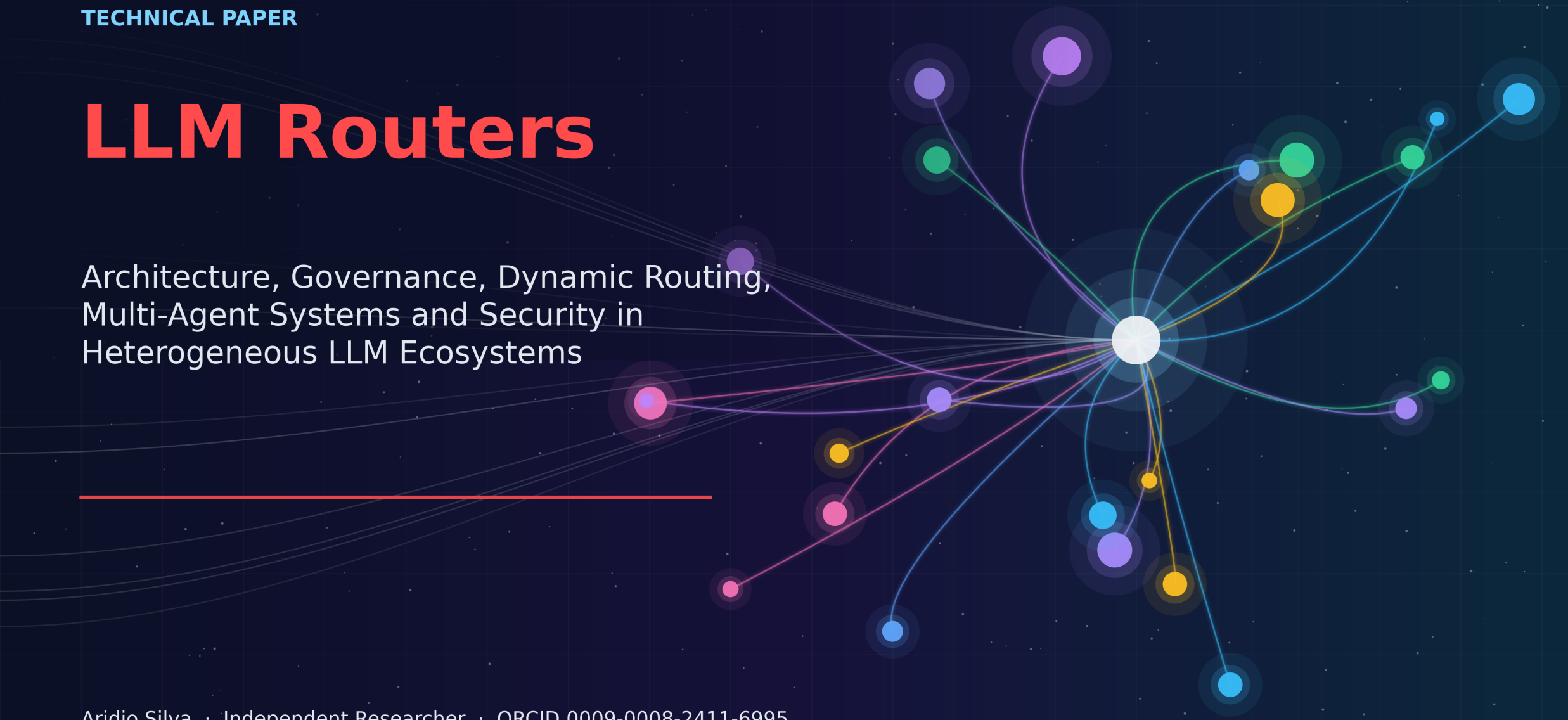
TECHNICAL PAPER

LLM Routers

Architecture, Governance, Dynamic Routing, Multi-Agent Systems and Security in Heterogeneous LLM Ecosystems



Aridio Silva · Independent Researcher · ORCID 0009-0008-2411-6995

© Aridio Silva @aridiosilva Outubro 2026

Abstract

- LLM routers decide, for each request, which model answers, and add gateway functions: authentication, budgets, fallback, caching and auditing.
- We cover formalization, strategies, API integration, multi-agent systems, GRC, security, benchmarks and maturity.
- Technical review with source auditing: arXiv, ACL, NIST, EUR-Lex and official documentation.

