// PROSSIMO

# Agents Need Knowledge, Not Just Data

On the effectiveness and cost of grounding AI agents in an organization's own knowledge

***Shota Ishii*** *(Prossimo)* | *October 2026*



*Photo: Lucrezia Carnelos on Unsplash*

## Executive summary

When an AI agent lacks an organization's knowledge, it guesses, and its guesses look like answers.

Most agents in business can now read an organization's data: its tables, documents and search results. Far fewer have access to what the organization knows *about* that data: which table is current, which records refer to the same supplier, which definition of "cost" each function uses. We call this governed body of knowledge, connected to the data and delivered to people and agents, a **knowledge layer**. This paper sets out the evidence for building one, and how to start.

Three findings stand out.

- **Answers improve on the same data.** Accuracy on an insurance database rose from 16.7% to 54.2% when the model queried modeled knowledge instead of raw tables. [3] The gain comes from modeling the knowledge, not from any particular tool, [8][9] and it is largest on complex questions. [3][10]
- **Agents can tell when to stop and ask.** A governed definition either answers or declines. A model guessing from raw tables can return a plausible wrong number with no warning. [8] For an agent that acts on its own results, knowing when to stop and ask matters as much as accuracy.
- **Agents can be cheaper and faster.** An agent that knows where the right facts are reads a few of them instead of everything that might be relevant, and reading less turns out to be more accurate [18]–[20] as well as cheaper. [10][15][16]

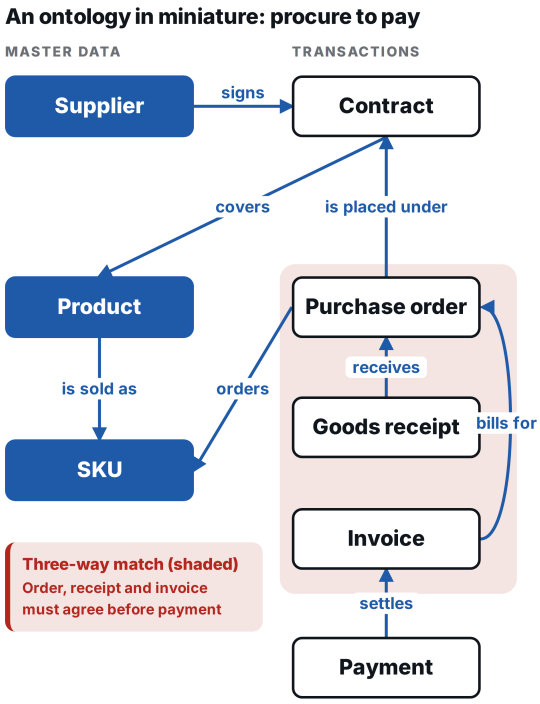
We recommend building the knowledge layer while building the first agents, one business decision at a time, rather than modeling the whole organization in advance. Section 6 describes how, and Section 7 sets out the limits of the evidence.

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

## 1. An agent that answers confidently and wrongly

An agent was asked whether a 4% price increase on one product would keep its margin above target. It searched the available data and seemed to find every table it needed. It found a cost table, calculated the margin and answered "yes" in a clear, natural reply.

The answer was wrong, for three reasons the agent had no way to see. The cost data was last year's: nothing in the table said *until when* the recorded cost was supposed to be valid or who maintained it. The largest supplier appeared under two different codes in the purchasing and accounting systems, so its costs were split across two records and the agent read only one. And "cost" specifically meant "landed cost" in procurement but "standard cost" in finance; the agent used the wrong one.




The agent here did not lack the data. It lacked the organization's knowledge about its data. Agent platforms have become good at giving models access to tables, documents and search results. What is usually missing is the organization's own definitions, the mappings between its codes and names, and the rules its experts apply, each with someone accountable for it. That knowledge is the subject of this paper.

> ***From our work.*** *The problem is not hypothetical, and in our experience its scale is larger than expected. In a product-data engagement for a US maker of ERP and point-of-sale software for beverage-alcohol retailers, more than 40% of item records were duplicate copies of the same product in the same container, arriving from different supplier feeds. Grape varietal names such as "Cabernet Sauvignon" appeared hundreds of times, with misspellings, inconsistent labels and no clear identity when blended with other varietals. An agent asked how many products a retailer carries, or which supplier sells a given wine, would have answered confidently from those records, and wrongly.*

## 2. Data and knowledge: our definitions

Words like "ontology" and "semantic layer" have become common in AI discussions, with overlapping meanings. We define the terms we use in two groups. The first covers what the organization holds:

| Term | What it is | In the example above |
|---|---|---|
| **Data** | Records in systems and documents | The cost tables, the supplier master, invoices |
| **Knowledge base** | Everything the organization holds to be true about its business, in whatever form: definitions, code mappings, rules, models | "Cost means standard cost in finance"; "supplier codes A-1023 and 88410 are the same company" (illustrative) |
| **Knowledge layer** | The knowledge base in use: connected to the data and delivered to people and agents, with a source and a date for every item and an owner for every kind of item | The agent can look up which cost table is current, and who says so |

