Arteus

Enterprise AI Assistants on a Unified Data Ontology

Arteus Insight turns corporate knowledge into answers and actions, Arteus Cue into real-time guidance, Arteus Trainer into workforce training. Use one, or combine them - every product runs on the same foundation: the Arteus Data Ontology.

Companies are entering the AI-fication phase. Automation moved individual operations into software. Digitalization moved entire processes. AI-fication moves knowledge and decisions: every business function - sales, support, document workflows, employee training - gets its own AI assistant that reads corporate knowledge, acts within the permissions of the employee it serves, and carries work through to a finished result in a corporate system.

Access to a strong language model is no longer an advantage; everyone has it. The advantage is a working path from an employee's intent to a verified change in a corporate system - with permitted context, permission checks, executable business operations, and quality control of the result. Arteus delivers that path as a product line where each product is adopted independently:

- **Arteus Insight** - the corporate platform for knowledge, AI assistants, and Agent Skills. From finding information to getting work done.
- **Arteus Cue** - the intelligent qualifier and copilot for sales and support. The right suggestion at the right moment.
- **Arteus Trainer** - the AI simulator for training and assessing customer-facing teams. Practice on the company's real processes.

Underneath them all sits the **Arteus Data Ontology**: a single intelligent layer that unifies documents, conversations, websites, meeting recordings, and corporate system data - with access rights enforced in the platform core, before any model is called.

Insight knows. Cue guides. Trainer coaches. One Data Ontology.

1. Why enterprise AI stalls

Enterprise investment in AI keeps breaking records - 581.7 billion dollars in 2025, double the year before - yet only about a quarter of companies report transformational impact (Stanford AI Index 2026; World Economic Forum with Kearney [1]). The gap is structural, and it has two roots.

Knowledge is scattered

Corporate knowledge lives in disconnected systems: documents, email, messengers, CRM, databases, task trackers, meeting recordings. Each source holds its own slice of context, and no shared layer with common structure, addressing, and access rules exists. Any AI tool bolted onto this landscape sees fragments instead of the whole picture - and building a corporate data ontology by hand becomes a separate multi-year project that must finish before the "real" AI rollout can start.

The last five percent

A strong base model with corporate context attached reaches roughly 95% answer accuracy. That number sounds respectable until it is converted into absolute terms: at thousands of dialogues a month, the remaining 5% means dozens of confident, wrong answers to real customers every week. In a regulated domain, every one of them has a price. And the 95% itself is achieved the expensive way - by loading maximum context into the model for every single question.

The popular fix makes it worse

The market's fashionable answer to both problems is the universal agentic wrapper: a general-purpose agent that "researches" the question in an open loop - read, reason, call a tool, repeat. The results are well known:

- every iteration multiplies model calls, so the token cost per request grows several-fold;
- each step's error feeds into the next, and the agent gradually drifts away from the original question;
- outcomes stop being reproducible from run to run;
- accuracy does not rise with the extra spend - it falls.

