Blog · 8 November 2026 · 29 min read

Corporate arithmetic of AI transformation: how not to burn all your payroll savings on tokens, subscriptions and servers

A CEO and a CFO need to understand a little of the technology and economics of AI: why heavy development burns almost all of a company's tokens, how to get through the double-cost pit, and when your own GPU cluster pays off. With a financial model in Excel.

TL;DR. A CEO and a CFO need to understand a little of the technology and economics of AI. Buying subscriptions and API tokens without a hard financial plan will raise the IT budget by about 5 %, and at Revolut-level spend by far more. At the start of a transformation a business meets four "demons": the illusion of cheap code, the double-cost pit, the provider price trap, and the risks of availability and regulation. In my model about 99 % of a company's tokens are burned by heavy development, where strong models and long context are needed. My solution is the "Dual Compute" model: move heavy development onto your own GPU nodes and leave light workloads and operational processes to cloud providers. I first wrote this article in Russian, for a market where US frontier APIs are not available. This version uses Western prices and Western salaries, and it was revised after peer review by industry colleagues. While I was translating it, Revolut published numbers that check my model against reality - see the end.

The million-dollar question: the reaction to my previous article

After my previous article on cutting up to 50 % of corporate IT headcount by 2028, colleagues from bank boards and large holdings started writing to me privately. If you put their questions together, you get roughly this:

"Sabynin, your forecast is pretty and hyped. But have you ever opened the corporate bills for tokens and subscriptions? Many have already started "playing" with this and know what the "AI toy" costs. If tomorrow I give every employee a notionally unlimited plan and do not get the payroll cut, I will not save anything - I will simply lose control of spending and wreck the P&L. Payroll, as usual, stays where it is, and a cloud bill of several hundred thousand dollars a month lands on top. So where is your economics? Who is ready to stake their job on everything will go according to your plan?"

And yes, the question is absolutely fair. It is a real management fear, and behind it lies the practice of running IT.

If you run AI transformation as a free handout of convenient tools to employees (as, for example, Meta did), you will not see any reduction in operating costs. At Meta, employees competed on an internal leaderboard over who would burn more tokens, and at first management said "Keep doing it. No limit". Later, as costs reportedly headed towards billions, Meta started managing token use. According to McKinsey, in 2026 93 % of the companies surveyed have already exceeded their AI budgets. My reading: unchecked, compute will "eat" the whole operating margin.

To avoid "burning" shareholders' money, you have to separate comfortable experiments for employees from hard corporate arithmetic from day one.

On the way, in my view, you will meet four "demons" of AI transformation costs. Let us take them one by one.

Demon one: the illusion of cheap code, or why heavy development burns about 99 % of tokens

Suppose you are a company (a mid-size bank) of 2,500 people, of whom about 500 are IT (Change and Run). The numbers are approximate, but typical. In any corporation AI adoption starts with office enthusiasm: marketing generates slogans and hundreds of creatives, lawyers proofread contracts, HR sets up chatbots to filter candidates. Management sees thousands of AI requests a day and concludes: "AI has already spread across the company on its own, time to buy a corporate plan". That is a mistake.

Before we look at the numbers, three basic terms in plain words, without which a CFO and a CEO cannot control the spend:

- **Compute** - the physical computing "brains": servers, specialized tensor accelerators (GPUs) and ultra-fast memory. They crunch billions of matrix operations around the clock so that we see the generated text or code. In the cloud you pay for someone else's compute - the capacity the provider bought and rents out to you with a markup. With your own node you buy a server with accelerators once and put it into a data-center rack.
- **Context (context window)** - the AI agent's "working memory" for a given request: how much text and code the model can hold in its head at once, measured in tokens. An office chatbot for a lawyer or HR needs 8,000-16,000 tokens. Top models advertise from 256k to 1M tokens, but in practice an honest 128k-256k for the core development team is more than enough for complex tasks.
- **Token** - the unit of data and billing that you are invoiced for. In essence it is a syllable or a piece of a word: in English one token is about four characters. When the model reads an incoming request, a monorepo or a document, you pay for input tokens. When it formulates an answer or generates code, you pay for output tokens, which at commercial providers cost several times more. At company scale the bill runs into billions of tokens a month.

