If most answers are "we don't" or "a person, eventually," there is no accuracy strategy yet.

| Self-assessment: your accuracy strategy                                                | Yes                      | Partly                   | No                       | Who owns this today |
|----------------------------------------------------------------------------------------|--------------------------|--------------------------|--------------------------|---------------------|
| 1. Does every knowledge domain have a named owner of truth?                            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |
| 2. When two documents conflict, does a rule (not the AI) decide which one wins?        | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |
| 3. When pricing or policy changes, is the old version retired from the AI within days? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |
| 4. Can you see which source every answer came from?                                    | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |
| 5. Do you measure the accuracy of the knowledge itself, not just the answers?          | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |
| 6. When an answer is wrong, does one fix reach every tool and agent?                   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |
| 7. When the AI does not know, does it say so instead of guessing?                      | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                     |

## MOSTLY YES

### You have a strategy

Check that it covers every agent and chat tool, not only the one it was built for.

## MIXED

### Start with 1, 3, and 6

Name an owner per domain, define what retires an old version, and route every correction through one place.

## MOSTLY NO OR PARTLY

### Build this before the next agent ships

Every new agent added without it is another place the same wrong answer gets served.

## The questions most teams have never asked

**Question 3** is the pricing example: what actually makes the old version go away, and how fast. **Question 5** separates the accuracy of the knowledge from the accuracy of the answer. Almost everyone instruments the agents; almost nobody instruments the knowledge underneath them. **Question 6** exposes the "every tool makes the mistake on its own" problem. A fix that only reaches one tool is not a fix.

Assessed by \_\_\_\_\_ Date \_\_\_\_\_ Re-assess on \_\_\_\_\_



# Efficiency



Getting accurate AI answers at a fraction of the token cost of the alternatives.

## THE DEFAULT: THE BETTER THE ROLLOUT GOES, THE WORSE THE BILL GETS

### 01 · ONE QUESTION

#### Fan-out

One question hits five or ten connectors. Each returns its own pile. Every new source adds to it.



### 02 · EVERY TIME

#### Re-reasoning

Deciding which of three conflicting documents is current happens after the tokens are spent. Then again for the next person.



### 03 · THE BILL

#### Cost outruns usage

Cost scales with usage and with corpus messiness at once. No economy of scale.

Every AI answer is paid for in tokens, and tokens are also the billing unit for the model providers. Every new model announces a lower token price, and the bill still goes up, because the waste is architectural: the same expensive reconciliation runs on every question, for every employee, in every tool.

## WITH A CONTEXT LAYER: ONE QUESTION, TWO WAYS

### AGENT WIRED TO RAW SOURCES

Agent → Wiki, Slack, Drive, CRM, Tickets

#### About 45,000 tokens

for one answer. Five searches, five piles to reconcile, and it all runs again for the next person who asks.

### AGENT WIRED TO A CONTEXT LAYER

Agent → Context layer → Sources

#### A fraction of the tokens

One retrieval returns the verified answer with its context already joined. Flat as sources grow. Built once, reused by everyone.

### WITH GURU

Connect once to Guru instead. The agent pulls from a single curated, verified knowledge layer rather than blindly searching many systems, so each answer requires a fraction of the tokens. One customer entity in Guru replaces five to ten distinct searches for the same information, repeated endlessly across every employee and workflow. The synthesis work happens once and gets reused by everyone, which means costs stay flat and predictable even as usage scales.

Guru's MCP server gives your existing tool one connection point to clean, verified, permission-aware knowledge. Adding sources does not move the number, because filtering happens before tokens are spent, not after.

Token figures are illustrative for a five-source question. Keep Claude, Copilot, or ChatGPT. Point them at one retrieval instead of ten.

#### Depends on ownership

A layer you can afford to run at scale only matters if it stays yours. Every correction, verified answer, and agent memory it accumulates has to be something you can see and take with you.

# The noisy context formula

Estimate your own number with four inputs you already have. Treat the result as a diagnostic, not a quote: if the number is large, noisy context is the reason, and the fix is architectural, not behavioral.

|                     |                      |   |                      |   |                          |   |                      |   |                      |
|---------------------|----------------------|---|----------------------|---|--------------------------|---|----------------------|---|----------------------|
| <b>Your numbers</b> | <input type="text"/> | × | <input type="text"/> | × | <input type="text"/>     | × | <input type="text"/> | = | <input type="text"/> |
|                     | questions per month  |   | sources per question |   | tokens per source result |   | your price per token |   | monthly context cost |

## WORKED EXAMPLE AT ENTERPRISE SCALE

|                                 |                                   |   |                                |   |                                    |   |                                    |   |                     |
|---------------------------------|-----------------------------------|---|--------------------------------|---|------------------------------------|---|------------------------------------|---|---------------------|
| <b>Without a context layer</b>  | <input type="text" value="4.4M"/> | × | <input type="text" value="5"/> | × | <input type="text" value="9,000"/> | × | <input type="text" value="\$ __"/> | = | <b>~200B tokens</b> |
| <b>With one, same questions</b> | <input type="text" value="4.4M"/> | × | <input type="text" value="1"/> | × | <input type="text" value="9,000"/> | × | <input type="text" value="\$ __"/> | = | <b>~40B tokens</b>  |

20,000 employees × 10 questions a day × 22 working days = 4.4M questions a month. "With" assumes one retrieval of about 9,000 tokens. If your retrieval is fatter the ratio shrinks, but it stays a multiple.

