Solving the Token Yield Crisis: How Songlines Control® Delivers Execution Efficiency for Enterprise AI

Published by: Cetus AI

Date: June 2026

Reference: CETUS-WP-004

Audience: CIOs, CFOs, and Enterprise Architecture Leaders

Executive Summary

> *"The right question is not how many tokens a system consumes. It is what useful outcome the system produces per token consumed. In other words: token yield."*
>
> — Arvind Jain, CEO at Glean, June 2026

Enterprise AI has shifted from simple chat interfaces to long-running, multi-step agentic workflows. As this shift accelerates, a new operational constraint has emerged: token consumption is scaling exponentially, but business value is not scaling at the same rate. Deloitte's 2025 Tech Value Survey found that enterprises are allocating an average of 36% of their digital budgets to AI, yet organisations like Uber have reported burning through annual AI budgets in a matter of months.

The root cause is architectural. Token usage is rarely driven by the user's prompt alone — it is driven by the scaffolding around the prompt: context retrieval, tool schemas, intermediate reasoning, and execution traces. When this architecture is inefficient, enterprises pay frontier-model prices for routine operational work.

This white paper examines the four architectural levers determining token efficiency and details how **Songlines Control®** — operating across its three deployment modes — acts as the runtime governance and orchestration layer to solve the token yield crisis.

Platform Architecture: One Platform, Three Modes

Before examining the token yield levers, it is important to understand how Songlines Control® is structured. Songlines Control® is a single platform with three operating modes. Each mode unlocks a progressively richer set of capabilities, and organisations can adopt each mode independently based on their current governance maturity and infrastructure requirements.

Mode	Deployment Model	Core Token Yield Capabilities	Governance Posture
Observe Mode	API-layer telemetry. No proxy, no code changes. Deployed in hours.	Real-time token consumption dashboards, cost attribution by user/team/workflow, anomaly detection, immutable audit trail, model usage inventory.	Post-call visibility and evidence. Governance through observation and reporting.
Enforce Mode	Transparent inline proxy. Sits between applications and AI models. Zero SDK changes required.	All Observe Mode capabilities, plus: inline PII redaction before the request reaches the model, payload size enforcement, prompt injection prevention, hard budget limits enforced at the point of execution, intelligent model routing based on cost, complexity, and data residency rules.	Pre-call enforcement. Policy is applied before the request is processed — not retrospectively.
Orchestrate Mode	Full agentic governance layer. Manages multi-step workflows, semantic caching, and sovereign infrastructure routing.	All Enforce Mode capabilities, plus: semantic response caching (eliminates redundant model calls), context window management for long-running agents, multi-model orchestration, audit-ready air-gapped deployment, and sovereign endpoint enforcement for classified workloads.	Runtime orchestration. Governs the full lifecycle of agentic workflows — not just individual requests.

The four token yield levers described in this paper are delivered across these modes. Observe Mode provides the visibility to identify waste. Enforce Mode eliminates waste at the point of execution. Orchestrate Mode compounds efficiency gains across multi-step agentic workflows.

The Token Yield Crisis: An Architecture Problem

The signals are now hard to ignore. Ramp recently reported a 4x year-over-year increase in monthly enterprise AI spend. Deloitte found that more than half of enterprises allocate between 21% and 50% of their digital initiative budgets to AI. Yet in too many organisations, token consumption is rising quickly while business value is not rising at the same rate.

The prevailing instinct is to treat this as a model selection problem — to find a cheaper model or negotiate better API pricing. This instinct is wrong. Token usage is rarely driven by the model alone. It is shaped by the full system around the model: how context is retrieved, how tools are exposed, how work is decomposed, how models are routed, and how prior execution is reused.

If that architecture is inefficient, token spend climbs even when output quality does not. The waste is not in any single prompt. It is in the design of the system. That is why token yield is fundamentally an architecture question — and why solving it requires a governance and control layer that operates at runtime.