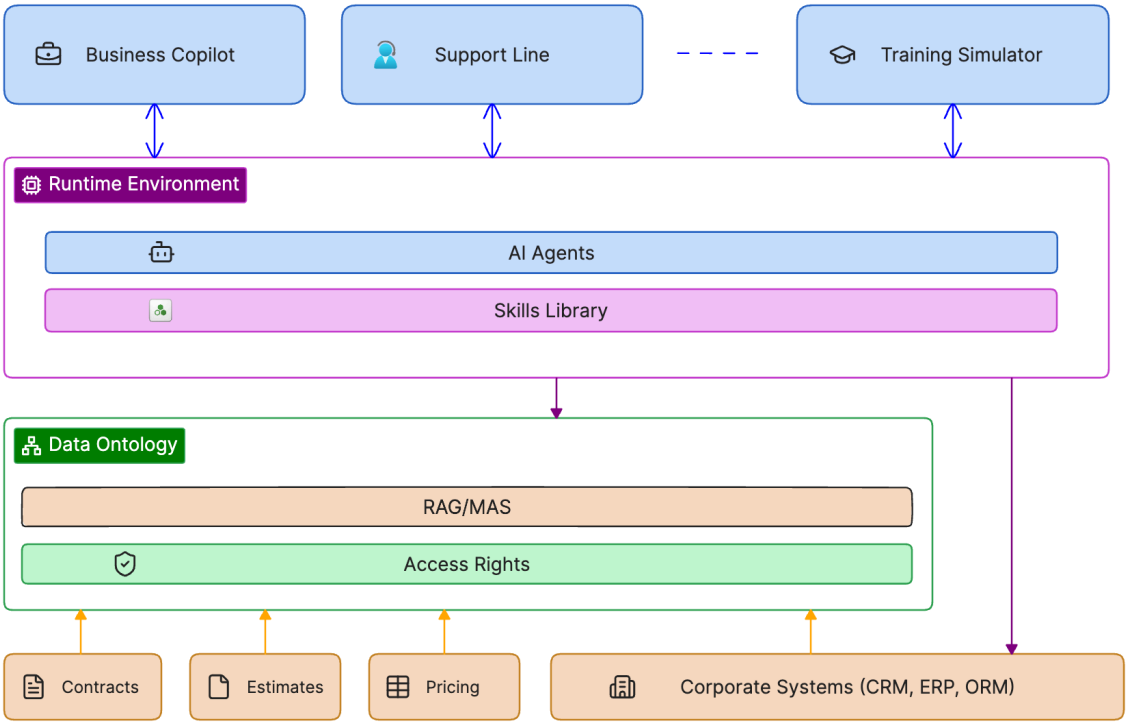
A loop is not a substitute for context. If the system does not already know where the company's knowledge lives, who is allowed to see it, and how it connects, no number of iterations will discover that reliably. The loop will just burn tokens guessing.

2. The Arteus approach: context first, agents governed

Arteus inverts the agentic loop. The platform first assembles the right corporate context from a structured, permission-aware knowledge layer - and only then lets specialized agents and skills act inside it.

Intent → permitted context → execution runtime → permitted skill → verified result.

- **Context is assembled, not rediscovered.** A search planner selects knowledge areas from the Data Ontology, retrieval runs over pre-built structure, and the results are re-scored and packed into a token budget. The system starts from a map, not from a blank page.
- **Agents are specialists with fixed roles.** Search planning, relevance evaluation, section summarization, confidence assessment - each is a dedicated agent with a bounded job, executed in a deterministic stage order. There is no free-running loop to drift or spiral.
- **Cost is bounded by design.** A hard budget of model calls per request makes the cost of every answer predictable - the opposite of an open agentic loop.
- **Permissions are enforced outside the model.** Access to knowledge is computed before the model is called; the right to execute an action is granted outside the model. An assistant cannot expand its own authority.
- **Actions are skills, not improvisation.** A business operation is a versioned Agent Skill with explicit access rights, execution limits, and a designated active version.



AI assistants architecture

To be clear: Arteus is not an argument against agents. Agent Skills and a multi-agent system are first-class citizens of the platform. The difference is where the agents operate - inside a governed ontology with enforced access rights, instead of an unbounded loop over raw data.

Dimension	**Plain agentic loop**	**Arteus: ontology-grounded execution**
Corporate context	Rediscovered by trial and error on every run	Assembled from the Data Ontology, where structure, links, and summaries already exist
Agent design	One general-purpose agent in an open loop	Specialized agents with fixed roles in a deterministic stage order
Errors	Compound with every iteration	Checked per stage; low-confidence output is corrected or refused
Token cost	Unbounded; grows with loop length	Hard model-call budget per request; predictable per answer
Reproducibility	Varies from run to run	Same context, same stages, same order
Access rights	Explained to the model in the prompt	Enforced in the platform core before the model sees anything
Result	Text that may have drifted from the question	A verified answer with sources, or an executed business operation

