The enterprise AI ROI guide for Ops teams

How to solve AI's last mile problem and the token cost crisis

OPENPRISE

The state of AI for enterprise

Enterprise AI has officially entered its awkward phase.

The demos are still magic. The models are still getting smarter. The board still wants AI everywhere, all the time. But in the midst of all that enthusiasm, two uncomfortable truths are showing up in the spreadsheet: **many AI projects are not making it into production, and the ones that do are costing more than anyone planned.**

That gap is not a model problem. It is a **data and infrastructure problem**. In other words, it is very much an Ops problem.

For GTM teams, the issue shows up in two connected ways.

1. **AI agents cannot safely touch mission-critical systems** like CRM, ERP, MAP, and data warehouses without a trusted execution layer in between. IT and InfoSec are not being difficult here. They are doing their jobs. Giving a non-deterministic AI agent direct write access to your CRM is a like handing a toddler the keys to the snack cabinet, except the snack cabinet is Salesforce and the toddler can delete the entire database.

2. AI has a different total cost of ownership than traditional SaaS. In SaaS, usage usually makes the business case look better. **With token-based AI, usage makes the invoice bigger**. That creates a very strange success problem: the more useful the system becomes, the faster it can blow through the budget.

Both problems share the same root cause. **AI is being asked to do too much unstructured, repetitive, data-prep-heavy work at runtime**. It is classifying titles, inferring industries, matching duplicates, filling in blanks, navigating cryptic schemas, and retrying bad outputs caused by bad inputs. None of that is “intelligence.” Most of it is ops data cleanup, and nobody should be paying frontier-model rates to do basic data prep.

The fix is a **governed data and AI orchestration layer** that sits between agents and systems of record. That layer prepares data before it reaches the model, exposes clean custom datasets that agents can actually understand, and executes business rules deterministically through governed workflows, APIs, MCPs, and reusable skill packages.

This is where Ops becomes the hero of enterprise AI production. Not by building another point-to-point integration. Not by becoming a prompting expert. By building the trusted capability layer that humans, apps, and agents can all consume.

This whitepaper explains the AI maturity gap, defines the last mile problem, breaks down the token cost crisis, and gives Ops teams three practical tools to start moving from pilot to production.

1 The AI adoption reality check

A lot of companies say they are “using AI.” That statement is technically true in the same way buying a treadmill technically means you own fitness equipment. The more useful question is: **what is AI actually allowed to do?**

For GTM organizations, AI use cases generally fall into five levels of maturity:

Level	Use case type	What it looks like in GTM	Production risk
1	Search for information	Find company details, summarize web pages, surface account context	Low
2	Create content	Draft emails, landing pages, call notes, ads, social posts	Low
3	Analyze	Review pipeline, identify patterns, summarize campaign performance	Medium
4	Recommend and plan	Suggest account plays, route options, segmentation strategies	Medium-high
5	Execute and take action	Create campaigns, update CRM records, route leads, trigger sequences	High

Levels 1 and 2 are where most enterprise AI adoption lives today. They are useful, low-risk, and relatively easy to approve because they do not require an agent to touch production systems. Level 3 and level 4 begin to use enterprise data, but often in a read-only or manually exported way. The agent analyzes, recommends, and drafts. A human still does the operational work.

Level 5 is where the ROI gets interesting and where the risk gets very real. It is one thing to ask an agent to write a follow-up email. It is another thing to let it create campaigns, update lead statuses, dedupe contacts, or change account ownership.

Level 5 has two flavors

Level 5A execution happens in lower-risk systems such as email engagement platforms, sequencing tools, content systems, or internal task tools. If something goes sideways, it is annoying, but usually recoverable.