The Four Levers of Token Efficiency

Architectural Lever	The Inefficiency Problem	The Songlines Control® Solution	Mode Required
Context Quality	Models process whatever they are given. Poor retrieval forces models to spend token budgets reasoning over irrelevant data rather than solving the problem.	Inline payload optimisation filters noise and redacts PII before the payload reaches the model, ensuring only high-signal context consumes tokens.	**Enforce Mode**
Model Routing	Defaulting to frontier models for routine operational work (search, tool selection, validation) means paying premium prices for commoditised reasoning.	Intelligent Model Routing automatically routes requests based on task complexity, cost thresholds, and data residency rules — zero code changes required.	**Enforce Mode**
Continual Learning	Systems solve the same class of problem from scratch every time, paying the same exploratory token cost repeatedly.	Immutable execution telemetry captures every AI interaction, enabling identification of repeatable workflows and elimination of redundant reasoning loops.	**Observe Mode** (telemetry); **Orchestrate Mode** (semantic caching)
Harness Design	Naive agent harnesses accumulate context endlessly, carrying unnecessary state forward at every step, leading to context bloat and degraded reliability.	Economic Control dashboards provide real-time token consumption visibility by user, team, and workflow — with hard consumption limits enforced inline and context window management for long-running agents.	**Observe Mode** (visibility); **Enforce Mode** (hard limits); **Orchestrate Mode** (context management)

Lever 1: Context Quality — Eliminating the Hidden Tax of Noise

The Problem

A short user instruction can trigger a massive token bill. In agentic systems, the visible prompt is often dwarfed by the system instructions, retrieved documents, and tool schemas injected into the context window. A prompt like "Analyse churn risk for these accounts and create follow-up tasks" may appear small, but the actual token load includes system instructions, tool schemas, retrieved documents, intermediate reasoning, execution traces, and memory.

In many enterprise systems, most of the tokens are not typed by the user at all. They are generated by the scaffolding around the task. If the retrieval architecture is noisy, the model is forced to act as a filter rather than an engine of insight — spending its token budget reasoning over irrelevant or conflicting information instead of acting on the right signal.

> *"Weaker retrieval forced the system to compensate with more tool calls, more reasoning loops, and more over-fetching. That is the hidden tax of poor context architecture."*
>
> — Arvind Jain, CEO at Glean

The Songlines Control® Response — Enforce Mode

This lever is addressed by Songlines Control® operating in **Enforce Mode**. In Enforce Mode, the platform acts as a transparent inline proxy sitting between enterprise applications and AI models. Before a request is processed, the platform intercepts the payload and applies inline redaction for sensitive data — protecting compliance while simultaneously reducing payload size. Organisations can enforce payload size limits and content quality rules at the infrastructure layer, ensuring that only high-signal, compliant context reaches the model. The result is a direct reduction in the "noise tax" and a dramatic improvement in the ratio of useful output to tokens consumed.

Critically, this enforcement happens before the request reaches the model — not retrospectively. No application code changes are required. The proxy is transparent to both the application and the model.

Lever 2: Model Routing — Right-Sizing Intelligence

The Problem

A large share of enterprise AI work is operational: search, retrieval planning, tool selection, validation, and execution management. These steps are critical, but they do not require the reasoning capabilities of a frontier model. When every step in an agentic workflow defaults to GPT-4o or Claude 3.5 Sonnet, the enterprise is effectively paying frontier prices for routine work. As usage scales, this becomes one of the most significant drivers of AI cost inefficiency.

The Songlines Control® Response — Enforce Mode

Model routing is also an **Enforce Mode** capability. Songlines Control® features an Intelligent Model Routing engine that acts as a dynamic switchboard within the inline proxy layer. It evaluates the incoming request against enterprise-defined policies and routes it to the most appropriate model — before the request is forwarded. Routine summarisation or data extraction is routed to faster, cheaper models. Complex reasoning tasks are preserved for frontier models. Sensitive data is automatically routed to sovereign, locally-hosted instances.

Routing Rule Type	What It Does	Token Yield Impact
Cost-based routing	Routes requests below a complexity threshold to lower-cost models automatically	Average 40% cost reduction on mixed workloads
Latency-based routing	Routes time-sensitive requests to faster, lighter models	Reduces end-to-end workflow time without sacrificing quality
Sovereignty routing	Ensures regulated data never leaves Australian data residency	Eliminates compliance risk while maintaining operational efficiency
Budget threshold routing	Automatically falls back to cheaper models when monthly budget thresholds are reached	Prevents budget overruns without interrupting service delivery

All routing is executed entirely at the infrastructure layer — with zero changes to the underlying application code.