End to end, a request travels eight links, each of them inspectable:

Link	**What happens**
Input	A message in a connected channel, a schedule trigger, or an external event
Data preparation	Documents, conversations, websites, and meeting recordings are ingested, parsed, split into semantic

Link	What happens
	fragments, enriched with summaries, and vectorized
Context assembly	Knowledge sections are selected, searched, re-scored for relevance, and packed into a token-budgeted window
Execution runtime	The assistant profile, staged request processing, a model portfolio with fallbacks, and dialogue memory
Permission check	Rights are intersected with knowledge sections before the model call; the right to act is checked before a skill runs
Action	A skill executes: a calculation, a document, a record in an external system, a customer reply
Corporate systems	Messengers, email, CRM, task trackers, project management, and external services over MCP
Quality control	Control-question runs, answer scoring, reports, and prompt-level corrections

3. The foundation: Arteus Data Ontology

Corporate knowledge is usually spread across documents, chats, websites, CRM, databases, task trackers, and meeting recordings. Each source stores its own part of the context, so ordinary search - and isolated AI tools - see fragments of the whole. The Arteus Data Ontology merges corporate data into a single intelligent layer, and the company gets it as a platform capability rather than as a multi-year prerequisite project.

The section as the ontological unit

The unit of the ontology is a section of the knowledge tree. Every section performs three functions at once:

- **Addressing.** The system knows which area of the business a piece of information belongs to.
- **Access.** Permissions of employees and assistants are inherited along the section structure.
- **Enrichment.** Summaries, relations, and additional context are built per section.

How the ontology is built

1. Arteus ingests documents, conversations, websites, meeting recordings, and data from corporate systems.
2. The material is normalized into a single structured representation.
3. **LensChunking** splits long documents into semantic fragments and preserves the links between distant parts of the text.
4. Derived summaries are built over documents and sections, giving assistants the big picture of a topic, project, or process.
5. A search planner selects the knowledge areas relevant to the question.
6. Retrieved data is re-scored for relevance and assembled into coherent context.
7. Permissions are checked before the language model is called, so forbidden data never enters its context.
8. The answer keeps its links to source documents and can be verified by the user.

LensChunking: the text between fragments is not thrown away

Conventional document retrieval breaks in a predictable place: the document is cut into fragments, three fragments from the middle land in the answer, and the connective tissue between them is lost. The answer formally cites its source - and still distorts the meaning.

LensChunking works differently. Documents are split into semantic fragments, and everything that lies between them is pre-compressed into **semantic bridges**. At answer time the bridges are inserted between the retrieved fragments, restoring the coherence of the long document: instead of three torn-out passages, the assistant reads the fragments together with a condensed account of what stood between them.

Bridges exist at every distance because summaries are built bottom-up into a **document summary tree**: first per fragment, then pairwise upward, all the way to a summary of the whole document. The mechanism spans neighboring paragraphs as easily as the beginning and end of a two-hundred-page specification.

The approach deliberately balances two known extremes. Classic flat-chunked RAG confuses fragments and loses the meaning between them. Graph RAG restores connections but demands disproportionately expensive indexing and offers weak control over graph integrity as knowledge keeps changing. LensChunking keeps coherence and structure while indexing stays cheap and updates stay incremental - proven on production knowledge bases exceeding 100,000 documents.

Section summaries: the layer above fragments

Above individual fragments, the ontology maintains derived summaries for whole knowledge sections - eight types, each answering its own class of questions: a dense map of section contents, project status, recent discussions, a commercial summary, workspace state, a CRM customer summary, agreements and promises, and representative quotes.