All corporate IT splits into two fundamental contours: **Change** (development) and **Run** (support). If you lay this structure over the arithmetic of token consumption, you get a telling picture:

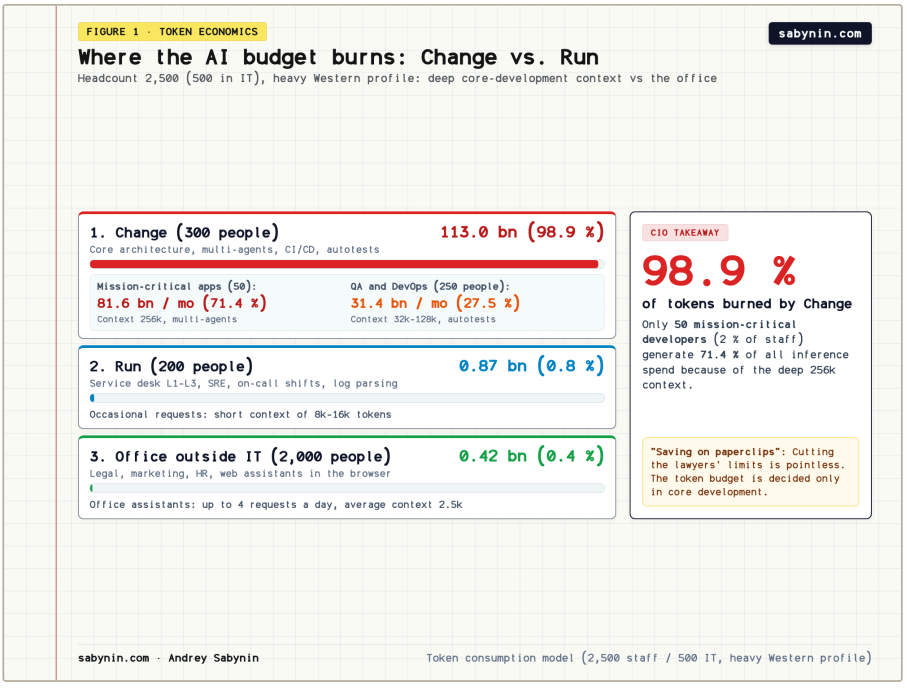


Figure 1. Where the company's AI budget burns: Change vs Run

Company summary (the detailed model with formulas for each unit, request norms and input/output prices is in the file **ai-infrastructure-spec-2026.en.xlsx**, Sheet 1):

Contour and role	Headcount (share)	Load profile (21 working days)	Average agent context (input + output)	Volume per month	Share of tokens
1. Change: mission-critical apps and architecture (multi-agent chains)	**50** (2 %)	300 agent calls/day	256k + 3k tokens	**81.6 bn**	**71.4 %**
2. Change: QA, analysis, DevOps (autotests, CI/CD, requirements)	**250** (10 %)	125 calls/day	46k + 1.8k tokens	**31.4 bn**	**27.5 %**
TOTAL CHANGE	**300 (12 %)**	—	**Deep context**	**113.0 bn**	**98.9 %**
3. Run (L1-L3, SRE, logs and incidents)	**200** (8 %)	20 calls/day	9.5k + 0.9k tokens	**0.87 bn**	**0.8 %**
4. Office outside IT (marketing, sales, legal, HR - chat in the browser)	**2,000** (80 %)	4 requests/day	2k + 0.5k tokens	**0.42 bn**	**0.4 %**
TOTAL COMPANY	**2,500 (100 %)**	—	—	**114.2 bn**	**100 %**

In total the company runs through **114 bn tokens a month**, of which the Change contour burns **113 bn (98.9 %)**, while all of Run plus the two-thousand-strong office together use only 1.3 bn (about 1 %). Fighting for savings by cutting the office's limits is classic saving on paperclips. You have to manage development and its efficiency first.

This is the Western profile, five times heavier than the modest profile of my Russian version. It is not my fantasy. Western developers have got used to burning a lot of tokens, and the vendors themselves confirm it:

- In July 2025 Anthropic introduced weekly limits for Claude Code after "one user consumed tens of thousands in model usage on a $200 plan"; the limits affected less than 5 % of subscribers.
- Cursor moved its $20 plan from request limits to compute billed at API rates in June 2025, and then refunded users surprised by the bills.
- GitHub moved Copilot to token-metered AI Credits from June 2026, saying that "agentic usage is becoming the default" and the old model was "no longer sustainable".
- At Meta, an internal "Claudeonomics" leaderboard showed more than 60 trillion tokens in 30 days (later reports give 73.7 trillion), with the top user at 281 billion.
- At Revolut, before they switched to a self-hosted model, engineers spent "$1,000 a month, $10,000 a month" each on AI.
- Independent measurements differ by method. Jellyfish (12,000 developers in 200 companies, Q1 2026) gives a median of 51 million coding tokens a month and a p90 of 380 million. Larridin (August 2026) gives a median billed spend of $213 a week per engineer, about $920 a month.