Level 5B execution happens in high-risk systems such as CRM, ERP, MAP, CPQ, data warehouses, and other systems of record. This is where enterprise AI is largely stuck. In Openprise field conversations with GTM Ops and IT teams, the pattern is consistent: agents may be allowed to recommend, maybe read, and occasionally act in low-risk systems, but **direct write access to core systems of record is still a hard no in enterprise organizations.**

That is why **AI ROI feels so slippery**. The use cases with the biggest value are also the use cases most likely to require production system access. And if the agent cannot act, someone still has to. Congratulations, you have automated the suggestion box.

Why this matters for ROI

The ROI gap is not just a "we need more AI adoption" issue. Adoption without execution does not fix AI's production problem. It actually makes the cost problem worse.

Traditional software adoption usually gives you a nice story to tell. More users, more usage, more value, same general subscription cost. AI changes that math. With token-based AI, more useful workflows often mean more prompts, more context, more tool calls, more retries, and more metered cost. High adoption is something you cheer for in every other software category. With AI, it can become the budget meeting nobody wants to attend.

This is the structural trap: **AI costs more as it becomes more useful**, and the **highest-value enterprise use cases are the hardest to govern**. The result is a growing pile of pilots that look great in a demo, struggle in production, and make finance wonder why the invoice has developed a personality.

2 AI's last mile problem

AI's last mile problem is the **gap between what an AI agent can do in a controlled demo and what IT will allow it to do in a live enterprise environment**.

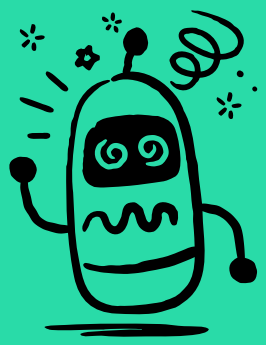
In the demo, the agent finds an account, drafts a message, updates Salesforce, routes a lead, and politely makes everyone feel like the future is already here. In production, the same agent asks for write access to CRM and InfoSec gives it the look. You know the look. The one that says, "Absolutely not, and also who even approved this meeting?"

> "I talked to probably 50-plus customers and prospects about this in the last month. I have seen zero cases, zero, where an agent is allowed to actually update even Salesforce. Period."
>
> **Ed King**
> CEO, Openprise

The 5 parts of the AI last mile gap

The last mile gap is not one problem. It is five problems stacked in a trench coat:

1. **Data governance and access control:** Agents need the right data, but not all the data, and definitely not uncontrolled access to sensitive records or PII.
2. **Prompt injection and security vulnerabilities:** If an agent can be manipulated, it cannot be trusted to act against production systems without guardrails outside the model.
3. **Dirty, fragmented enterprise data:** Most GTM data is duplicated, incomplete, stale, and spread across systems with different formats and definitions.
4. **Inconsistent execution of business rules:** Routing, matching, hierarchy management, campaign naming, segmentation, and lifecycle transitions need to run the same way every time.
5. **Legacy systems without modern AI-ready interfaces:** The systems that matter most were not designed for agents. They were designed for humans, APIs, batch jobs, and occasionally emotional support from your Salesforce admin.



The root issue is deterministic execution. **AI models are probabilistic.** They can be brilliant, but they are not reliably predictable, repeatable, or auditable on their own. That is fine when the task is drafting an email. It is not fine when the task is updating the field that decides whether a hot lead goes to sales today or disappears into the abyss.

Why IT and InfoSec say no

Imagine a CRM analyst whose performance review says: 20% of data entry is wrong, workflow steps are skipped daily, policy compliance is inconsistent, and when asked what happened, the explanation may or may not match reality. Oh and on a bad day, this analyst might delete the database (something an AI agent has actually done).

Would that person get admin access to Salesforce? No. They would get a very short meeting with HR and an escorted walk from security out the front door.

That is how IT and InfoSec evaluate ungoverned agents. People cut AI an amazing amount of slack because the output feels impressive. But when the work requires precision, auditability, and security, “feels smart” is not a control framework.