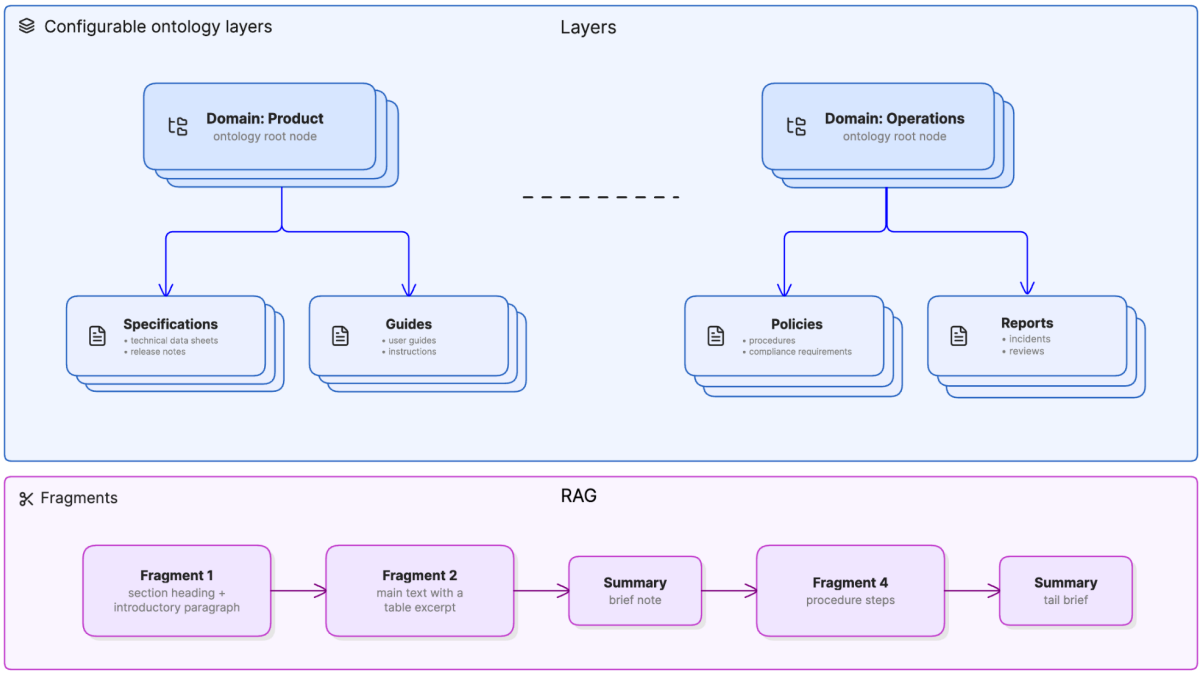
Summaries update incrementally: when new data arrives, only the affected subtree is rebuilt. The practical effect is easy to see - asked "where does the project stand?", the assistant answers with a synthesized project status instead of three random messages from a chat log.

Context assembly for a specific question

The search planner first drafts a hypothetical answer and determines which knowledge sections could contain it. Retrieval then runs over two vector indexes - titles and full text - so a document can be found both by how it is named and by what it says. A dedicated relevance agent re-scores the results. The final window is packed within a token budget: grouped by document, in reading order, with semantic bridges inserted. A hard cap on model calls per request keeps the cost of every answer predictable.

What the Data Ontology delivers

- **Coverage.** One layer spans documents, messages, recorded conversations, and corporate system data.
- **Precision.** LensChunking, section summaries, and multi-stage context selection reduce lost meaning and information noise.
- **Connectedness.** Assistants see the relations between documents, sections, processes, and sources.
- **Freshness.** Changes in connected sources update the corresponding part of the structure.
- **Verifiability.** Users can open the documents and fragments behind every answer.
- **Security.** Access to knowledge is resolved by the user's role before context reaches the model.
- **Reusability.** The same layer powers Insight answers, Cue suggestions, Trainer scenarios, and Agent Skill execution.



Data Ontology schematic view

4. Arteus Insight: knowledge that gets work done

The corporate platform for knowledge, AI assistants, and Agent Skills.