In my model a developer of mission-critical apps burns about 1.6 bn tokens a month. Honestly, that is about four times Jellyfish's p90; in dollars it matches Larridin's median spend only if caching works, and it sits at the lower end of Revolut's range. QA, analysts and DevOps use about 0.13 bn each, between Jellyfish's median and p90. Whether the flat subscriptions themselves lose the vendor money is debated. SemiAnalysis estimates that subscription margins are far below API margins, and that a plan stays profitable only while users stay on cheaper models or use a small share of their limits. Either way, flat plans hid the real consumption, and as vendors switch to metering, the bill follows.

And note: in development a weak AI model is a disaster. In the complex Change cycle a weak model without deep context does not hold the architecture, piles up hacks and spaghetti code (and then outsourcing vendors come to clean up the "mess" at three times the rate). That is exactly why developers quietly paste your work code into their personal Claude and ChatGPT accounts, creating security risks - simply because these models really do handle complex tasks.

By late 2026 engineers simply need an honest 128k-256k context for AI agents on strong flagship models.

Where does the bill come from? At list prices without caching (Claude Sonnet 5.5: $2 per 1M input tokens, $10 per 1M output) the API for the Change contour costs **about $243k a month** - roughly $2.9M a year. With prompt caching, where the agent re-reads the same repository and about 90 % of the input comes from cache, the bill drops to $53-67k a month for mid-tier models (OpenAI gpt-6.1 (mid-tier), Claude Sonnet 5.5, gpt-5.3-codex) and about $118k a month for Claude Opus 5.5 (links to prices at the end). The CFO will not be pleased by either number, although against a $5M monthly IT payroll it does not look scary - until usage grows.

Demon two: why the timing of the transformation matters (the double-cost trap)

In consultants' presentations everything is smooth: AI introduced, productivity up, headcount down 30-50 %, costs down. In real life the CEO and the CFO fall into the classic "double-cost pit".

An example. Perimeter: the IT of a bank or a holding with 500 people. At a fully loaded cost of $10,000 per specialist per month (a European banking level: about €80k gross plus about 22 % employer contributions and overhead), the base payroll is about **$5 million a month**.

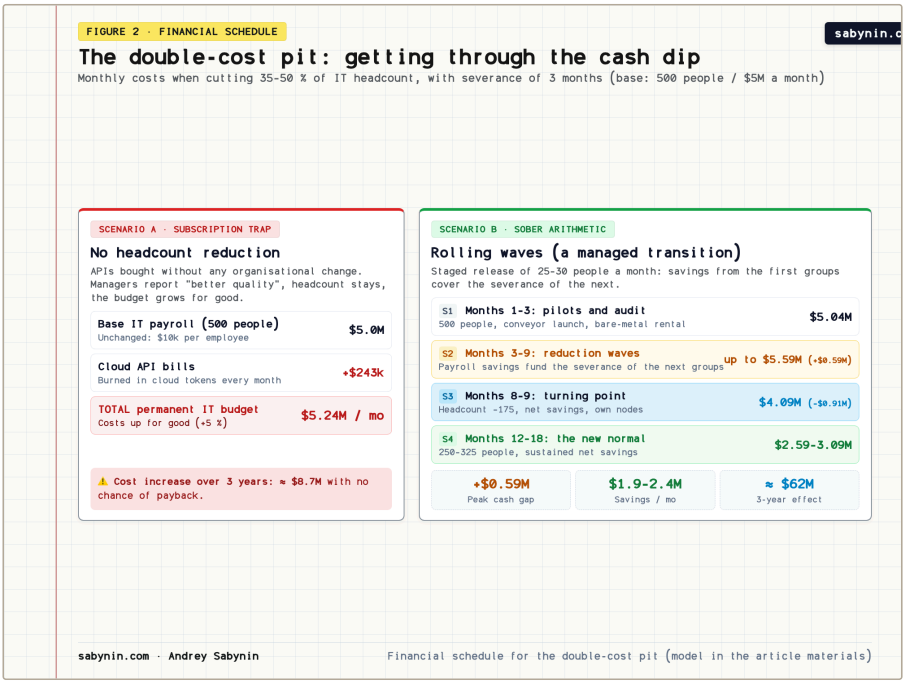


Figure 2. The double-cost pit: getting through the cash dip of a payroll optimization