This is why sophisticated GTM teams still download CRM data into files and then manually load it into AI tools for analysis. It’s clunky. It’s not scalable. But it avoids giving the agent direct access to the system of record. The workaround tells you everything: **the data is useful, the model is useful, and the missing piece is a trusted control layer for enterprise execution**.

The workarounds companies use today

Teams are not ignoring the last mile. They are actively working around it. The problem is that most of these adhoc workarounds do not scale.

Workaround	Why teams try it	Why it fails at scale
Read-only mode	Agents can analyze and recommend without creating write-risk.	Humans still review and execute. The agent becomes a very expensive intern with good ideas and no permissions.
One-off custom integrations	IT can tightly control one agent-to-system path.	Every new use case creates another brittle connector to build, monitor, and maintain.
iPaaS middleware	Existing middleware can sit between agents and systems.	Most tools are built for IT developers, require manual field mapping, and do not understand GTM data models out of the box.
Platform-native skills	Agent platforms provide action builders inside their ecosystem.	Business logic gets trapped in one platform, then rebuilt again for Claude, Agentforce, Copilot, Vertex, Glean, or whatever shows up next quarter.

These approaches are not wrong. But they are incomplete. They do not give Ops teams a repeatable, scalable, governed way to build a proper data infrastructure across agents, systems, and use cases.

What enterprise AI actually needs

Enterprise AI needs a trusted control layer between AI agents and systems of record. That layer should validate actions, prepare data, enforce business rules, control access, maintain audit trails, and expose production systems through safe, purpose-built interfaces.

Think of it like this: the AI models are the brain, AI agents are the nervous system, and skills are the hands, eyes, and legs. Without skills, an agent can only recommend. But with governed skills, an agent can act.

The difference between those two is the difference between “interesting demo” and “production-ready enterprise AI.”

Tactical tool 1

Last mile readiness audit

Use this checklist to prioritize which AI use cases are ready for production and which are still blocked by infrastructure gaps. Rate each area as none, partial, or solved.

1. **Data governance and access:** Can the agent access only the data it needs, with role-based controls and audit trails?
2. **Security and prompt injection controls:** Are guardrails enforced outside the model, not just inside the prompt?
3. **Data readiness:** Are the required fields clean, normalized, deduped, enriched, and easy for an agent to understand?
4. **Deterministic execution:** Are business rules executed by governed workflows, not reinvented by the model every run?
5. **System connectivity:** Can the agent act through approved APIs, MCPs, or workflows without direct system-of-record access?

3 The AI token cost crisis

A traditional SaaS tool has three cost buckets: build and deployment, maintenance, and transaction cost. Build costs are typically understood. Maintenance costs are usually the biggest and most stable line item. Transaction costs are either negligible or predictable enough to forecast.

AI flips that model. The transaction cost, meaning inference and token cost, can become the largest, most variable, and least predictable part of the whole equation.

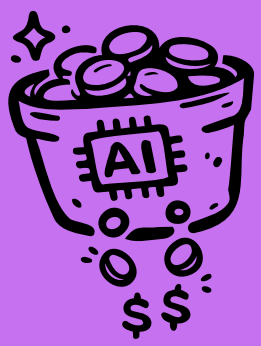
Cost category	Traditional SaaS	AI workflows
Build and deploy	Usually a fixed or one-time project cost.	Can be high, especially with custom agents and forward-deployed engineering.
Maintenance	Usually the largest recurring cost, but relatively stable.	Still present, plus ongoing prompt, workflow, model, and governance tuning.
Transaction cost	Often small, fixed, or easy to model.	Dominant, metered, usage-based, and hard to predict. Every prompt, context window, tool call, retry, and output counts.

This is why AI budgets quickly spiral out of control. The tool works, people use it, and suddenly the math breaks. That is not a failure of enthusiasm. That is a failure of unit economics.

Why token cost is so hard to budget

Token math looks simple from far away. Input tokens plus output tokens times model rate equals cost. Easy. Then real life walks in carrying a giant bag of edge cases.