Arteus Insight unifies the company's knowledge and data into the Data Ontology, then turns them into answers and executable business operations. On the platform, a company builds AI assistants for different competencies: each receives its own role, access to the right knowledge, a set of Agent Skills, and explicit boundaries of autonomy.

For the employee, Insight is a single window into corporate context. Through it, one can find information, prepare a document, run a calculation, create a task, update a record in an external system, or launch any other permitted process.

What the product does

- Connects documents, knowledge bases, conversations, websites, meeting recordings, and corporate systems.
- Brings scattered data into the shared structure of the Data Ontology.
- Gives employees precise answers with the sources they rest on.
- Creates specialized AI assistants for sales, support, HR, legal, analytics, and other competencies.
- Assigns each assistant its own knowledge, Agent Skills, permissions, and autonomy level.
- Executes permitted business operations inside corporate systems.
- Supports skill versioning, scheduled runs, and reactions to external events.
- Grows autonomy step by step: from preparing a draft to independently executing approved actions.

Agent Skills: governed capabilities with concrete results

An Agent Skill is a governed business capability of an assistant with a concrete outcome. For example:

- prepare a commercial proposal;
- build a report or a spreadsheet;
- create a task in a work system;
- upload and process a document;
- analyze a sprint or a project;
- update the CRM;
- prepare a suggestion for an employee;
- run an action on a schedule or on an event.

Skills carry versions, access rights, and constraints, with an explicitly designated active version - editing a skill changes nothing in production until the new version is published. Skills execute inside the platform's context environment with direct access to the assistant's permitted corporate context: unlike classic function calling, which invokes tools "blind", a skill sees the retrieval results, section summaries,

and interaction history it is allowed to see. External systems connect over the MCP protocol; the scheduler and event triggers make assistants proactive.

The company gains the ability to develop assistants as digital work roles - while keeping full control over what they are allowed to do.

Key value

- One platform for AI assistants across every department.
- Wide coverage of corporate knowledge and processes.
- High precision through the Data Ontology, LensChunking, and specialized retrieval.
- The shift from finding information to completing work.
- Controlled permissions and boundaries of autonomy.
- Ready Agent Skills reused in new scenarios.

5. Arteus Cue: the right suggestion at the right moment

The intelligent qualifier and copilot for sales and support.

Arteus Cue analyzes the customer dialogue, recognizes the need, and suggests the employee's next move. Every answer and recommendation is built on the company's single Data Ontology: current products, pricing, policies, interaction history, and permitted data from corporate systems.

Precision from shared context

Cue reads the same knowledge layer as every other Arteus assistant. It weighs the content of the current conversation, corporate rules, product characteristics, previous requests, and the customer's data in permitted systems - and hands the employee a suggestion grounded in company facts and the full permitted context of the dialogue.

Verifiable by design

For every suggestion, the employee can open the documents and data fragments used to prepare it, understand where the recommendation came from, and quickly double-check sensitive details before they reach the customer.

What the product does

- Analyzes the live dialogue and the history of the customer relationship.
- Detects need, intent, objections, and buying signals.
- Qualifies the request according to the company's rules.
- Builds a ready-to-send answer from the Data Ontology.
- Shows the sources behind every answer and suggestion.
- Proposes a clarifying question and the next best step.

- Adjusts the length, detail, and tone of the reply on request.
- Produces a final conversation summary and structured data for the CRM.
- Handles approved routine scenarios autonomously.
- Escalates complex or insufficiently confirmed requests to a human - with full context attached.

Key value

- One standard of accuracy for the whole team.
- Verifiable answers with transparent sources.
- Less dependence on the experience of any single employee.
- Instant access to current information during the conversation.
- Consistent lead qualification.
- Fuller CRM records.
- Autopilot that stays under control.

Field-proven: in a pilot with professional sellers on a major marketplace, the suggestion mode navigated large, highly variable product catalogs, mirrored each seller's own communication style, and kept a human confirming every reply - in a channel where answering fast and answering right both decide the deal.