Comparison of three transformation scenarios (the month-by-month schedule of payroll, severance and the self-financing of waves is in ai-infrastructure-spec-2026.en.xlsx, Sheet 3 "Payroll & waves"; severance is assumed at 3 months of salary - adjust to your jurisdiction):

Parameter (base: 500 IT, payroll $5M/month)	Scenario A: "Subscription trap"	Scenario B1: "Two rounds of layoffs" (typical mistake)	Scenario B2: "Rolling waves + lease" (target)
Optimization mechanics	Headcount is not cut ("people work better")	175 people laid off in two rounds, months 4 and 6	25-30 people a month from month 3 to month 9 (self-financing)
Infrastructure	Subscriptions and API for everyone	Cloud until month 7, then a one-off purchase of two nodes ($1.15M)	Day 1 - bare-metal rental, then a 3-year lease of the nodes

Parameter (base: 500 IT, payroll $5M/month)	Scenario A: "Subscription trap"	Scenario B1: "Two rounds of layoffs" (typical mistake)	Scenario B2: "Rolling waves + lease" (target)
Peak cash gap	**+$243k a month**, forever	**+$1.9M** in month 4	**+$0.59M** in month 3 (severance paid from the savings of earlier waves)
Legal and HR risk	Zero (the money simply burns)	Critical: mass layoff, collective-redundancy rules, seniors leave	Lower: small monthly waves, 50 seniors protected by a retention program; still check WARN Act (US) and collective-consultation (EU) thresholds - waves of 25 are not automatically below them
Monthly budget a year later	**$5.24M** (+5 % to base)	**$3.3M** (if the business survives the shock)	**$3.1M** excluding severance, and **$2.59M** from month 15 (-48 %)
Change in costs over 3 years	**Cost increase of about $8.7M**	**Cost decrease of about $47M** (with a high risk of stopping delivery)	**Cost decrease of about $62M** (without expensive bridge loans)

The main secret of the transition is **rolling waves and self-financing**: headcount is reduced not in one go, but by 25-30 people a month from month 3 to month 9, as each team moves onto the AI conveyor. Payroll savings from the first cohorts (by month 5 already about $750k a month) gradually "pay" the severance of the next ones, squeezing the cash gap from +$1.9M down to a manageable +$0.59M without expensive bridge loans.

In scenario A I take the API bill for the heavy profile at list prices, $243k a month on top of the same payroll. At the level Revolut reported, $1,000-10,000 a month per engineer (The Register), the same 300 people in Change would cost $0.3-3M a month.

And so that 50 key architects and lead engineers do not quit at the first rumours, before the start they are moved into a protected category of lead AI architects. Team leads' bonuses are decoupled from the number of their reports and tied to the cost of delivering a change into production. Directing 10-15 % of the saved payroll into a retention fund (+25-30 % to the salary of those who stay) turns resistance into internal selection by the teams themselves.

Demon three: dependence on someone else's infrastructure, or how not to fall into the price trap

A classic platform expansion is unfolding on the market: providers **tie corporate clients to their APIs** - first a low price, then dependence.

A vivid example is advertising on Meta. Businesses built their customer acquisition on the platform because that is where the customers are, and leaving is hard. Meta's average price per ad grew 12 % year on year in the first and second quarters of 2026 and about 14 % in the third.

In AI it may well be the same. While prices are low, companies rebuild their processes and cut people, getting used to an external API. And once a company has tied its critical processes to an external API without its own compute, the price trap snaps shut. Saying "we are leaving" will be hard: hiring people back is difficult, and part of the institutional memory will be lost. A price increase will then simply move the payroll savings into the pocket of a dominant provider. By the way, that is exactly why, as I said in the first article, you have to keep the best expertise even while optimizing. And vendors are already changing the rules: Cursor in 2025 and GitHub in 2026 moved from flat plans to metered billing.

Demon four: business continuity and regulation

If you depend on a cloud provider, demand a guaranteed SLA of 99.99 % on response latency. If models are embedded in a live operational process (credit decisions, anti-fraud) and the external service hangs, the business stops.