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

| Term | What it is | In the example above |
|---|---|---|
| **Ontology** | The schema of meaning: what kinds of things exist and how they relate. One component of the layer | Supplier, product, cost, and how they connect |
| **Semantic layer** | Governed definitions of metrics and dimensions from which database queries are generated. One component | “Gross margin” defined once, used by every report and agent |
| **Knowledge graph** | The actual entities and relations, stored according to the ontology. One component | This supplier, these products, these contracts |

When people say “ontology”, they often mean the whole knowledge layer. In our terms, the ontology is one component of the layer: the most visible one, but rarely the first an agent needs. The others include vocabularies, identity mappings between codes and names, rules and the documents themselves, each linked to its source.

The second group covers what an agent works with when it runs:

| Term | What it is | In the example above |
|---|---|---|
| **Instructions** | Standing directions on how the agent should behave: its role, rules and limits, written by whoever deploys it. Today they are often where an organization’s intent ends up, as a paragraph of text | “You are a pricing analyst. Never quote a margin below target” |
| **Memory** | What the agent keeps from its own past work: preferences, earlier results, corrections. Usually unreviewed and without an owner | The agent recalls that a manager asked for margins in yen last week |
| **Context** | The information assembled for an agent to carry out a task: the request, the relevant items from the knowledge layer, its instructions and the results of the tools it has used. Rebuilt for each task; not normally kept | The current cost table, the supplier mapping and the pricing limit |
| **Prompt** | Everything sent to the model at one step: instructions, context and the request | All of the above, plus “Would a 4% increase keep the margin above target?” |

Memory is what one agent happens to remember. Anything worth keeping should move into the knowledge layer, with a source and an owner.

Two boundaries matter. First, “semantic layer” here has the narrow meaning used by the vendor-led Open Semantic Interchange specification: governed metrics and dimensions. [1] Second, we keep **what the organization wants or intends** separate from **what it knows**. The knowledge layer can record that a pricing limit exists and what it says. How strictly the organization is bound by it, and which goals take priority, belong to a separate model of intent, which we describe in another paper. [2]