### Where each number comes from

- **Questions per month:** employees × questions per day × working days. Ten a day is a reasonable starting point for a knowledge worker.
- **Sources per question:** count the connectors on your busiest agent.
- **Tokens per source result:** 5,000 to 10,000 is a fair default for a returned document or thread.
- **Price per token:** your model's input price. Use the blended rate if you run several.

### Pricing the gap

The delta in the worked example is about 160B tokens a month. At \$3 per million input tokens that is roughly **\$480,000 a month**. At \$1 per million it is still **\$160,000**. For a 1,000-person company, divide by 20: tens of thousands of dollars a month, before tool schema overhead.

Overhead most people miss: with half a dozen tools connected, the tool schemas alone can load a couple hundred thousand tokens before the agent takes a single action.

### READING YOUR RESULT

A big number is not a reason to ration AI. It means the same reconciliation is being paid for on every question. Reduce sources per question to one and the number drops by the same multiple, with no change in how people work.

### YOUR RESULT

Monthly context cost today \_\_\_\_\_

With one retrieval per question \_\_\_\_\_

# Ownership



Your company's collective knowledge, your context, always belongs to you. It's structured, portable, and never held hostage by a vendor.

## THE STAKES: THE ONLY LAYER IN THE STACK THAT COMPOUNDS

- Your verified knowledge
- The memories your agents collect
- The decisions they make and the corrections people make
- The skills they use to do their work
- What supersedes what, and what a good answer looks like here

Models get cheaper and swappable every quarter. Vendors get acquired. Apps get replaced. This is the one thing that took years to accumulate and cannot be rebuilt on any timeline a business would accept. You must be able to see all of it and take all of it with you.

Ownership has no Day Two symptom. Nothing feels wrong until year two, at a renewal or a vendor consolidation. It is decided the day you sign. And it never means build it yourself: the context layer is a distinct layer you control, separate from any single agent, model, or chat tool.

## AGENTS ARE SWAPPABLE. THE KNOWLEDGE IS NOT.

### WITHOUT A CONTEXT LAYER

- Each agent absorbs its own private copy of your knowledge
- Corrections and memory live inside tools you cannot export
- Switching vendors means starting over
- Every experiment starts from zero

### WITH A CONTEXT LAYER

- Knowledge, skills, rules, and logs are visible and exportable
- Plug in a new agent: it works from day one
- Unplug it later: the knowledge stays in the layer
- Try new agents, models, and chat tools freely

### WITH GURU

Guru is built on the opposite premise. Your knowledge layer belongs to you, and the agents on top of it are swappable. Connect any agent to Guru, and if a better model or tool comes along next year, switch to it freely. Your verified company knowledge stays intact, portable, and yours.

The knowledge base is documents, in open formats, and it is the asset. Every document is retrievable as structured JSON over the REST API and through the CLI. System prompts, skills, and quality rules are readable instructions, visible and editable in the product. The interaction data in the AI Agent Center is visible and reviewable by your owners.

### The whole chain

Adoption fails because people get burned. Accuracy fails because nothing arbitrates truth. Efficiency fails because the work repeats per query. Ownership fails because curation lives vendor-side.

# The ownership checklist

Can you see it, and can you export it today? Run this against every AI vendor in your stack, including Guru. For each item the bar is the same: today, in an open format, without a services engagement. Ask before you sign, not at renewal.

| What you should get back from every AI vendor                                | See it                   | Export it                | Format | Notes |
|------------------------------------------------------------------------------|--------------------------|--------------------------|--------|-------|
| 1. Your verified knowledge, in an open format                                | <input type="checkbox"/> | <input type="checkbox"/> |        |       |
| 2. Agent prompts, configurations, and skills                                 | <input type="checkbox"/> | <input type="checkbox"/> |        |       |
| 3. The training data and memories your agents collected                      | <input type="checkbox"/> | <input type="checkbox"/> |        |       |
| 4. Logs of every question, answer, and correction                            | <input type="checkbox"/> | <input type="checkbox"/> |        |       |
| 5. The verification rules and decisions your team wrote down                 | <input type="checkbox"/> | <input type="checkbox"/> |        |       |
| 6. Access over standard protocols (MCP, API), not only a proprietary surface | <input type="checkbox"/> | <input type="checkbox"/> |        |       |

## TWO QUESTIONS TO ASK EVERY VENDOR, INCLUDING US

### "What exact bytes do we get back the day we leave?"

Not a promise of a future export. The actual files, formats, and endpoints available today.