Hence regulation in plain words:

1. **Few bank security chiefs are comfortable sending the source code of critical systems and scoring to an external SaaS cloud** without a private deployment and a data processing agreement. The code may contain database structures, tokens and data under banking secrecy, and external AI agents do not watch for that.
2. **Financial regulators treat critical IT outsourcing seriously.** A financial institution has to know where its data and systems are, keep control over the provider and answer to the regulator for it. Many simply do not want to hand their code and data to a third party.
3. **Code leaking through personal subscriptions into a public service** is a direct route to regulatory findings and personal liability of management.

Handing critical processes to an external API without nodes of your own is not an option. So what do you do?

"Demon slayer": your own server nodes and the dual-contour model

If external clouds carry such risks, do you need to build a data center? No. The basic compute block is **one server node with 8 H200 accelerators and about 1.1 TB of GPU memory (HBM)**. That is enough to run quantized flagship open models locally (DeepSeek-V4, GLM 5.3, Kimi K3.1) and give developers an honest 256k context. For the heavy Western profile the company needs two such nodes (the calculation is below).

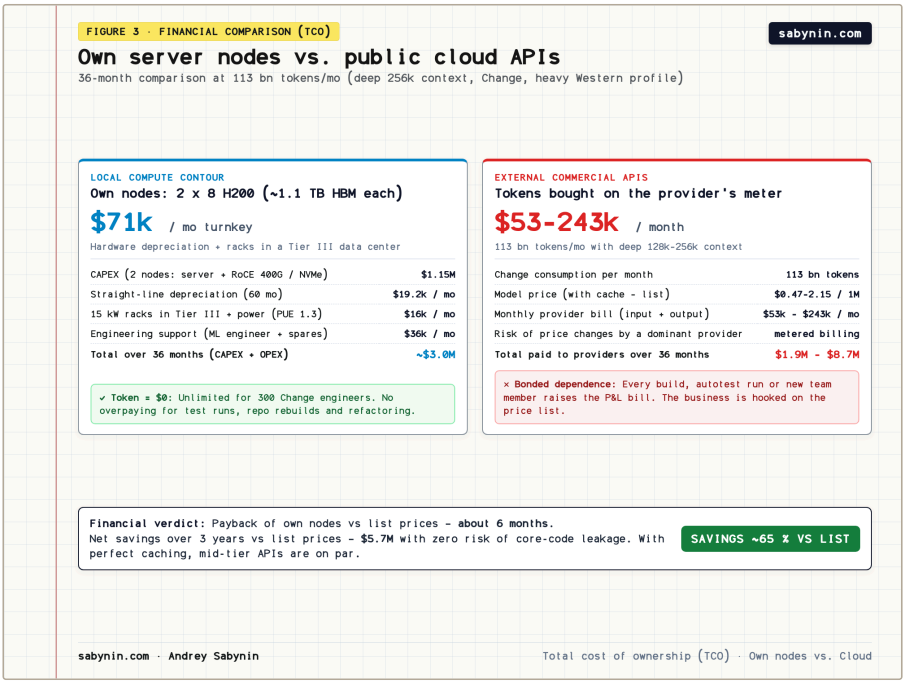


Figure 3. Own nodes vs public cloud APIs

Financial comparison of own nodes with the commercial cloud API (US prices, taken with a margin; the CAPEX is per node):

Metric (113 bn tokens/month, two nodes)	Public cloud API (metered, list price without cache)	Own nodes: base TCO (purchase, 5-year depreciation)	Own nodes: CFO stress test (3-year lease at 8 %)
One-off CAPEX	$0	**$575k per node** (8x H200 server $470k + 400G network $60k + 60 TB NVMe $45k)	**$0** (entry through rental, then lease)
Depreciation / lease payment	—	**$9.6k a month per node** (60 months)	**$18.0k a month per node** (36 months at 8 %)
Operations (OPEX, Tier III data center)	Included in the price	**$26k a month per node** (15 kW rack $6k, power $2k, 0.5 FTE engineer $13k, spares and support $4k, network $1k)	**$26k a month per node**
TOTAL monthly (TCO)	**$243k a month** (with cache: $53-118k)	**$71k a month** (fixed, unlimited)	**$88k a month** (fixed, unlimited)
Cost per 1M tokens	**$2.15** (with cache: $0.47-1.04)	**$0.63** (marginal cost of an extra token = $0)	**$0.78**
Total cost over 36 months	**$8.7M** (+ risk of price changes)	**$3.0M** ($1.15M CAPEX + $1.87M OPEX)	**$3.2M** (full lease and OPEX)
Net savings over 3 years (tokens only)	**Baseline ($0)**	**+$5.7M** (payback about 6 months)	**+$5.6M** (no CAPEX)

The main comparison with the cloud

For $71k a month your two nodes give you **unlimited compute and freedom to choose models**: on your own nodes the marginal cost of an extra token is zero. At list prices the same 113 bn tokens cost about $243k a month.