6. Arteus Trainer: practice on the company's real processes

The AI simulator for training and assessing sales and support teams.

Arteus Trainer knows the company's products, processes, and standards. It creates realistic practice dialogues from policies, the knowledge base, and successful conversations; evaluates the employee's actions; and produces personal recommendations.

Training inside the corporate context

Trainer runs on the company's Data Ontology, so the virtual customer asks relevant questions, knows the real product line, and follows the defined process logic. Two linked agents do the work: an examiner agent plays the customer, and a judge agent scores the answers against the same policies and criteria that govern the production environment.

Scenarios in minutes, not weeks

New scenarios are assembled from what the company already has:

- successful calls and chats of top performers;
- product and technical policies;
- sales scripts and support standards;
- typical objections and conflict situations;
- new products, pricing, and process changes;

- real mistakes found in production dialogues.

Accumulated company experience turns into a practical program of training and knowledge assessment - fast.

What the product does

- Simulates a customer with a defined goal, personality, and behavior.
- Generates dynamic scenarios from the company's knowledge and processes.
- Builds training scenarios from successful calls, chats, and policies.
- Supports training, exam, and free-simulation modes.
- Scores answers against configurable criteria.
- Fact-checks the employee's statements against the Data Ontology.
- Analyzes process adherence, qualification completeness, argumentation, and communication quality.
- Shows strengths, mistakes, and growth areas.
- Produces personal recommendations for each employee.
- Gives managers analytics by person, group, scenario, and competency.

Key value

- Training built around the company's real processes.
- Best practices of strong performers become scalable scenarios.
- New policies reach the operational team faster.
- Employees rehearse difficult situations without customer risk.
- Assessment relies on shared criteria and corporate knowledge.
- Mentors spend less time on repetitive training and checks.

7. Trust: permissions, autonomy, and verified answers

The questions that stop most enterprise AI rollouts sound like this: what happens if the assistant sees a document this employee must not see, performs an action this employee has no right to perform, or states a falsehood with confidence? Arteus answers with three independent control loops - over what the assistant sees, what it does, and what it says. All three run in the platform core, outside the model.

Loop one: data access is resolved before the model is called

Employees are grouped into roles. A role defines named permissions, the list of permitted knowledge sections, and the visibility scope of dialogues. Permitted sections inherit down the knowledge tree, and the intersection of requested and permitted sections is computed **before** retrieval: if the intersection is empty, retrieval returns nothing. Source lists and answer sections are filtered as well.

The practical consequence takes two minutes to verify: the same question from two employees with different roles produces two different answers, and a forbidden document physically never enters the model's prompt - so it can be neither retold nor "accidentally" mentioned.

Loop two: the right to act is checked before execution

Before a skill runs, the platform checks who owns the skill, which skills are forbidden to this particular assistant, whether external connections are in read-only mode, and whether personal data must be masked before an external model call.

The key thesis: **the model may propose an action, but it cannot grant itself the right to execute it.** The check lives in the API layer and the skill execution layer - in platform code. Constructions where limits are merely explained to the model in words hold until the first successful attempt to talk around them.

Loop three: confidence is checked before the answer ships

Every answer is bound to its sources. A realtime confidence check intercepts incorrect generations before the user sees them. Regression runs over a base of control questions accompany every knowledge update. And when the knowledge base offers insufficient grounds, the assistant refuses to answer instead of guessing.

This stack has been proven where the price of error is highest: an AI consultant in a regulated accounting-and-tax domain reached **99.5% accuracy on control runs** - with the remainder turning into refusals routed to a human, not into wrong answers. The same deployment absorbs regulatory updates within days and has sustained **4,000 concurrent requests** at a public industry event.

Access control runtime layer

Autonomy as a dial, not a leap

An assistant's independence is set by configuration:

- **Suggest.** The assistant prepares the reply; a human sends it.
- **Confirm.** The action is prepared and executes only after human approval.