1. **You do not fully control token consumption.** Reasoning steps, planning phases, hidden tool instructions, long context windows, dynamic documents, and retry loops all consume tokens before the business task is complete.
2. **You may not control which model runs.** Model routers can switch models based on perceived task complexity, which can change cost mid-workflow.
3. **Enterprise automation is usually metered.** Personal subscriptions may cap usage. Enterprise APIs generally charge for every token and tool call. Process one million records by accident instead of one thousand, and finance will notice. Possibly from space.
4. **Cheaper model rates do not always mean lower workflow cost.** If a model is cheaper per token but uses more tokens per task through reasoning, tool calls, or retries, your cost per completed action can still rise.



This is why simply switching to cheaper models or writing better prompts will only get you so far. Those are real levers, but they just optimize the work AI is already doing. They do not eliminate work AI never should have been doing in the first place.

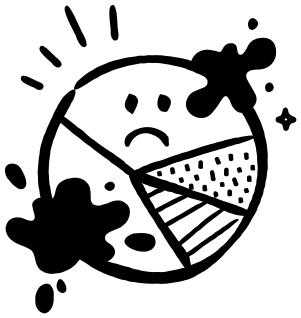
Where your tokens actually go

In GTM workflows, token waste usually hides in four buckets:

Token bucket	What happens	Why it is expensive
Classification	The model determines persona, department, seniority, buying role, lead source, or territory logic.	You are paying a frontier model to sort records into bins. Very fancy bins, but still bins.
Inference and lookup	The model infers industry, company size, technology usage, or missing firmographics because the record was not enriched upstream.	The model repeats lookup and reasoning work record by record.
Deduplication and matching	The model compares records at runtime to decide whether two contacts or accounts are the same.	Identity resolution should be persistent and deterministic, not rediscovered every time an agent runs.
Retries	Bad inputs produce bad outputs, so the workflow reruns. Multi-step agents multiply the cost of each retry.	Retries are the silent killer because they charge you again for the whole chain.

The painful part is that **none of these buckets represent the highest-value work AI should be doing**. AI is great at language, ambiguity, pattern recognition, and synthesizing unstructured context. It should not spend most of its day figuring out that "Sr. Mgr, Demand Gen, EMEA" means a senior marketing contact at a regional level because nobody standardized job titles upstream.

The dirty data multiplier



Dirty data has always been an Ops problem. AI just put a massive price tag on it.

Before AI, a duplicate account or blank firmographic field created operational friction. Someone wasted time. A campaign got messy. A routing rule fired wrong. Not great, but manageable. Now every duplicate, blank field, non-standard title, stale domain, and cryptic custom object can trigger token spend. The junk drawer has not changed. And now it's charging you rent.

That is **the dirty data multiplier**. Bad inputs increase prompt size, reasoning steps, API calls, retries, and create additional verification work. What's worse, **dirty data makes AI outputs less trustworthy**, so adoption stalls and the cost-to-value ratio goes down.

The fix is not to beg the model to be cheaper. The fix is to **stop sending the model work that deterministic automation can handle upstream**.

> "AI is only as cheap as your data is clean. Most enterprises are paying their AI models to do work that should be done deterministically upstream. Asking a frontier model to figure out a contact's seniority, infer a company's industry, or dedupe records at runtime is expensive, error-prone, and completely avoidable. The cheapest token is the one you never spent."
>
> **Ed King**
> CEO, Openprise

Tactical tool 2

Should this task use AI?

Use this decision framework before designing a workflow. The cheapest token is the one you never spend.

Use AI when the task requires language, cultural context, synthesis, ambiguity handling, unstructured input, or web research where the source location is unknown.

Use deterministic automation when the task has predictable inputs, rule-based outputs, clear business logic, low tolerance for error, or needs to run the same way every time.