And if you switch on prompt caching?

The fair question from colleagues after the Russian publication: why compare with list prices if providers bill re-read context at a discount? In agentic development up to 90 % of input tokens are cache hits: at every step the agent re-reads the same repository. Recalculated with caching (Anthropic bills new input at the 5-minute cache-write price):

Monthly bill for Change	Modest profile (22.6 bn, x0.2)	Heavy Western profile (113 bn, x1)	x2 (226 bn)
OpenAI gpt-6.1 (mid-tier), with cache	$11k	**$53k**	$107k
Claude Sonnet 5.5, with cache	$12k	**$59k**	$118k
gpt-5.3-codex, with cache	$13k	**$67k**	$133k
Claude Opus 5.5, with cache	$24k	**$118k**	$236k
Own nodes, purchase	$36k (1 node)	**$71k** (2 nodes)	$107k (3 nodes)
Own nodes, lease	$44k	**$88k**	$132k

The honest conclusion: at the heavy profile a mid-tier API with perfect caching is 10-25 % cheaper than your own nodes, and at the modest profile it is cheaper still. But your nodes run open frontier models - Kimi K3.1, GLM 5.3 - and against frontier APIs (Opus-class), weak caching or list prices they win by a wide margin, on top of the price risk and security. The lease does not save money against a cached API - its job is to remove CAPEX and the cash gap at the start.

But caching has two "buts". It is not guaranteed: it depends on how the agent builds its prompts, and cache entries live for minutes. A study of GitHub Copilot traces measured about 90 % cache hits within a turn but only about 55 % across turns. And

the main thing: the core-system code, database schemas and banking secrets will not go to an external cloud anyway - see demon four.

What about subscriptions? Claude Team Premium and ChatGPT Business Premium cost $100 per seat a month on annual billing. For 300 developers that is $30k a month - cheaper than your own nodes. But they have usage limits, and heavy agentic use above them is billed at API rates on top (Claude Enterprise bills usage at API rates on top of the seat). Subscriptions are a cheap entry, not the price of heavy use.

So the logic of the dual contour stays; only the order is clarified: start with the API or with rented servers, measure the real token volume and cache hit rate for 2-3 months, and move development onto your own nodes when the volume approaches the threshold or security and the regulator require it. The threshold and the sensitivity to the cache share are on Sheet 4 of the spreadsheet.

Will the nodes really handle the load?

The question your CTO will ask: "Each of your nodes has about 1 TB of GPU memory. If dozens of strong engineers load tasks with 256k context in parallel - won't the servers fall over?"

My answer: **it will cope, with room to spare**. Fifty people never press "Enter" in the same millisecond. The load is spread over the day, and the model architectures (DeepSeek-V4, GLM 5.3, Kimi K3.1) sustain the flow thanks to latent compression of attention memory (MLA/DSA), deduplication of shared repository prefixes (RadixAttention) and offloading of idle sessions to fast NVMe drives.

How many nodes for 113 bn tokens a month if you buy them? Back-of-the-envelope: A working month is 21 days of 10 hours, 756 thousand seconds. 113 bn tokens over that time is about 150 thousand tokens per second. It sounds like a lot, but about 90 % of it is cache: the agent re-reads the same repository, and the node does not recompute it but takes it ready from memory. What really has to be computed is about 14.7 thousand tokens per second of new input and about 2.8 thousand tokens per second of output.

How much does a node give? DeepSeek published data from its own service: a node of 8 H800 accelerators processed on average about 74 thousand tokens per second of input, more than half of them from cache, and about 15 thousand tokens per second of output. DeepSeek runs input processing (prefill) and output (decode) on separate nodes; I fold both into one node and take three times less, also because we have a single node rather than a cluster and a quantized model: about 10 thousand tokens per second of new input and 5 thousand of output. So the heavy profile needs two nodes, loaded at about 75 % on input and 30 % on output. The rest is headroom for peaks and night batch jobs: test runs, reviews, documentation. At x2 you need a third node.