2×+

cost savings with quality kept (RouteLLM)

13.4%

of calls to GPT-4 recover 50% of performance (RouteLLM)

−24.7%

commercial router below the best single model (LLMRouterBench)

© Aridio Silva @aridiosilva Outubro 2026

Contents

1 Introduction

2 Theoretical foundations

3 Methodology

4 Routing strategies

5 Architecture and OpenAI API

6 Products

7 Models

8 Authentication

9 Agents and swarms

10 Governance, Risk and Compliance

11 Threats and vulnerabilities

12 Evidence and benchmarks

13 Historical evolution

14 Maturity and deployment

15 Discussion and recommendations

16 Conclusion

© Aridio Silva @aridiosilva Outubro 2026

A heterogeneous ecosystem calls for a router

Context

Proprietary, open and reasoning models differ in cost, latency, context and capability.

Problem

Always using the strongest model is expensive; always using the weakest fails on hard queries.

Question

How to decide, for each request, which model to use while controlling cost, risk and compliance?

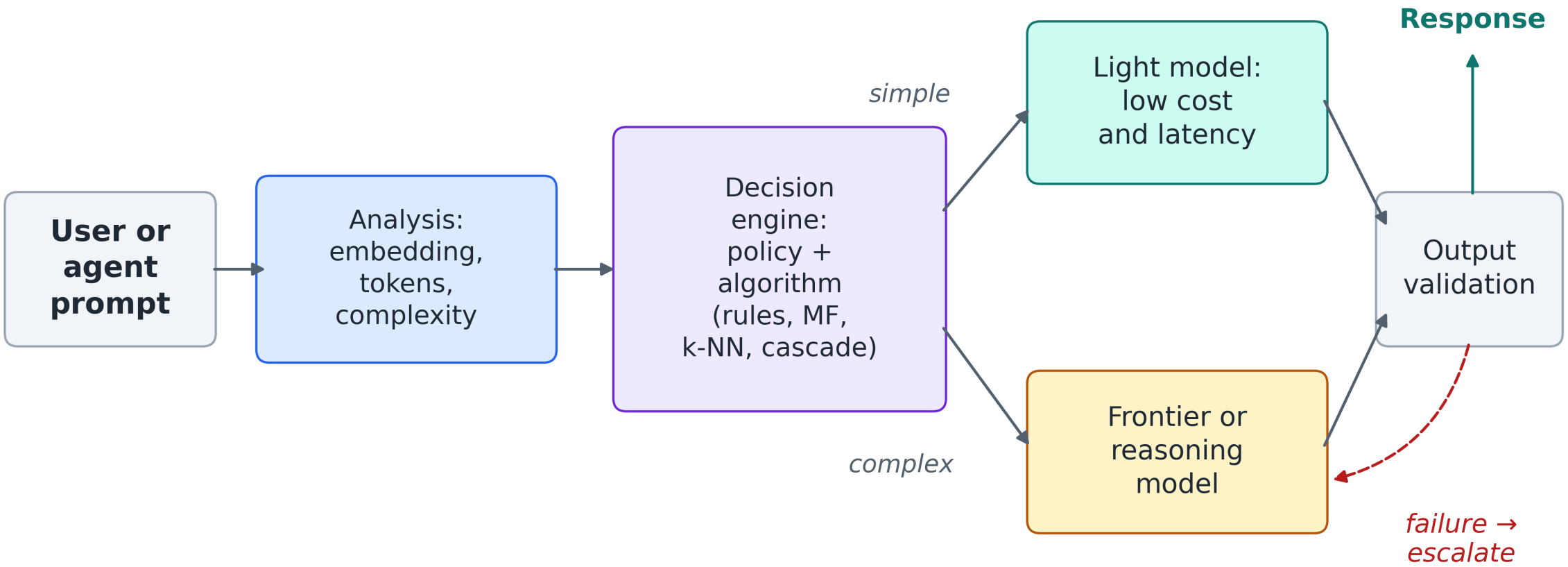
US$ 1.25 × US$ 75

per million output tokens: Claude 3 Haiku × Claude 3 Opus, 2024 prices

© Aridio Silva @aridiosilva Outubro 2026

Figure 1. Conceptual decision flow of an LLM router

From request to response: prompt analysis, policy, routing algorithm, dispatch and validation



Goal: the smallest model able to solve each request with quality.

Source: prepared by the author from Ong et al. (2025), Chen et al. (2023), Aggarwal et al. (2024)

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

The trilemma: cost, latency and quality

Cost

Reasoning and large models cost far more per token.

Latency

Larger models take longer to the first token (TTFT).

Quality

Smaller models fail on reasoning, code and nuanced analysis.

Resilience

Providers go down and impose limits (RPM, TPM); the router fails over.

Principle

Allocate each request to the smallest model able to solve it with satisfactory quality.