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

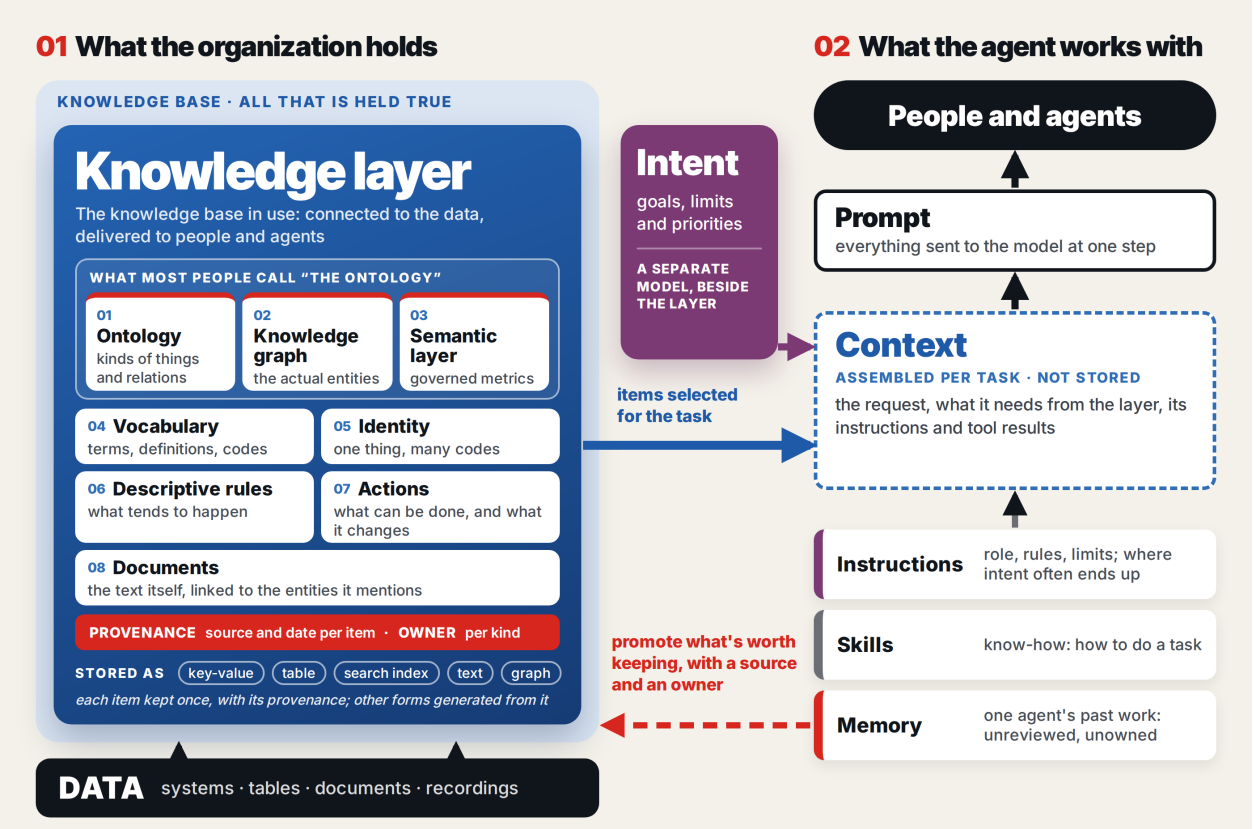



*Figure 1. What the organization holds (the knowledge layer inside the wider knowledge base, with intent beside it) and what an agent works with when it runs. Context is assembled for each task from the layer; memory worth keeping is promoted into the layer.*

## 3. What the evidence shows

The published studies show three patterns.

**On the same data, modeled knowledge produces better answers.** In a benchmark built on a standard insurance data model, GPT-4 answered 16.7% of business questions correctly when querying the database directly, and 54.2% when querying a knowledge graph built over the same data. [3] Adding checks against the ontology, and letting the model repair failed queries, raised accuracy to 72.6%, with 8% of questions answered "I don't know" instead of wrongly. [4] On BIRD, a text-to-SQL benchmark of 95 databases, supplying expert-written hints with each question raised GPT-4's accuracy from 34.9% to 54.9%. [5] Because those hints are written for each question in advance, this is closer to an upper bound than to what an organization would achieve with reusable definitions. Other studies point the same way: grounding retrieval in an ontology raised answer correctness by 40% on average across four models, [6] and in a deployed customer-service system, LinkedIn cut the median time to resolve a customer issue by 28.6% over about six months. [7]

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

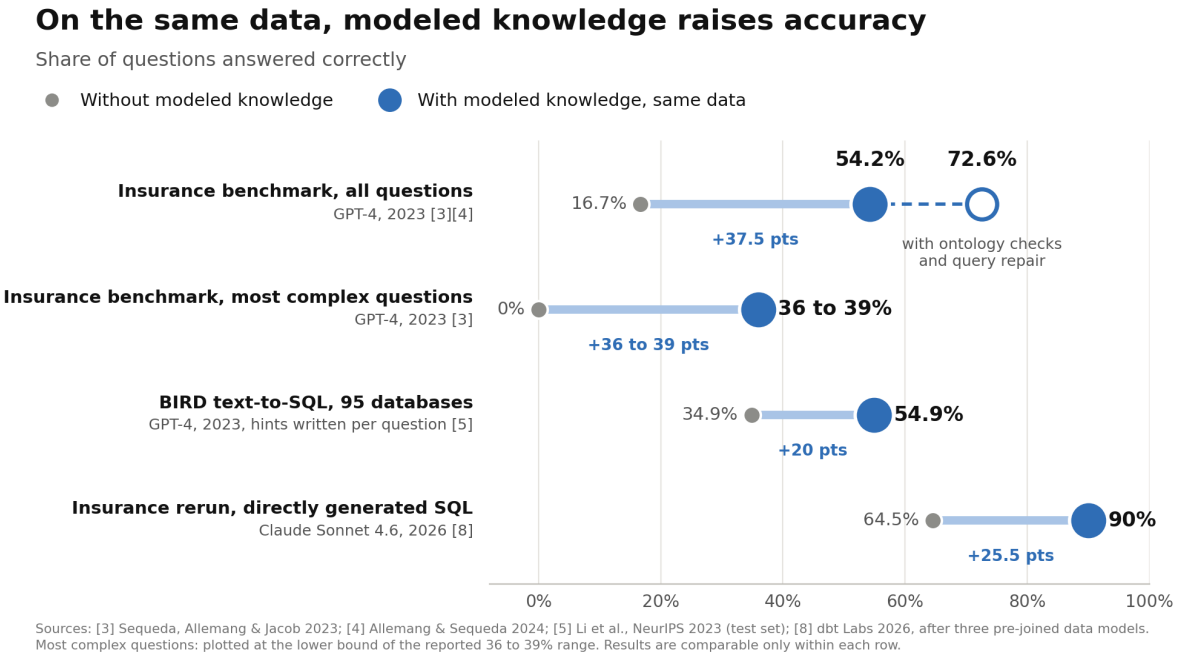



*Chart 1. On the same data, modeled knowledge raises accuracy. Sources: [3][4][5][8]. Results are comparable only within each row.*