- **Bounded autopilot.** Independent execution of an approved skill set, under rate limits.
- **Refuse.** When grounds in the knowledge base are insufficient, the assistant declines to answer.

Companies start with suggestions, accumulate quality statistics, and expand autonomy as trust grows - on the same platform, without re-engineering.

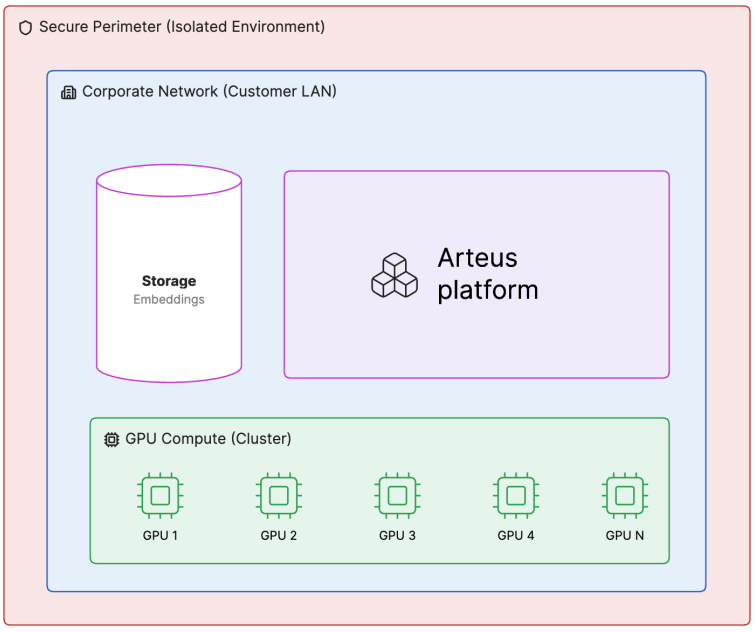
Personal data protection

Masking of personal data runs as a separate stage before any model call, implemented as an autonomous service in the deployment: names, tax and social identifiers, bank account and card numbers, monetary amounts, identity document data, contacts, and addresses are detected and replaced automatically. The module combines a lightweight neural model with hybrid rules, adding microsecond-scale latency on standard CPUs - no GPU fleet required.

8. Deployment and model independence

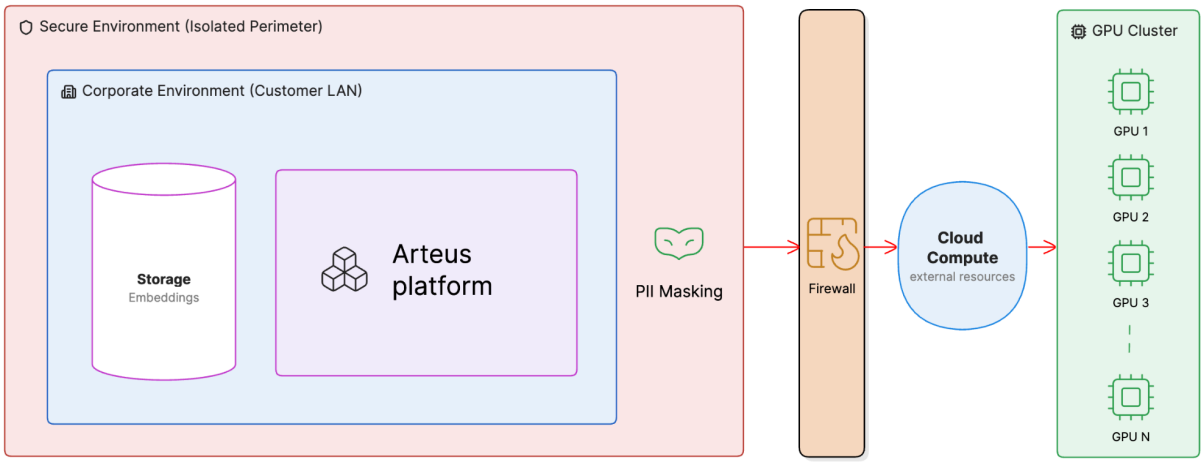
Arteus deploys inside the customer's perimeter in two schemes - on the same platform build. Switching between them is a deployment parameter: assistants, skills, and access policies stay unchanged.