Ask these questions:

1. Does the task require language understanding or judgment?
2. Is the input unstructured or unpredictable?
3. Is the expected output rule-based and repeatable?
4. What is the cost of an error?
5. Would a rules engine, enrichment workflow, matching process, or API do this faster, cheaper, and more reliably?

If the answer to question 5 is yes, do not make the model do it. AI has enough to do without sorting the operational laundry.

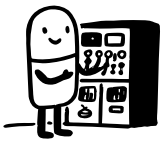
4 Solving both problems with data and AI orchestration

The last mile problem and the token cost crisis look different on the surface. One sounds like an IT and security blocker. The other sounds like a finance and usage problem. Underneath, they are the same problem: **AI is missing a governed data and execution layer**.

That layer is what Openprise provides.

The orchestration layer as solution architecture

Openprise sits between your AI agents and systems of record as a **trusted execution layer for GTM**. Instead of letting an AI agent roam freely through Salesforce, Marketo, HubSpot, NetSuite, data warehouses, enrichment vendors, and internal databases, Openprise gives the agent safe, purpose-built capabilities to call.



The agent still does the work that requires intelligence: interpreting a goal, planning a sequence of actions, summarizing context, and generating language.

Openprise handles the work that requires accuracy: data preparation, matching, enrichment, validation, governance, routing, naming conventions, field updates, API execution, and auditability.

That separation matters. It gives IT a compliant, governed control layer they can inspect. And it gives Ops teams a no-code platform they can own. It gives agents clean data and governed tools. And it gives finance a way to reduce token waste before it becomes an unwelcome budget surprise.

Solving the last mile with the AI Skills Factory

The Openprise AI Skills Factory is the no-code layer where Ops teams build, govern, and deploy reusable AI skills and tools for agents. A skill is not just a prompt. It is a package of logic, data, tools, and guardrails that helps an agent do something safely and consistently.

Capability	What Ops builds	What the agent experiences
Custom datasets	Unified, cleansed, enriched, documented datasets such as Customers, ABM Accounts, Whitespace Accounts, Product Usage, or Event Targets.	A simple, agent-readable data source instead of six custom Salesforce objects and a small archaeological dig.
Custom workflows and APIs	Deterministic workflows that enforce routing logic, hierarchy rules, naming conventions, lifecycle stages, policy checks, and write-back controls.	A safe API call: “create campaign,” “match lead to account,” “update status,” or “route hand raiser.”
Custom MCPs and skill packages	Reusable, platform-agnostic skills that package datasets, APIs, instructions, and guardrails for Claude, Agentforce, Copilot, Vertex, Glean, or future agents.	A governed toolbox the agent can discover and use without direct system-of-record access.

The design principle is reuse. Ops should not build a custom dataset for every single agent use case. That creates the same mess with a more futuristic label. Instead, Ops should build a small catalog of reusable datasets and capabilities that support many workflows. Customers. Target accounts. Open opportunities. Event audiences. Product users. Suppression lists. Intent signals. Buying groups. These are building blocks. Once they exist, agents can use them across use cases.

A good example is event outreach. Today, an events coordinator may ask Ops to pull a target list, enrich it, load it into Marketo, create a Salesforce campaign, brief sales, coordinate sequence copy, manage RSVPs, and update campaign status. With a governed skill, the coordinator can ask an agent for a healthcare dinner list in San Francisco. The agent plans the work, but Openprise handles the deterministic execution behind the scenes: audience building, matching, enrichment, campaign creation, naming conventions, response status updates, and write-back. The result is not "AI writes an email." The result is "AI helps run the workflow, safely."

Solving token cost with four mechanisms

Openprise reduces AI token consumption by moving repetitive data work upstream and doing it deterministically before data reaches the model.