**The improvement comes from the knowledge, not the tool.** A 2026 rerun of the insurance benchmark by dbt Labs, a vendor of semantic-layer software, is useful precisely because it complicates the vendor's own story. On the questions the semantic layer could express, the AI was correct every time, against only 51% to 63% for queries generated directly from the database. On the remaining questions, it returned an error rather than an answer. Then three small pre-joined data models were added, and both methods could use them. Accuracy through the semantic layer rose to between 98% and 100% of all questions, and directly generated queries rose too, from 64.5% of all questions to between 84% and 90%. [8] Modeling the data improved the answers for every method of querying it. A 2026 preprint testing three frontier models points the same way: adding a semantic-layer document raised accuracy by 17 to 23 percentage points, while the choice of model made no significant difference. [9]

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

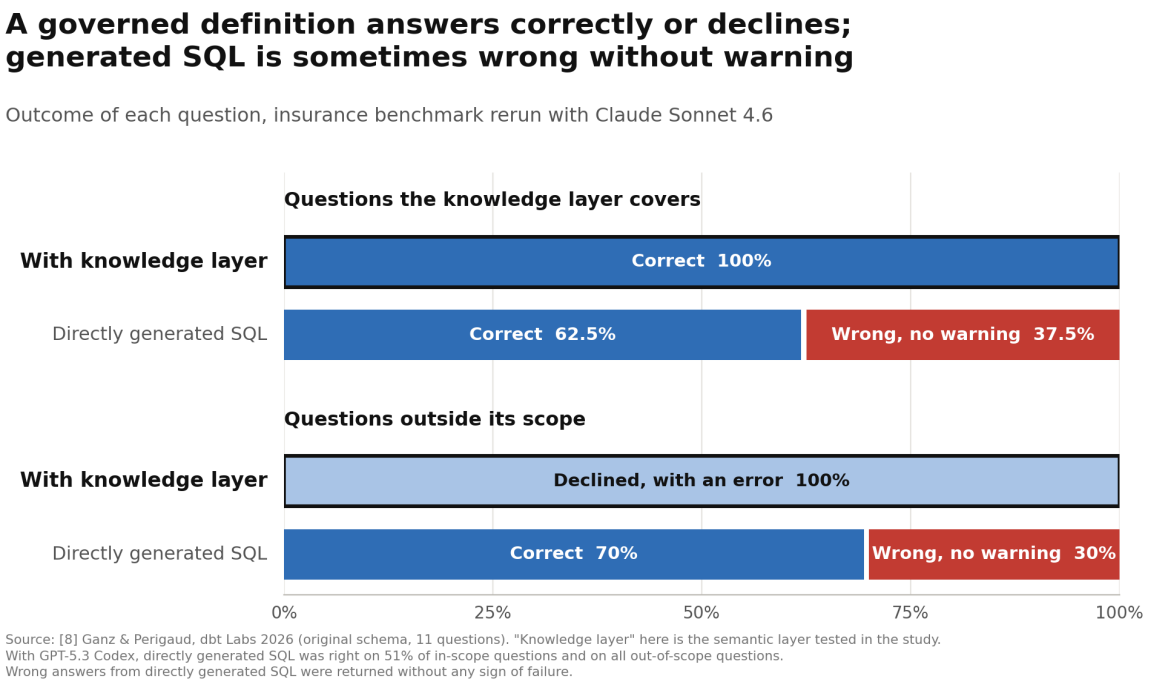



*Chart 2. A governed definition answers correctly or declines; generated SQL is sometimes wrong without warning. dbt rerun of the insurance benchmark, original schema, Claude Sonnet 4.6. [8] With GPT-5.3 Codex, generated SQL was right on 51% of in-scope questions and on all out-of-scope questions.*

**The improvement is largest on complex questions.** In the insurance benchmark, questions over the most complex parts of the schema went from 0% correct to between 36% and 39%. [3] A 2025 benchmark comparing graph-based and plain document retrieval found them tied on simple fact lookup, with graph-based methods ahead by 10 to 13 points on complex reasoning and summarization. [10]