© Aridio Silva @aridiosilva Outubro 2026

Measuring a router: PGR, APGR and CPT

Threshold routing α

Weak model if P(strong wins | query) < α; otherwise, strong model.

PGR

(r_router − r_weak) / (r_strong − r_weak): fraction of the performance gap recovered.

APGR

Mean PGR as α varies: area under the PGR × cost curve.

CPT(x%)

Smallest fraction of strong-model calls needed to reach a PGR of x%.

A router that always calls the strong model has PGR = 1 and no savings; hence APGR and CPT.

© Aridio Silva @aridiosilva Outubro 2026

Five ideas that justify routers

1. **No model is best at everything.** Cost, latency and specialty vary.
2. **Queries vary in difficulty.** Many are simple.
3. **Cost is asymmetric.** From 60× to about 100× between models.
4. **The decision can be predicted before generating.** One call per query.
5. **Decouple app and model.** Creates a single control point.

© Aridio Silva @aridiosilva Outubro 2026

Seven objectives and how to measure them

Objective	How the router pursues it	KPI
Cost efficiency	Simple queries to cheap models	Share of strong-model calls
Quality	Strong model only where required	PGR, APGR, CPT
Low latency	Smaller models, less reasoning	TTFT, total latency
Resilience	Fallback, circuit breaker	Error rate, availability
Portability	Single OpenAI-format API	Time to switch model
Governance	Virtual keys, budgets, auditing	Control coverage
Observability	Central log of cost and model	Cost per team, agent and app

© Aridio Silva @aridiosilva Outubro 2026

Design principles and costs to consider

Principles

- Smallest sufficient model
- One call per query
- Tunable trade-off (threshold α)
- Compare with a simple baseline
- Fail safely (fallback)

Costs and limits

- Overhead and extra calls in cascades
- Uncertain gain from the algorithm
- New point of failure and attack
- Continuous gateway operation

© Aridio Silva @aridiosilva Outubro 2026

Technical review with source auditing

1

Initial survey

Exploratory research that scoped the topics.

2

Primary sources

Papers, repositories, documentation, NVD and standards.

3

Complementing

New papers, OWASP, regulatory milestones and CVEs.

4

Correction and marking

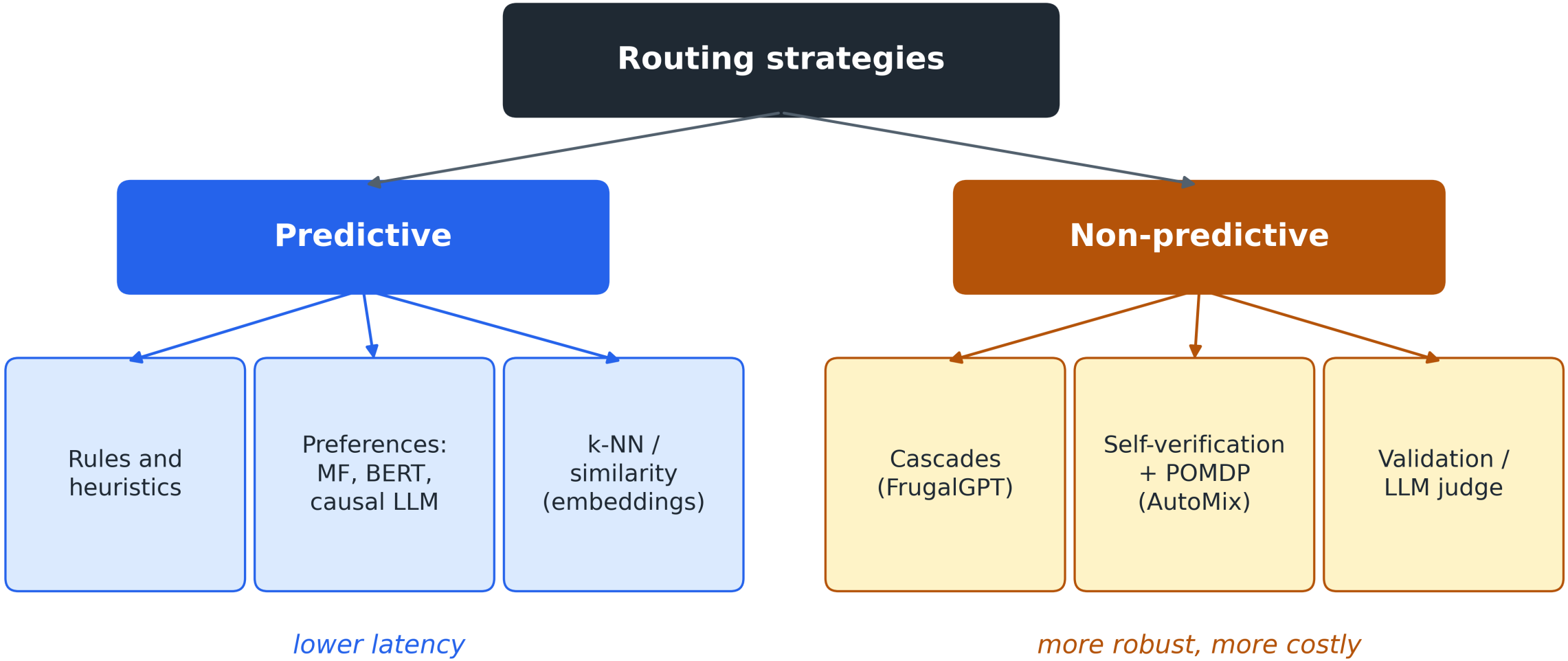
Errors corrected; what lacks a source is flagged.