A provocative thesis: development without high availability, and client support in a cheap cloud

When architects bring an estimate of well over a million dollars for a redundant internal cloud (two sites in different data centers, geo-replication), ask a direct question: **what happens if the internal AI server for developers hangs for 45 minutes?**

Will payments stop? Will ATMs stop dispensing cash? Will the regulator revoke the licence? Nothing will happen. Programmers will sigh, go and pour a coffee, discuss the architecture at the whiteboard or write code by hand, as they have for thirty years. Heavy development (the Change contour, about 99 % of tokens) is an internal contour where business customers are used to waiting months for features; it will still be much faster than "yesterday".

A caveat that colleagues fairly made: "they will write by hand" works as long as the people are there. Once the headcount is already cut and production rests on agents, node downtime has to be counted through operational risk. The fix is cheap: open pay-as-you-go accounts with two cloud providers in advance and get the rate limits raised before an outage. While your nodes

are up, such a reserve costs almost nothing; at 5 days of downtime a year it costs about $2k a month at the heavy profile on a mid-tier model with cache (Sheet 4).

True 24/7 continuity is critical only in client services and support:

Architectural criterion	Contour 1: heavy Change	Contour 2: client Run and office
Share of traffic and context	**About 99 % of all tokens**, deep context 128k-256k	**About 1 % of all tokens**, short requests of 2k-10k
Criticality of 1 hour of downtime	**Zero for revenue**: programrs drink coffee or write by hand; client payments do not stop	**Critical (24/7)**: client bots, scoring and the call center stop
Security requirements	**Hard ban on external clouds**: core banking source code, database schemas and banking secrecy inside	**External cloud acceptable** after masking personal data at an internal DLP gateway
Where it runs and which models	**Own nodes on-prem** (8 accelerators, ~1.1 TB HBM each): open flagships `DeepSeek-V4`, `GLM 5.3`, `Kimi K3.1`	**Cheap external APIs** with the provider's geo-redundancy: `Claude Haiku 5.5`, `GPT Luna`, `DeepSeek-V4.1-Flash`
Fault tolerance	**No 2N cluster** (saving over a million dollars). Insurance: cold spares in the data center (replacement in 4 hours) + a 4-bit fallback on rented bare metal within an hour	**SLA 99.99 % out of the box** thanks to the infrastructure of a large cloud provider
Price	**$71k a month** for two nodes (unlimited compute)	**Under $1,000 a month** (hard quotas per department)

The architecture of the frugal CIO

The hybrid dual-contour model:

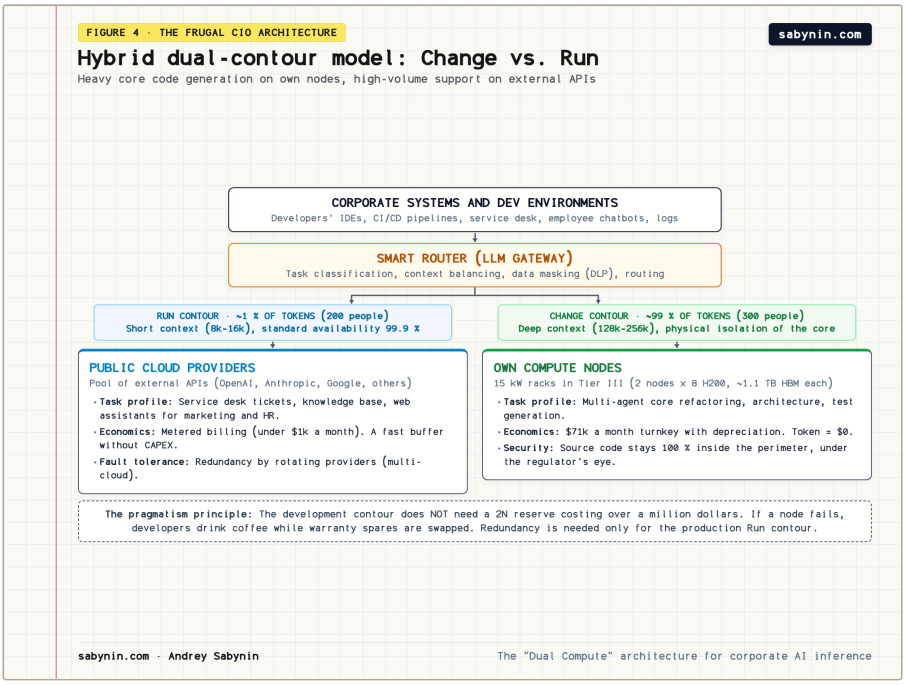


Figure 4. The architecture of the frugal CIO: the dual-contour model

1. **A single internal AI gateway**: all requests go through one corporate entry point (access checks, DLP masking of data, audit and smart routing).
2. **Differentiated quotas**: the office and client services get hard limits on light external models with SLA 99.99 %, cutting off spam.