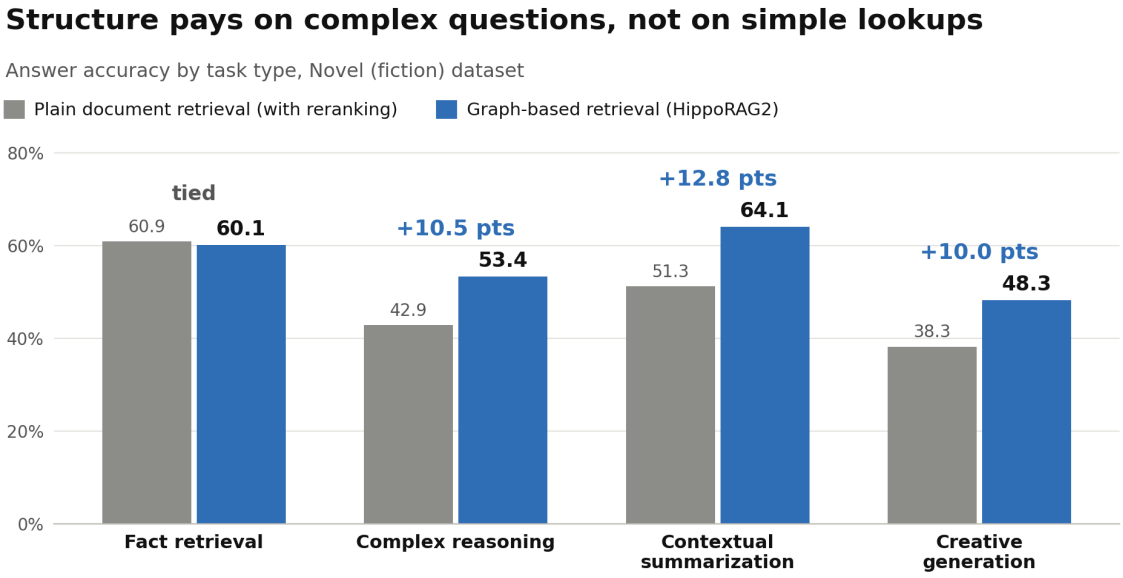



*Chart 3. Structure pays on complex questions, not on simple lookups. GraphRAG-Bench, Novel (fiction) dataset. [10]*

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

## 4. Why agents need it more than chat assistants do

A chat assistant answers one question. An agent carries out many steps and acts on the results, so small errors that go unnoticed compound. If each step of a ten-step task is right 95% of the time, the whole task is right about 60% of the time (0.95 to the tenth power is 0.599). Published agent benchmarks are consistent with this arithmetic: the best agent on the updated leaderboard of one benchmark of realistic workplace tasks completed 30% of them, [11] and in a retail customer-service benchmark, an agent that succeeded on 61% of tasks at the first attempt succeeded on all of eight repeated attempts less than 25% of the time. [12] These benchmarks do not isolate missing knowledge as the cause; they show how little room for error an agent has.

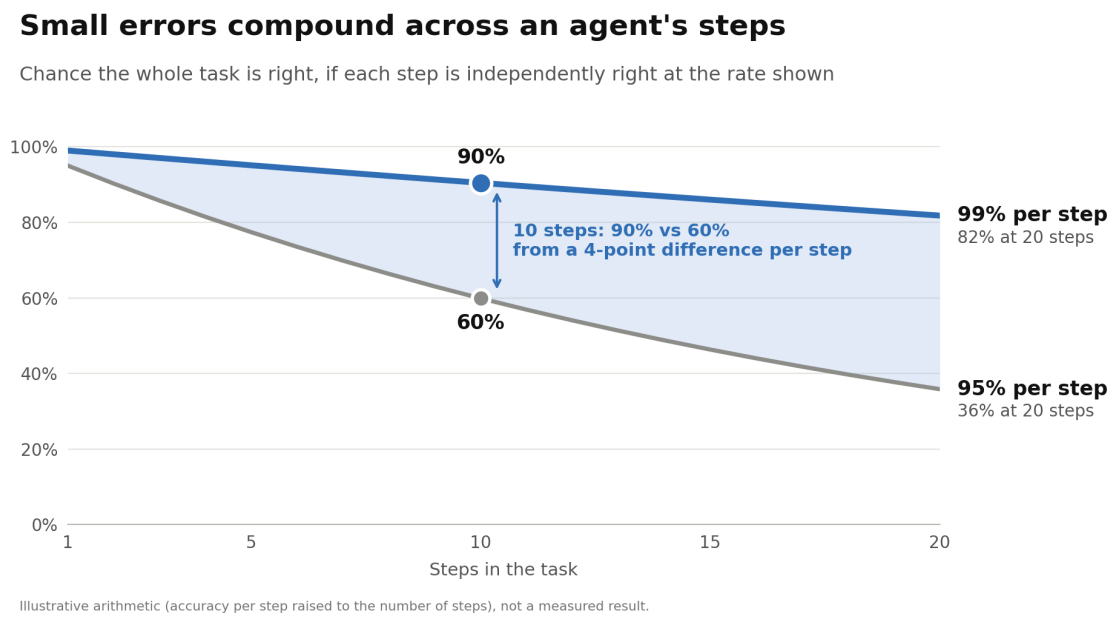



*Chart 4. Small errors compound across an agent's steps. Illustrative arithmetic, assuming each step is independently right at the rate shown.*

The problem grows when several agents share a task. Independent pieces of work, such as fetching prices or drafting a comparison table, can be handed to separate sub-agents without difficulty. But if one agent reads a delivery "window" as the contractual period and another as the loading schedule, their results will not reconcile, and neither will notice. Anthropic, describing its own multi-agent system, notes that the approach suits broad research tasks but not work in which all agents must share the same context or depend heavily on each other. [13] Shared definitions are what allow several agents to work from the same assumptions.