Audited source: title, authorship, date, venue and data checked on the page itself; link accessible between 28 and 29 Oct 2026.

© Aridio Silva @aridiosilva Outubro 2026

Figure 2. Taxonomy of LLM routing strategies

Predictive routing decides before generating (1 call per query); non-predictive generates, evaluates and escalates (≥1 call)



Source: Hu et al. (2024); Ong et al. (2025); Ding et al. (2024); Chen et al. (2023)

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

4 STRATEGIES

Six strategies, six trade-offs

Strategy	Calls	Strength	Limitation
Rules	1	Transparent and auditable	Fragile to paraphrases
Preference (SW, MF, BERT, causal LLM)	1	Over 2× savings	Recalibrate the threshold
Semantic k-NN	1	Simple and incremental	Depends on labeled base
Cascade	1 to N	Up to 98% savings reported	Accumulated latency
Self-verification + POMDP	1 to N	Works with black-box APIs	Noisy verification
LLM-as-a-Judge	2 to N	Flexible criterion	Judge cost and bias

© Aridio Silva @aridiosilva Outubro 2026

4 STRATEGIES

RouteLLM: human preferences become a router

SW ranking

Similarity-weighted Bradley-Terry; no training.

Matrix Factorization

Bilinear score between model and query embeddings.

BERT

Fine-tuned BERT-base with a logistic head.

Causal LLM

Llama 3 8B predicts the win probability.

~65,000

Chatbot Arena comparisons (64 models)