3. **An isolated core contour**: development of critical systems lives on your own servers under security control, without the cost of a redundant second cluster.
4. **Rent first, lease later**: day 1 - rent dedicated servers (about $18k a month per 8-GPU node at roughly $3 per GPU-hour, my estimate); when scaling up, move the nodes into a 3-year lease.
5. **Access in exchange for savings**: access to agents is opened to teams strictly in exchange for a commitment to cut the cost of a feature and reduce headcount in waves.

Conclusion

AI transformation is definitely not an IT director's technology project. It is a hard financial reorganization of the company, and it can be calmly counted in money.

And if you simply pull AI agents over the old bloated processes and the previous headcount, you are guaranteed to get a worse P&L and, quite possibly, a lock-in to providers.

The winners will be the executives who approach the process meticulously from day one - with a pencil, a calculator and the CFO: they will work out the schedule of rolling waves, agree with the key core of engineers and build their own compute contour.

A reality check: Revolut

While I was translating this article (the Russian version came out on 1 November), news came in that checks my model against reality. On 3 November The Register reported on a talk by David Abitbol, ops and infrastructure lead at Revolut, at AI Engineer NYC (13 October 2026). These are the company's own figures, not an independent audit, but the match is telling:

My model	Revolut
Heavy development on your own nodes with open flagships, including GLM 5.3	GLM 5.3 self-hosted for about 2,000 engineers, after GLM 5.1
One 8-accelerator node per about 150-300 developers	56 B200 GPUs, that is seven 8-GPU nodes, for about 2,000 engineers - about 285 per node
An honest 256k context is enough	"Long context is a scam" - 256k is what is really needed, despite 1M support
A single gateway with smart routing	A model router: vision queries go to Gemini or GPT Luna, code stays on GLM
Dual contour: frontier models for the rest	"The frontier is the exception" - GPT and Claude are not replaced entirely
Uncontrolled spend is the first trap	Engineers spent "$1,000 a month, $10,000 a month" each with "not so much ROI", so the launch came with strict budget limits
Own nodes are several times cheaper than the API at heavy use	Inference cost for AI coding down about 90 % in under a year; about $0.30 per 1M tokens on GLM 5.1

Revolut also reports about 60 % faster coding time and about 70 % shorter lead time, without separating the effect of AI from other factors. My cost per 1M tokens is about twice theirs ($0.63 against $0.30), on older H200 silicon: they spread seven nodes across 2,000 engineers and tuned the serving stack (quantization, disaggregated prefill and decode). The direction is exactly the one I described.

The detailed token calculation by unit, the context methodology, the node specification with US prices, the payroll model with rolling waves, and the cache, range and break-even analysis are in the file **ai-infrastructure-spec-2026.en.xlsx**.

Sources

- API prices: OpenAI API pricing, Claude API pricing, Claude plans
- Heavy usage on flat plans: Anthropic weekly limits, quoted by Simon Willison, Cursor pricing change, TechCrunch, GitHub Copilot usage-based billing
- Measured usage: Jellyfish, AI engineering trends, Larridin via SaaStr, GitHub Copilot trace study, arXiv 2608.00101
- Subscription economics: SemiAnalysis
- Meta token leaderboard: Fortune, The Decoder
- McKinsey, AI budget overruns: Burning through the AI budget
- Hardware and data-center costs: GPU server prices, gpuperhour, enterprise SSD prices, StorageReview, CBRE data centers, colocation pricing, Brightlio
- Salaries: Glassdoor, software engineers in Germany, Playroll, employer costs in Germany
- Node throughput: DeepSeek-V3/R1 Inference System Overview
- Revolut: The Register, 3 November 2026, talk at AI Engineer NYC



Andrey Sabynin
Former bank management board member, CIO, founder of an AI-native IT company. Hands-on AI transformation and IT cost reduction for banks and fintech.