1. **Eliminate LLM-based data preparation.** Classification, enrichment, deduplication, segmentation, and normalization run as deterministic rules. The model stops doing the work that a rules engine can handle for a fraction of the cost.
2. **Compress prompt context.** Clean, normalized, structured data carries more signal in fewer tokens. The prompt gets shorter, examples get cleaner, and the model does not need to read a novella about your CRM schema before doing the task.
3. **Reduce retries.** Cleaner inputs produce cleaner first-pass outputs. That lowers retry rates, which matters a lot in multi-step agent workflows where each retry drags the whole chain behind it.
4. **Replace inference with enrichment.** Pre-enriched firmographic, technographic, intent, and account data means the agent does not have to look up or infer what Openprise already appended.

What customers can expect

Token reduction depends on the workflow, the volume, the starting data quality, and how much LLM-based data preparation Openprise replaces. The ranges below provide a practical planning model for GTM teams.

Tier	Customers	Applies to
Conservative	Up to 30%	General GTM AI workloads touching CRM or account data. Gains come from cleaner inputs, smaller prompts, and fewer retries.
Mid-range	40-60%	Lead scoring, segmentation, personalization, ABM orchestration, account research, and conversational AI grounded in CRM data.
High-value agentic	Up to 80%	AI SDR agents, automated account research, agentic personalization, and AI-powered lead scoring pipelines where Openprise replaces whole LLM preparation steps.

For a representative AI SDR workflow processing 10,000 contacts per month, Openprise estimates the workflow can drop from about 23 million tokens to about 6.3 million tokens when classification, enrichment, and deduplication are handled upstream and retry overhead drops from roughly 15% to under 5%. That is a 73% reduction in token consumption on one high-volume workflow, with better reliability because clean inputs make the agent's job easier.

A useful rule of thumb

Better prompts optimize the work AI is doing. Better data eliminates work AI never should have been doing. One is adjusting the faucet. The other is finding the burst pipe in the basement.

5 Ops must own the AI production problem

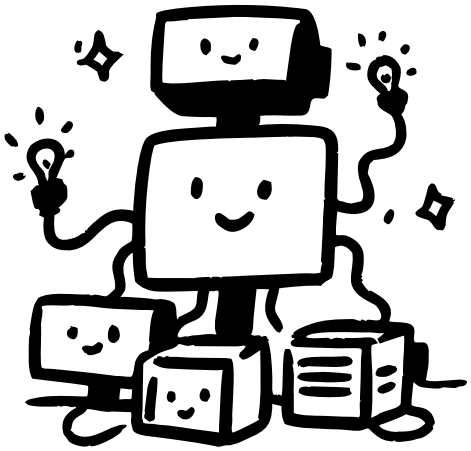
This is not "just an IT problem." IT owns risk. InfoSec owns control. But Ops owns the GTM data model, the workflows, the business rules, and the day-to-day reality of how work actually gets done.

Every high-value GTM AI use case eventually touches something Ops already manages: routing, scoring, enrichment, list loading, territory assignment, attribution, campaign membership, lifecycle stages, buying groups, lead-to-account matching, suppression rules, handoff logic, and system write-back.

That makes **Ops the natural owner of the AI production bridge**. Not because Ops should bypass IT. Please do not put that on a t-shirt. Ops should lead because Ops understands what the capabilities need to do, while IT and InfoSec define the controls required to do them safely.

The mindset shift: capability as a service

For years, Ops teams have wired together point tools. CRM to MAP. MAP to enrichment vendor. Enrichment vendor to data warehouse. Data warehouse back to CRM. It worked, mostly, and then everyone pretended the field mapping spreadsheet was normal.



The agentic world requires a different mindset. **Ops must stop thinking only in point-to-point integrations and start thinking in reusable capabilities**. A capability is something humans, apps, and agents can all consume: "match this lead to the right account," "create a governed campaign," "return current customers in healthcare," "route this hand raiser," "enrich this record using the approved vendor waterfall."

This is similar to the transformation IT went through with service-oriented architecture. The point was not to connect every system to every other system forever. The point was to **create shared services that could be governed, reused, and scaled**. GTM Ops now needs the AI version of that architecture.