~120,000

GPT-4 judge labels

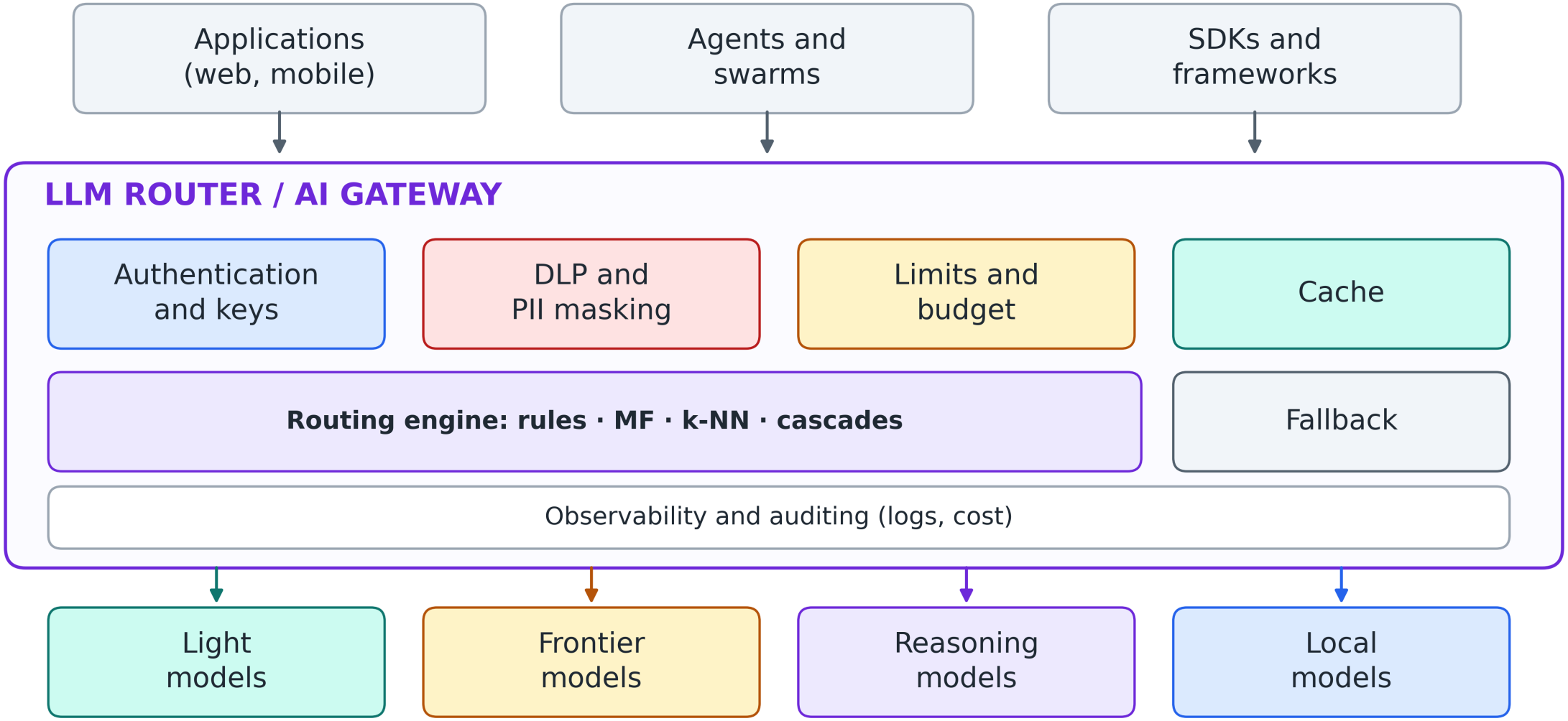
US$ 700

approximate cost of the labels

© Aridio Silva @aridiosilva Outubro 2026

Figure 3. Reference architecture of an LLM Router / AI Gateway

Control plane between clients and providers, exposed through an OpenAI-compatible API (POST /v1/chat/completions)



© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

The OpenAI standard: the request

```
curl http://localhost:4000/v1/chat/completions \
  -H "Authorization: Bearer $KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "auto-router-cost-saving",
    "messages": [
      {"role": "user",
       "content": "Capital of France?"}
    ],
    "stream": false
  }'
```

- POST /v1/chat/completions
- Authorization: Bearer <virtual key>
- model: model, alias or policy
- messages: list with role and content
- stream, tools, response_format…

Format launched with the ChatGPT API on 1 Mar 2023.

© Aridio Silva @aridiosilva Outubro 2026

The response and streaming

```
{
  "id": "chatcmpl-example-123",
  "object": "chat.completion",
  "model": "openai/gpt-4o-mini",
  "choices": [{
    "index": 0,
    "message": {"role": "assistant",
      "content": "Paris."},
    "finish_reason": "stop"
  }],
  "usage": {"prompt_tokens": 25,
    "completion_tokens": 8,
    "total_tokens": 33}
}
```

- stream: true → SSE events "data: {...}"
- chat.completion.chunk objects, same id
- The stream ends with data: [DONE]
- include_usage: usage in the last chunk
- The model field is the model actually used

Illustrative example; invented values.

© Aridio Silva @aridiosilva Outubro 2026

Open-source and managed: the ecosystem

Solution	Delivery	Highlight
LiteLLM	Open-source, self-hosted	100+ providers, virtual keys, spend control
RouteLLM	Open-source (Apache-2.0)	Trained routers, threshold calibration
LLMRouter	Open-source (research)	16+ routers in one interface
OpenRouter	SaaS	One key; fallback via the models field
Portkey	SaaS, open core	Fallback, cache, conditional routing, guardrails
Helicone	Open-source, observability	Cache, rate limits, cost alerts

© Aridio Silva @aridiosilva Outubro 2026

Models: what can be stated with a source

60×

Claude 3 Haiku × Opus: US$ 1.25 × US$ 75 per million tokens (2024)

~100×

Mixtral-8x7B × GPT-4: US$ 0.24 × US$ 24.7 per million tokens

2–5×

cost variation at comparable performance (RouterBench)

