AI ARCHITECTURE & ENTERPRISE STRATEGY REPORT

The Fly Effect: Event-Driven Agent Architectures and the Cost Misconception

Lowering Enterprise LLM Costs, Starting from the Principle of Inertia in a Moving Car

Mehmet Keskinkılıç · November 2026 **Category:** System Design & Financial Optimization

Approach: Deterministic State Machines + Targeted Inference

Abstract: *Many businesses postpone AI transformation and autonomous agent integration because of "unpredictable API costs". An important cause of this hesitation is not a lack of technology but an architecture that triggers large models at every step of a workflow. Inspired by the physics of a fly inside a closed car, this paper sets out a roadmap for an "Acceleration Agent Model": routine operations stay at zero token cost and the LLM runs only when an anomaly occurs. The figures in this report are example-scenario assumptions, not measurements.*

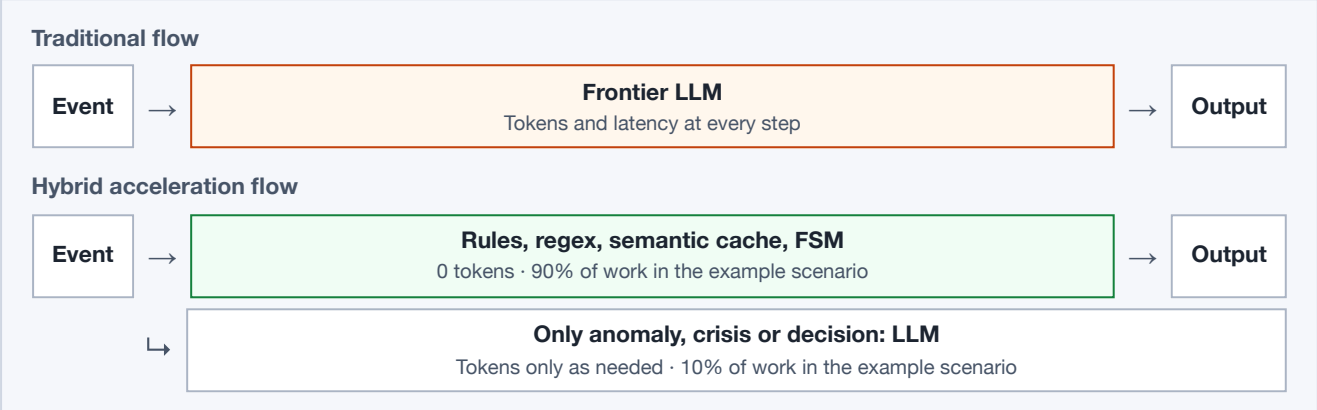
1. The Physical Metaphor: Why Isn't the Fly in a Closed Cabin Thrown Around?

A fly inside a closed car travelling at 100 km/h makes no extra effort to hover or to fly from the windscreen to the back seat. There are two basic physical reasons: *the air mass carried along at the same speed* and the principle of *inertia*. The fly moves relative to the still air in the cabin, not relative to the road. It spends a lot of energy keeping its balance only when the car brakes hard, accelerates, or a window opens and lets in turbulence.

Traditional agent designs behave as if the fly had to fly outside the car against a 100 km/h wind: every routine data entry, status check and simple approval step triggers a large language model (LLM) from start to finish, and token consumption runs out of control.

2. The Enterprise Mistake: The Reflex to Attach an LLM to Every Step

Behind the claim that "AI transformation will exceed our budget" there is often a monolithic agent design that passes every operational event straight to an expensive frontier model. Yet a significant share of enterprise processes (customer requests, invoice checks, ERP updates, ticket routing) can run on predictable, deterministic rules. That share varies by process and has to be measured; this report's example scenario assumes that 90% of the work finishes without reaching a model.



© 2026 Mehmet Keskinkılıç · mehmetkeskinkilic.com

3. The Three Pillars of the Architecture

A. Deterministic Gating Layer

Every input first passes through zero-cost or micro-cost checks. If it matches a standard pattern, a predefined API route or a rule-based validation, the operation completes directly in code. No LLM API call is made and no tokens are spent.

B. Workflow Based on a Finite State Machine (FSM)

The agent's processes are modelled with state-oriented frameworks such as LangGraph. Moving from one state to another, decision rules run in code. Only an unexpected exception, an unstructured data anomaly or a strategic fork with several options makes the system raise an "acceleration" alert and branch to the LLM node.

C. Semantic Caching and Diff-Based Inference

Repeated queries are caught with lightweight local embedding models and served straight from the cache. Even when an LLM call is required, the company's full document set is not resent each time; prompt caching and sending only the changed state parameters (the delta) reduce spending.

4. Cost and Performance Comparison (Example Scenario)

The table below is not a measurement of a real deployment. It is an example scenario for a mid-sized support or processing centre that handles 10,000 operational events a day.

Metric / Stage	Monolithic LLM Agent	Hybrid Acceleration Agent	Savings & Effect
Routine Flow (75%)	~1,500 tokens per call	0 tokens (runs at rule and API level)	100% LLM token savings
Similar / Repeated Input (15%)	Model reasons from scratch	0 tokens (semantic cache match)	100% LLM token savings
Anomaly & Decision (10%)	Runs with expensive context	Small or optimized model + minimal diff context	70% token optimization
Example Response Time	1,800 ms – 3,500 ms	< 40 ms (routine) / 900 ms (anomaly)	Over 97% lower on routine work, about 50–75% lower on anomalies
Estimated Monthly Cost	$4,500 – $7,000 / month	$350 – $600 / month	**About 87% – 95% estimated savings**

Calculation assumption: About 300,000 events a month × ~1,500 tokens = ~450 million tokens, at an average of roughly $10–16 per million tokens. On this basis the token cost of the 10% anomaly share in the hybrid model is about $135–215; the range in the table sits above that as a cautious estimate. Rule writing, infrastructure, maintenance and cache operating costs are excluded from both columns.

© 2026 Mehmet Keskinkılıç · mehmetkeskinkilic.com

5. Board Summary and Action Plan

> **Strategic Conclusion:** AI transformation does not mean "handing every process to an LLM". Successful transformation separates the deterministic reliability of software engineering from the problem-solving ability of LLMs at the right thresholds. When, as in the example scenario, 90% of the work runs without spending tokens and AI steps in only for the remaining 10% of uncertain moments, the cost barrier can become an advantage.

Action plan

1. **Measure:** Record current LLM calls by event type, with volume, tokens, latency and error rate. Replace this report's ratios with your own data.
2. **Separate:** Move high-volume routine work with verifiable results to rules and state machines.
3. **Open the cache under control:** Run the semantic cache with a similarity threshold, audit samples, and keep personal or sensitive data from crossing between answers.
4. **Set an escalation rule:** Send anomalies to a small model first. If confidence is low, escalate to a large model or a person.
5. **Monitor continuously:** Track the share of work completed without a model, wrong rule and cache answers, and total monthly cost including engineering.

Mehmet Keskinkılıç
Aviation Operations Specialist & Indie iOS Developer
© 2026 Mehmet Keskinkılıç. May be shared with attribution. Latest version: mehmetkeskinkilic.com/writing/