

Insights



A framework for AI token optimisation

Learn how to optimise token expenditure, reduce technical debt and elevate enterprise AI maturity.

hippodigital.co.uk



Introduction

As organisations transition from localised generative artificial intelligence (AI) experimentation to autonomous, agentic workflows, the expenditure of running models on cloud platforms can scale rapidly and non-linearly.

Addressing this financial and operational challenge requires moving beyond reactive prompt adjustments toward a systematic governance discipline integrated directly into your Software Development Lifecycle (SDLC).

The billing paradox and token waste

There is a key misalignment between the use of AI and the actionable outputs gained. While individual API calls can appear to be cost-effective, when aggregated into automated workflows costs things can escalate unexpectedly. This billing paradox is driven by three primary mistakes that often demand manual rework and increased cognitive load, while offsetting any productivity gains:

1. Deep document dumps

Feeding AI with massive documents, raw and unstructured data and multi-page transcripts without properly scoping the sub-task's specific needs.

2. Unoptimised prompts

Providing overly general large system prompts that try and encapsulate all possible domain scenarios, due to lack of central skills and prompt management

3. Long chat chains

Unnecessary back and forth with AI models, reusing the full conversation each time, causing bills to process exponentially increasing conversations.

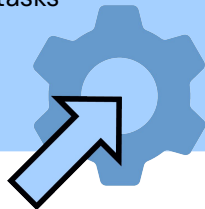


The Token-to-Value Framework

Evaluating the health of your AI on a cost-per-token metric or monthly bill is ultimately a flawed approach. Performance must be viewed as a balance between output quality and the compute efficiency and operational overheads required to achieve it. Our token-to-value framework aims to rationalise this.

The value layer

Value depends on task importance and the human effort needed to reach production readiness. For instance, generating a detailed Statement of Work is high value but requires legal review, making zero-shotting realistic only for lower-stakes, routine tasks like text summarisation.



The waste layer

We've identified the three primary drivers of AI waste, which compound when input-to-output ratios are large or when repetitive corrective loops occur. Note that "waste" is relative; while higher-paid staff might justify waste through higher hourly potential savings, this approach misses the broader economic point.



3 core disciplines for token optimisation

Achieving sustainable AI efficiency relies on embedding three targeted operational controls directly into your engineering workflows and SDLC:

1. Task decomposition

Replace monolithic system prompts that inflate token ingestion and degrade instruction following.

- **Modular skills:** Isolate workflows using sub-agents for discrete sub-tasks
- **Centralise repositories:** Version control and audit prompts as foundational assets
- **Targeted state sharing:** Programmatically extract exact files or APIs to create precise scaffolded context

2. Tool-agnostic architecture

Avoid coupling prompts to a single vendor to preserve flexibility and limit cost.

- **Dynamic routing:** Route formatting tasks to cost effective models and preserve frontier models for complex tasks
- **Prompt caching:** Cache deterministic context and static schemas
- **Model context protocol (MCP):** Fetch isolated data snippets on demand rather than passing static context dumps

3. Governance

Complement technical optimisation with structured organisational governance.

- **Evaluation frameworks:** Assess prompts and outputs on every change to ensure conciseness
- **Peer Validation:** Review AI-generated assets prior to merge to avoid expensive token consumption in debugging
- **Anti-Bloat Limits:** Reject unnecessarily verbose AI code or duplicate logic in CI/CD pipelines



Moving toward a token-optimised state

Adopting these disciplines fundamentally shifts enterprise architecture from an unoptimised to an optimised state:

Architectural Dimension	Unoptimised State	Optimised State
Context Management	Uncurated context dumping; full repository or transcript ingestion.	Isolated, task-specific state injection via deterministic scaffolding.
Prompt Design	Monolithic, multi-page system prompts; ad-hoc individual prompting.	Modular, version-controlled prompt repositories managed as software assets.
Model Selection	Uniform usage of top-tier frontier models for all tasks regardless of complexity.	Abstraction layer featuring dynamic model routing based on task difficulty.
Interaction Pattern	Multi-turn conversational thrashing and iterative prompt corrections.	Zero-shot or single-pass execution supported by precise context inputs.
Infrastructure Cache	Re-ingestion of identical prompt contexts on every API call.	Systemic prompt caching and Model Context Protocol (MCP) data fetching.
Downstream Impact	High code verbosity, increased technical debt, low output fidelity.	Maintainable codebases, zero-shot production readiness, high token efficiency.



Phased operational deployment

Implementing the Token-to-Value framework and enabling fiscal governance over compute resources requires a structured, phased operational deployment:

1. **Consolidated telemetry via AI gateways:** Standardise model interactions by routing traffic through an enterprise-grade gateway. This allows for the non-intrusive tracking of consumption metrics, including model selection, throughput volumes, and iterative conversation thrashing.
2. **Fidelity feedback integration:** Embed lightweight feedback loops within developer environments to quantify artifact usability. By categorising the level of manual intervention required post-generation, organisations can calibrate their Fidelity Index with empirical data.
3. **Unit-level performance benchmarking:** Codify departmental efficiency targets to identify resource-heavy outliers. Significant token value variance serves as a signal for targeted intervention, such as prompt refinement workshops or the decomposition of monolithic workflows.
4. **Automated bloat prevention:** Establish organisational bloat limits, by creating automatic checks in CI/CD pipelines and prompt evaluation frameworks. By rejecting verbose or redundant AI-generated logic, leadership can preserve architectural integrity and long-term codebase maintainability.

Scaling generative AI doesn't need to go hand in hand with growing API bills. By shifting from ad-hoc prompting to an structured discipline focused on Token-to-Value optimisation and embedded in design and architecture practices organisations can systematically reduce API expenditures, elevate output quality, and build durable, highly governed digital capabilities.

Hippo can help audit your AI delivery model, find opportunities to optimise your token spend, embed best practices and establish robust AI governance. [Get in touch to speak to our AI team.](#)

Colin Bull, Director of AI