There is a further reason, and it may matter the most. An agent works in a loop, using tools and deciding from the results what to do next. [14] At each step, it has to judge whether it knows enough to act. A governed definition gives it something to check against: it either answers or declines. In the dbt study, the semantic layer returned an explicit error on questions outside its scope, where directly generated queries sometimes returned plausible wrong numbers with no sign of failure. [8] Without a knowledge layer, an agent has a much weaker signal of when to stop and ask.

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

## 5. Speed and cost

A common assumption is that adding structure makes agents slower and more expensive. The evidence suggests the opposite, provided the structure is used to send the agent less text, not more.

The amount of text an agent reads per question varies enormously by design. On one benchmark, plain retrieval used about 880 tokens (the units of text a model reads, roughly three-quarters of a word each) per query, while the most expensive graph-based method used up to about 40,000. [10] *How* the knowledge is prepared matters as much as how it is stored. Deferring expensive summarization until a question needs it cut indexing costs to 0.1% of the full graph approach, and made global queries more than 700 times cheaper at comparable quality. [15] Sending easy questions to plain retrieval and only hard ones to long inputs cut costs by 39% to 65% with little loss of quality. [16] Knowledge that agents consult repeatedly can also be cached: Anthropic charges about a tenth of the normal price to reread cached input, and in one published example a cached 100,000-token prompt began answering in 2.4 seconds instead of 11.5. [17]

Reading less is also more accurate. Language models make poor use of information in the middle of a long input. [18] On a test designed to require more than literal matching, 11 of 13 models fell below half of their short-input performance at 32,000 tokens. [19] In one experiment, a focused input of about 300 tokens outperformed the full 113,000-token history for every model family tested. [20]

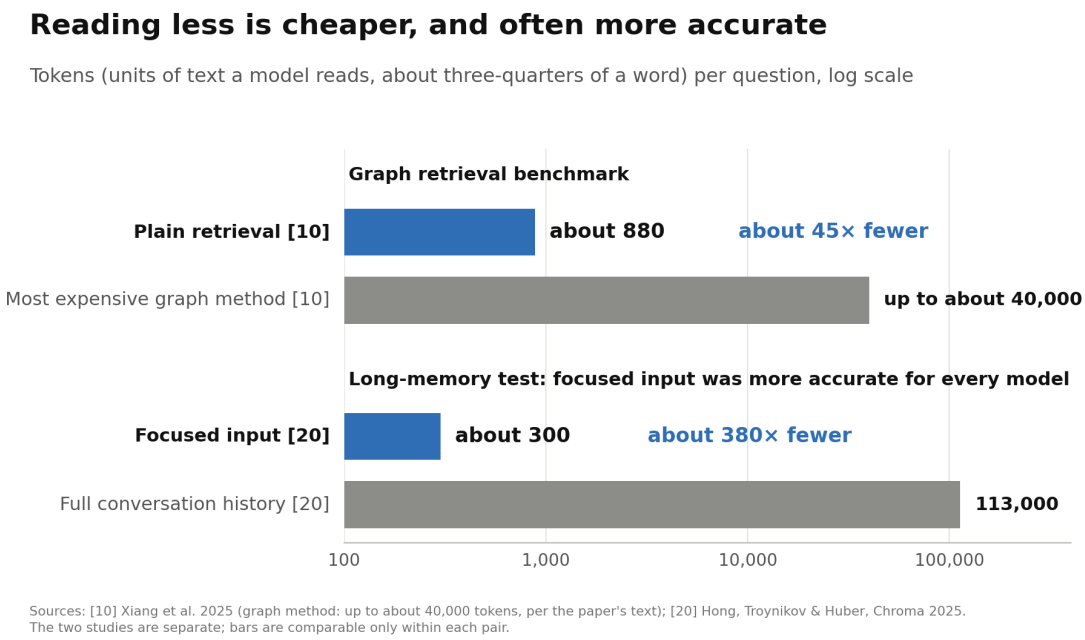



*Chart 5. Reading less is cheaper, and often more accurate. Tokens per question, log scale. [10][20] The benchmark paper's own figures show up to about 331,000 tokens for the most expensive method; we use the lower figure from its text.*

Many vendors and commentators treat a knowledge layer as an ontology implemented as a graph. A knowledge layer is not the same thing as a graph database, and speed depends on keeping the two apart. Most facts an agent needs are found fastest by simple means: a code mapping or the current version of a table by a direct lookup, a total by a database query, a relevant passage through a search index. A graph earns its cost for questions that follow several relationships at once. What made the expensive graph methods expensive was building and summarizing the graph with a language model and reading broadly across it at question time. [10] We therefore store each item once, with its source, and serve it through whichever form answers the question fastest. An agent that knows which

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

cost table is actually current, and which codes refer to the same supplier, can fetch two facts. An agent that does not must read much more and hope the right facts are in it.