What Ops should do now

The last mile gap is not one problem. It is five problems stacked in a trench coat:

1. **Audit current AI use cases** against the maturity model. Separate search, content, analysis, recommendation, and true execution use cases. Be honest about what is actually running in production.
2. **Map last mile gaps for each blocked use case.** Identify whether the blocker is data access, governance, data quality, deterministic execution, or system connectivity.
3. **Establish a baseline token cost by workflow.** Start with high-volume workflows such as AI SDR, account research, lead scoring, segmentation, and personalization.
4. **Identify LLM prep work.** Find every place the model is classifying, inferring, matching, normalizing, or retrying because upstream data is messy.
5. **Build one trusted production skill.** Choose a workflow that matters, define the data, encode the rules, expose a safe API or skill, and get IT/InfoSec sign-off.

The bigger opportunity

The first teams to solve this will be the teams that can scale AI past the pilot. Their unit economics will work. Their IT teams will trust the execution control layer. Their AI budgets will survive contact with actual usage.

The opportunity is bigger than GTM, too. The data catalog and capability layer Ops builds for sales and marketing is the same kind of infrastructure Finance, HR, Legal, and Operations will eventually need. The Ops team that figures out how to build enterprise-wide capabilities for humans, apps, and agents becomes more than a support function. It becomes one of the strategic architects of the business.

Tactical tool 3

90-day AI production readiness roadmap

Days 1-30

Audit and baseline

Inventory AI use cases. Map maturity levels. Assess last mile gaps. Baseline token cost for the highest-volume workflows. Identify where tokens are spent on prep work instead of intelligence.

Days 31-60

Build the foundation

Unify, clean, normalize, dedupe, and enrich the data for one priority workflow. Publish the first custom dataset. Define the governed actions the agent can request.

Days 61-90

Deploy the first production skill

Build the deterministic workflow and API. Package the dataset, workflow, guardrails, and instructions as a reusable skill. Test with IT and InfoSec. Deploy, measure token reduction, and document the model for the next skill.

Enterprise AI should be owned by ops. Not IT

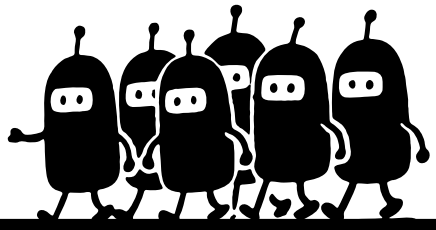
The last mile problem and the token cost crisis are two faces of the same structural challenge: **enterprise AI is being deployed without the governed data and execution layer it needs underneath it.**

Without that layer, agents stay trapped in demos, read-only analysis, and manual workarounds. Token costs keep climbing because models are asked to do repetitive data prep at runtime. IT keeps saying no because ungoverned execution is not safe. Finance keeps asking where the ROI went. Everyone is technically doing AI, and somehow the work still lands back on Ops.

There is a better path. **Put a trusted control layer between agents and systems of record**. Make GTM data clean, unified, enriched, and easy for agents to use. Execute business rules deterministically. Package those capabilities as reusable skills that any agent can call.

That is how AI moves from pilot to production. It is also how Ops moves from ticket-taker to infrastructure architect.

The question is no longer whether enterprises need this layer. The question is whether Ops leads the build or waits for someone else to build a version Ops cannot control.



About Openprise

Openprise makes your GTM data smarter with AI and automation. As the only AI and automation platform built for modern go-to-market teams, Openprise unifies your data silos, orchestrates your processes, and consolidates point solutions so Ops leaders can build smarter GTM data — your data, your way, your timeline. Fortune 500 companies and high-growth enterprises alike rely on Openprise and its partner ecosystem to unlock cleaner data, more efficient operations, and AI-ready pipelines. To learn more, visit www.openprisetech.com and follow us on LinkedIn, X, and Facebook.