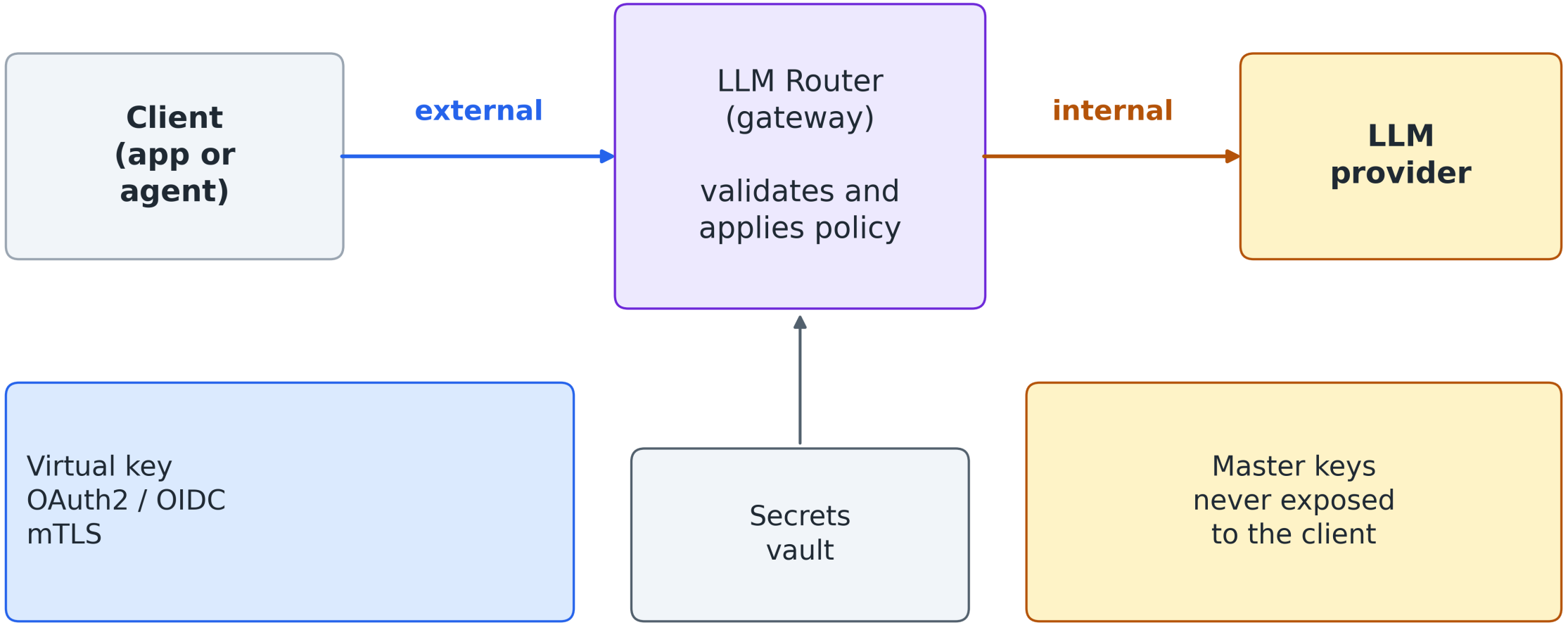
Previous model	GPT-5 model
GPT-4o	gpt-5-main
GPT-4o-mini	gpt-5-main-mini
OpenAI o3	gpt-5-thinking
OpenAI o4-mini	gpt-5-thinking-mini
GPT-4.1-nano	gpt-5-thinking-nano
OpenAI o3 Pro	gpt-5-thinking-pro

Prices and catalogs change: check the official pages.

© Aridio Silva @aridiosilva Outubro 2026

Figure 5. Two-layer authentication

The client only knows virtual credentials; the providers' master keys stay in the gateway vault



Source: LiteLLM and OpenRouter documentation (2026)

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

8 AUTHENTICATION

Virtual keys: five best practices

1. **Least privilege** allowed models per key
2. **Short lifetime** and automatic rotation
3. **Budgets and limits** per key, team and global
4. **Block api_base** parameters that change the destination
5. **Separate the planes** data × control, by network and role