## 6. Build the knowledge layer while building the agents

If a knowledge layer is this useful, why do many organizations resist "ontology projects"? Enterprise-wide data models and master-data programs have frequently run for a year or more before anyone used them. Gartner expects 80% of data and analytics governance initiatives to fail by 2027, and advises tying governance to specific business outcomes. [21]

Meanwhile, knowledge engineers have long recommended building ontologies from the most important concepts outward rather than all at once, [22] and a peer-reviewed "pay-as-you-go" method built an enterprise knowledge graph step by step, driven by business questions, over more than three years in production. [23] We follow that approach, and we make the need for knowledge visible on the organization's own data before anyone pays to build it. In the first four weeks of an engagement we:

1. choose one business decision and write down, with its owner, what goes wrong today and how it is measured;
2. write 20 to 30 questions the agent must answer, with the expert who will judge the answers;
3. build the agent on the organization's existing data and documents only, and run the questions;
4. classify every wrong answer by its cause: a missing definition, an unresolved identity, an unwritten rule, knowledge held only by an expert, a misread goal, an access problem, bad source data or a model error;
5. add the missing knowledge, first from documents and records, then through short sessions with experts, and run the questions again.

The difference between the two runs is the organization's own evidence of what a knowledge layer is worth, and a list of exactly which knowledge to build next.

Two ground rules apply. Personal mailboxes and drives are left out unless the organization clears them in writing. And if experts cannot commit a few hours each, we recommend not starting: an agent can only learn an organization's knowledge from the people who hold it.

This is especially urgent for established organizations. Much of the knowledge that keeps operations running is held by experienced staff rather than written down, and it leaves with them when they retire. Short, structured sessions that turn an expert's judgment into owned definitions and rules serve agents, but they also preserve what the organization would otherwise lose.

After that, the knowledge layer grows one decision at a time. When a second agent reuses a definition or a code mapping, that item moves from the team that built the first agent to the named business owner for that kind of knowledge, with a test and change control. The companion paper, *How to Structure the Knowledge Layer*, describes the method in detail: where the knowledge comes from, how it is matched to an organization's own codes and names, and how it is delivered to agents.

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

## 7. Limits of the evidence

- **Models keep improving.** Queries generated directly from raw schemas have improved sharply since 2023, [8] and stronger models narrow the gap: in one biomedical study, grounding in a knowledge graph lifted a small model by 22 points but GPT-4 by 6. [24] The lasting case for a knowledge layer rests not only on accuracy but also on consistency, explicit refusal outside scope, and answers whose sources can be checked.
- **The evidence is uneven.** Much of it concerns analytics questions over structured data, several studies come from vendors, and deployed results are few.
- **Some effects are unmeasured.** We found no study measuring how resolving duplicate records (entity resolution) changes answer accuracy, and none measuring how much knowledge built for one agent is reused by the next. Nor did we find a published comparison of latency or cost between answering through governed definitions and generating queries from raw tables. We expect governed definitions to be cheaper, because they replace repeated generation and repair with a single compiled query, but that is our reasoning, not a measured result.
- **Structure has costs.** Building and maintaining knowledge takes expert time, which is scarce in most organizations. For simple lookups, plain retrieval is often the better choice. [10]

## 8. Three questions for your organization

1. **Which of your agents' answers could be wrong without anyone noticing?** Would the agent say "I don't know," or give a plausible number?
2. **Where is the definition of your most-used metric written down, and who owns it?** If two systems disagree, which one does the agent use?
3. **If your most experienced planner left tomorrow, which rules would leave with them?** Could an agent find those rules anywhere?

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.

**About.** Prossimo works on data foundations, AI agents and finance transformation for Japanese and international companies. This paper is part of our research program on knowledge and intent in AI. If you work on knowledge modeling, agent governance or data platforms and would like to compare notes, we would like to hear from you.

**Contact.** Shota Ishii, shota@prossimotech.com (essays: shotaishii.substack.com)

## References

[1] Klahr, J., Bhatia, K., Chilukuri, Y. & El-Rayess, N. (2026). "Open Semantic Interchange (OSI) Updates: Specification Now Live, New Working Group Members and More." Snowflake Blog, January 27, 2026. https://www.snowflake.com/en/blog/open-semantic-interchanges-specs-finalized/

[2] Ishii, S. & Rampalli, K. (2026). "Agents Need Purpose, Not Just Permission." Prossimo white paper. http://www.prossimotech.com

[3] Sequeda, J., Allemang, D. & Jacob, B. (2023). "A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases." arXiv:2311.07509. https://arxiv.org/abs/2311.07509

[4] Allemang, D. & Sequeda, J. (2024). "Increasing the LLM Accuracy for Question Answering: Ontologies to the Rescue!" arXiv:2405.11706. https://arxiv.org/abs/2405.11706

[5] Li, J., Hui, B., Qu, G. et al. (2023). "Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs." NeurIPS 2023. https://arxiv.org/abs/2305.03111