### "Can we disconnect one agent without losing what it learned?"

If corrections, memory, and rules live inside the agent, the answer is no. If they live in a layer the agent reads from, yes.

### What "open format" means here

Files and records a different system can read without the vendor's help: structured documents, plain-text prompts and rules, and logs you can query. A proprietary export that only re-imports into the same product does not count.

### Where to look first

Chat tools and assistants, because that is where corrections and memory quietly accumulate. Some enterprise plans do not offer data export at all. Item 3 is the one most vendors cannot answer.

#### VENDOR ASSESSED

Vendor \_\_\_\_\_ Contract renews \_\_\_\_\_

Reviewed by \_\_\_\_\_ Date \_\_\_\_\_

#### DECISION

- ☐ Exports today ☐ Gaps to fix in contract  
☐ Do not proceed

# Four angles on the same missing piece of infrastructure

The context layer is not one more AI tool. It is what every AI tool in the company should draw from, which is what makes it infrastructure rather than an application. With all four decisions in place you get high adoption, real impact, managed cost, and knowledge that stays yours, which frees every team to try new agents freely.

## HOW GURU FITS

### Guru is the governed knowledge layer that provides the trusted context enterprise AI needs.

It is designed around the same four decisions, and it works with the AI tools you already run rather than replacing them. Claude, ChatGPT, Copilot, Gemini, your support and IT agents, and your coding agents all draw from one verified source.

01

#### Adoption: Knowledge Agents

AI-powered agents that operate autonomously within boundaries defined by your team. Built by an owner, used by a team, and improved by every interaction, in Slack, Teams, the browser, the Guru web app, or any AI tool connected via MCP.

02

#### Accuracy: Verification, Quality Rules, AI Agent Center

Experts approve before anything goes live. The Agentic Knowledge Base drafts updates from where knowledge is created. Quality Rules audit the knowledge daily and hide what fails. Every answer lands with its full trace.

03

#### Efficiency: Guru MCP Server

Verified company knowledge for every AI tool. Search verified knowledge, generate cited answers, and draft and update knowledge through one connection point to clean, verified, permission-aware knowledge.

04

#### Ownership: open formats, open access

The knowledge base is documents, in open formats, and it is the asset. Structured JSON over the REST API and CLI, readable prompts, skills, and quality rules, and interaction data your owners can review. Access over MCP and API.

## ABOUT GURU

Guru is the self-improving, governed knowledge layer that powers enterprise AI; structuring your company's knowledge, enforcing policy and accuracy automatically, and delivering trusted answers to every person and AI system across your stack.

[See Guru against your own knowledge](#)[getguru.com](https://getguru.com)

# Four things to do this week

None of these requires new software, and each produces something you can put in front of your leadership team by Friday.

## 01 ADOPTION · PAGE 5

### Stand up one AI pod

One function, one exemplar, one process owner, one workflow, one outcome metric. Fill in the pod charter and set a 30-day target.

Owner \_\_\_\_\_ Due \_\_\_\_\_

## 02 ACCURACY · PAGE 7

### Ask the seven questions

Find out whether you have an accuracy strategy before the next agent ships. Score it with the people who own the knowledge.

Owner \_\_\_\_\_ Due \_\_\_\_\_

## 03 EFFICIENCY · PAGE 9

### Run the formula

Size your noisy context cost with four numbers you already have. Count the connectors on your busiest agent and plug in your model's input price.

Owner \_\_\_\_\_ Due \_\_\_\_\_

## 04 OWNERSHIP · PAGE 11

### Run the checklist

Ask every AI vendor in your stack what bytes you get back the day you leave, and whether you can disconnect one agent without losing what it learned.

Owner \_\_\_\_\_ Due \_\_\_\_\_

## ABOUT GURU

Enterprise AI is only as good as the knowledge behind it. When knowledge is fragmented, outdated, or ungoverned, AI produces unreliable answers, makes the wrong decisions, creates compliance risk, and erodes trust. Guru solves this at the foundation, structuring and strengthening scattered knowledge, governing and continuously improving it, and powering every AI and human workflow from one trusted source. The result: a knowledge layer that gets more accurate over time, not less.

## SOURCES AND NOTES

McKinsey, The State of AI in 2026: On the road to ROI (August 2026). McKinsey, The state of AI in early 2024: Gen AI adoption spikes and starts to generate value (May 2024). Salesforce, 2026 Connectivity Benchmark Report, a survey of 1,050 enterprise IT leaders. The 75/20/5 distribution is Guru's observation across its customer base. Token counts are illustrative for a five-source question at typical result sizes; run the formula with your own numbers.