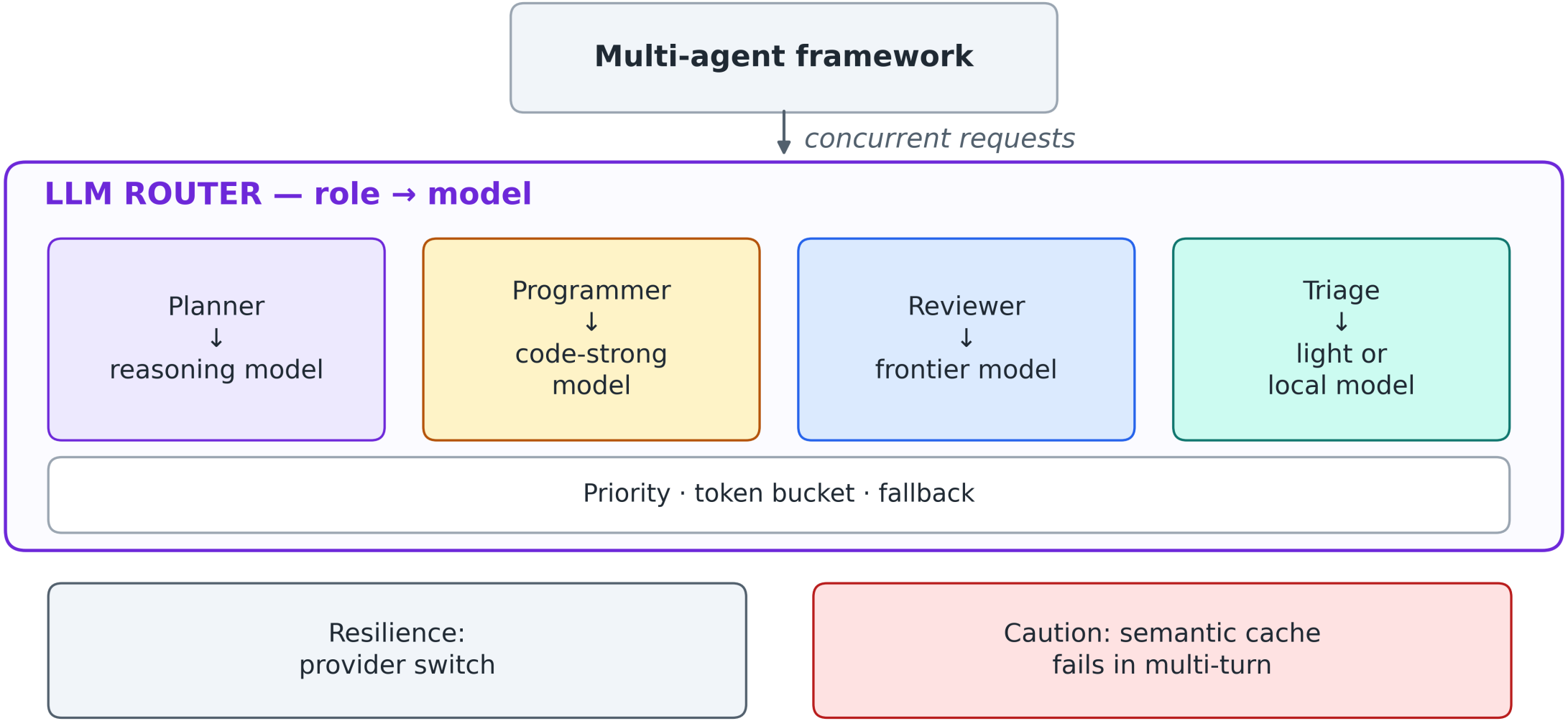
CVE-2024-6587

The api_base parameter redirected the call and leaked the OpenAI key.

© Aridio Silva @aridiosilva Outubro 2026

Figure 6. Role-based routing in multi-agent systems and swarms

The gateway maps each role to a model and protects the swarm from traffic spikes and provider failures



© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

Swarms: traffic and cost containment

Priority queues

The orchestrator goes ahead of background tasks.

Token bucket

Controls RPM and TPM with backpressure, without breaking agent state.

Cache with caution

Semantic cache is designed for single-turn use and can err in agentic flows.

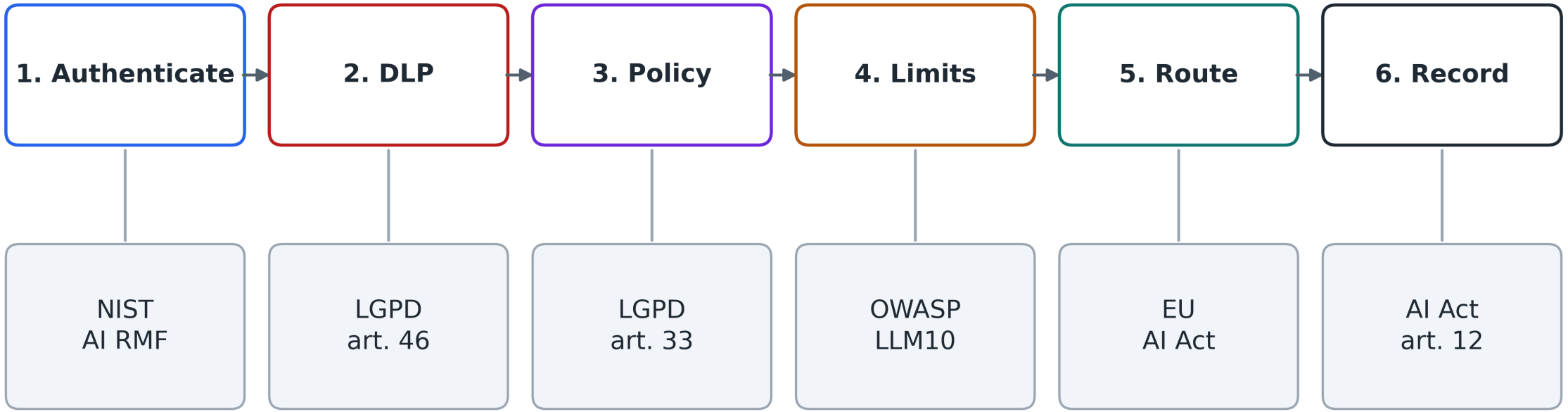
Fallback and budget

Provider switch on failure; budget per agent or run.