Full private contour. The entire platform, language models included, runs on the customer's infrastructure. There is no exchange with external services; data does not leave the company perimeter in any form. This is the scheme for organizations whose policies forbid any external transfer; it requires GPU capacity on site.



Private deployment

Soft contour. The platform, authentication, permissions, the masking module, and the databases stay inside the customer; only de-identified context - stripped of personal and identifying data - travels to external language models. The scheme grants access to the strongest available models while the data itself stays home. It runs in production today, including an enterprise deployment over a knowledge base of more than 100,000 documents with per-employee access isolation.



Hybrid deployment

No single-vendor dependency

Language models in Arteus are a portfolio, not a commitment. A model is chosen per role - by cost, accuracy, and risk - with fallback provider chains behind every choice: inexpensive, fast models for summaries and service operations, strong models for customer-facing answers. Replacing a model provider is a configuration change, not a rewrite; no workflow depends on the availability of any single vendor.

Cost stays predictable at every level: a hard cap on model calls per request, daily limits per assistant, and hourly and daily limits on skill execution protect the budget from runaway spend by design.

9. One ontology, one product line

Insight knows. Cue guides. Trainer coaches. Every Arteus product runs on the company's single Data Ontology: the same knowledge, the same permissions, the same verified facts - serving answers, real-time suggestions, and training scenarios.

Because the foundation is shared, value compounds:

- every source connected for Insight makes Cue's suggestions sharper and Trainer's scenarios more realistic;
- every Agent Skill built for one process is reusable in the next;
- every access policy is defined once and enforced everywhere - before any model call;
- every quality check strengthens the whole product line at once.

Each product works on its own - none of them requires the others. Start where the value is most visible, and grow from there:

- **Arteus Insight** - create an AI assistant for your process.
- **Arteus Cue** - test Cue on a real dialogue.
- **Arteus Trainer** - create a scenario from your own process.

10. Conclusion

Enterprise AI becomes useful not when a language model is connected, but when that model receives the right context, respects permissions, performs only authorized actions, and returns a verifiable result.

Arteus brings these requirements together around an enterprise data ontology:

- The knowledge tree connects addressing, access, and enrichment.
- LensChunking preserves meaning across long documents.
- Derived summaries provide an up-to-date operational picture.
- Specialized agents assemble and validate context.
- Versioned skills turn knowledge into governed action.

Scattered documents and events become a unified operational layer on which an organization can build advisors, copilots, analysts, training simulators, and proactive agents - without creating a separate data environment for every use case.

Glossary

Data ontology. The Arteus model for organizing enterprise knowledge by connecting section structure, addressing, permissions, and derived data views.

RAG - Retrieval-Augmented Generation. An approach in which a model receives material retrieved from an external knowledge base and uses it as evidence for an answer.

MAS - Multi-Agent System. A system in which narrowly scoped agents perform specialized stages of a workflow.

LensChunking. The Arteus mechanism for preserving continuity in long documents through hierarchical summaries and semantic bridges between retrieved fragments.

Skill. A versioned operation executed by an assistant within assigned permissions and policies.

PII Masking. Detection and redaction of personal or identifying information before context is sent to an external model.

MCP - Model Context Protocol. An open protocol for connecting models and agent systems to external tools and data sources.

Source Note

Technical statements in this paper are based on the architecture and configuration of the Arteus platform.

Market observations reference public sources (Stanford HAI AI Index 2026; the World Economic Forum report prepared in partnership with Kearney, "The AI-First Operating System").

Product metrics reflect measured results of production deployments of the Arteus platform.