Lever 3: Continual Learning — Compounding Execution Efficiency

The Problem

Human workers do not solve the same problem from scratch every time. They document processes, reuse successful approaches, and build institutional knowledge. Enterprise AI systems, however, often pay the same exploratory token cost repeatedly for similar tasks. Every execution produces signal about how similar work should be done next time — which tools were useful, which retrieval path worked, which steps were unnecessary. If that signal is not captured and reused, the system keeps paying the same exploratory cost again and again.

The Songlines Control® Response — Observe Mode + Orchestrate Mode

Continual learning spans two modes. The foundation is **Observe Mode**: through its Immutable Audit Trail, Songlines Control® captures a cryptographically signed record of every AI interaction — including the prompt, the model used, the tokens consumed, the latency, the policy decision applied, and the output. This rich telemetry provides the data infrastructure for continual learning. Enterprise architecture teams can analyse this data to identify highly repeatable workflows, refine system prompts, and understand where token spend is concentrated.

The compounding efficiency gains are then realised in **Orchestrate Mode**, which adds semantic response caching. When a request is semantically equivalent to a prior interaction, Orchestrate Mode returns the cached response — eliminating the model call entirely. For organisations running high-volume agentic workflows, semantic caching alone can reduce token consumption by 20–40% on repeatable task classes.

The best enterprise AI systems will compound in this way. Each completed task should improve the economics of the next related one. Songlines Control® provides both the data infrastructure (Observe Mode) and the execution infrastructure (Orchestrate Mode) to make that compounding possible.

Lever 4: Harness Design — Managing Context Bloat

The Problem

As agents take on longer-running, multi-step work, the harness — the system managing the agent's state — becomes a major determinant of both quality and cost. A naive harness keeps expanding the active context window. It carries more instructions, more tools, more state, and

more intermediate outputs forward at every step. Cost grows as the workflow grows. Reliability usually degrades too.

The Songlines Control® Response — All Three Modes

Harness design is addressed progressively across all three modes. **Observe Mode** provides the visibility layer: through its Economic Control dashboard, organisations gain real-time, month-to-date visibility into token consumption at the user, team, and workflow level — making context bloat visible before it becomes a budget incident. **Enforce Mode** adds hard consumption limits enforced inline, so that when a specific agent or workflow begins exhibiting context bloat, administrators can set automatic thresholds that block or reroute runaway workflows. **Orchestrate Mode** addresses the root cause directly, with active context window management that prunes unnecessary state from long-running agentic workflows — reducing the token cost of each step without degrading output quality.

Economic Control Capability	What It Provides	Mode	Business Impact
Real-time MTD spend dashboard	Month-to-date token consumption and cost by model, user, and workflow	Observe	CFO-ready AI cost reporting without manual aggregation
Hard budget limits	Automatic enforcement of spend thresholds per team or workflow	Enforce	Prevents the "Uber scenario" — burning through annual budgets in months
Cost attribution	Every token attributed to a specific user, workflow, and business unit	Observe	Enables chargeback and accurate ROI measurement by AI initiative
Anomaly detection	Alerts when token consumption patterns deviate from baseline	Observe	Early warning system for runaway agentic workflows before they become budget incidents
Context window management	Active pruning of agent state to reduce per-step token cost	Orchestrate	Reduces long-running workflow costs by 20–40% without degrading output quality

Conclusion: Execution Efficiency is the Real AI Moat

The CIOs and CFOs who succeed in the next phase of enterprise AI will not be those who simply deploy the most models. They will be those who master execution efficiency.

Token yield is fundamentally an architecture question. It requires a control plane that can make waste visible (Observe Mode), eliminate it at the point of execution (Enforce Mode), and compound efficiency gains across multi-step agentic workflows (Orchestrate Mode).

Songlines Control® delivers this control plane as a single platform. Organisations can start with Observe Mode — deployed in hours, with zero infrastructure changes — and gain immediate visibility into where token spend is concentrated. From there, Enforce Mode adds inline enforcement that eliminates waste before it reaches the model. Orchestrate Mode completes the

picture, adding semantic caching, context management, and sovereign infrastructure routing for the most demanding agentic workloads.

The organisations that build this infrastructure now will not just reduce their AI costs. They will define how their enterprise competes in the AI era.

To see how Songlines Control® can address the token yield challenge in your organisation, contact us to request a 14-Day Observe Mode Baseline engagement.

Contact:
Cetus AI | Brookwater, QLD
hello@cetusai.com.au | www.cetusai.com.au