[6] Sharma, K., Kumar, P. & Li, Y. (2025). "OG-RAG: Ontology-grounded retrieval-augmented generation for large language models." Proceedings of EMNLP 2025, 32962-32981. https://aclanthology.org/2025.emnlp-main.1674/

[7] Xu, Z., Cruz, M. J., Guevara, M. et al. (2024). "Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering." Proceedings of SIGIR 2024. https://doi.org/10.1145/3626772.3661370. https://arxiv.org/abs/2404.17723

[8] Ganz, J. & Perigaud, B. (2026). "Semantic Layer vs. Text-to-SQL: 2026 Benchmark Update." dbt Developer Blog, April 7, 2026. https://docs.getdbt.com/blog/semantic-layer-vs-text-to-sql-2026

[9] Rumiantsau, M. & Fokeev, I. (2026). "Semantic Layers for Reliable LLM-Powered Data Analytics: A Paired Benchmark of Accuracy and Hallucination Across Three Frontier Models." arXiv:2604.25149. https://arxiv.org/abs/2604.25149

[10] Xiang, Z., Wu, C., Zhang, Q. et al. (2025). "When to use Graphs in RAG: A Comprehensive Analysis for Graph Retrieval-Augmented Generation." ICLR 2026. arXiv:2506.05690. https://arxiv.org/abs/2506.05690

[11] Xu, F. F., Song, Y., Li, B. et al. (2024; updated 2025). "TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks." arXiv:2412.14161. Figure from the updated leaderboard. https://arxiv.org/abs/2412.14161

[12] Yao, S., Shinn, N., Razavi, P. & Narasimhan, K. (2024). "τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains." arXiv:2406.12045. https://arxiv.org/abs/2406.12045

[13] Anthropic (2025). "How we built our multi-agent research system." Anthropic Engineering Blog, June 13, 2025. https://www.anthropic.com/engineering/multi-agent-research-system

[14] Schluntz, E. & Zhang, B. (2024). "Building effective agents." Anthropic Engineering Blog, December 19, 2024. https://www.anthropic.com/engineering/building-effective-agents

[15] Edge, D., Trinh, H. & Larson, J. (2024). "LazyGraphRAG: Setting a new standard for quality and cost." Microsoft Research Blog, November 25, 2024. https://www.microsoft.com/en-us/research/blog/lazygraphrag-setting-a-new-standard-for-quality-and-cost/

[16] Li, Z., Li, C., Zhang, M., Mei, Q. & Bendersky, M. (2024). "Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach." EMNLP 2024 Industry Track. https://arxiv.org/abs/2407.16833

[17] Anthropic (2024). "Prompt caching with Claude." Claude Blog, August 14, 2024 (updated December 17, 2024). https://claude.com/blog/prompt-caching

[18] Liu, N. F., Lin, K., Hewitt, J. et al. (2023). "Lost in the Middle: How Language Models Use Long Contexts." Transactions of the Association for Computational Linguistics. https://arxiv.org/abs/2307.03172

[19] Modarressi, A., Deilamsalehy, H., Dernoncourt, F. et al. (2025). "NoLiMa: Long-Context Evaluation Beyond Literal Matching." ICML 2025. https://arxiv.org/abs/2502.05167

[20] Hong, K., Troynikov, A. & Huber, J. (2025). "Context Rot: How Increasing Input Tokens Impacts LLM Performance." Chroma Technical Report, July 14, 2025. https://www.trychroma.com/research/context-rot

[21] Gartner (2024). "Gartner Predicts 80% of D&A Governance Initiatives Will Fail by 2027, Due to a Lack of a Real or Manufactured Crisis." Press release, February 28, 2024. https://www.gartner.com/en/newsroom/press-releases/2024-02-28-gartner-predicts-80-percent-of-data-and-analytics-governance-initiatives-will-fail-by-2027-due-to-a-lack-of-a-real-or-manufactured-crisis-

[22] Uschold, M. & Gruninger, M. (1996). "Ontologies: principles, methods and applications." The Knowledge Engineering Review 11(2), 93-136. https://doi.org/10.1017/S0269888900007797

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution. 

[23] Sequeda, J. F., Briggs, W. J., Miranker, D. P. & Heideman, W. P. (2019). “A Pay-as-you-go Methodology to Design and Build Enterprise Knowledge Graphs from Relational Databases.” The Semantic Web: ISWC 2019, LNCS 11779, 526-545. https://doi.org/10.1007/978-3-030-30796-7_32

[24] Soman, K., Rose, P. W., Morris, J. H. et al. (2024). “Biomedical knowledge graph-optimized prompt generation for large language models.” Bioinformatics 40(9), btae560. https://doi.org/10.1093/bioinformatics/btae560

Copyright © 2026. KK ProssimoTech. All Rights Reserved. For limited distribution.