© Aridio Silva @aridiosilva Outubro 2026

Figure 7. GRC pipeline in the gateway and regulatory mapping

Each request step implements a verifiable control and produces audit evidence



Each control produces evidence for auditing.

Source: NIST AI 100-1; NIST AI 600-1; Reg. (EU) 2024/1689; LGPD; OWASP

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

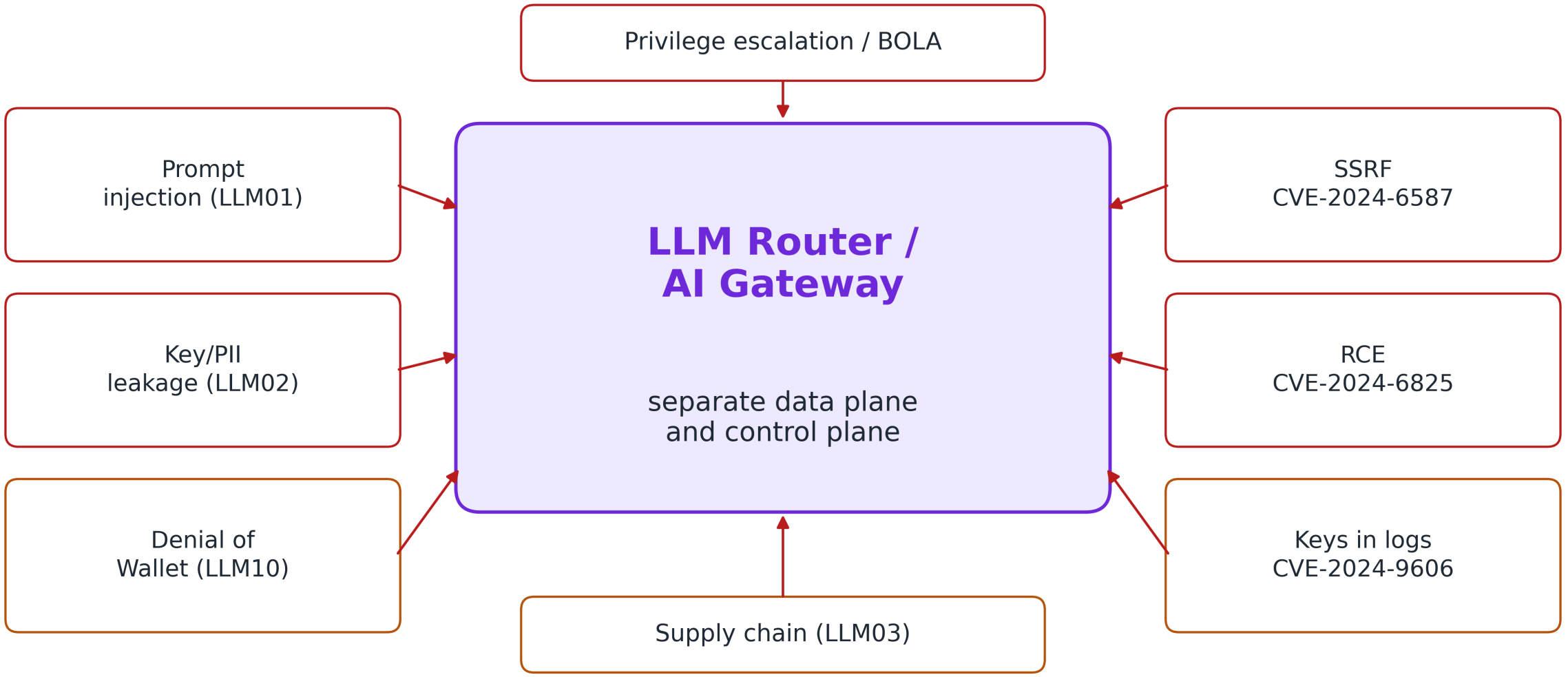
Reference frameworks: from NIST to LGPD

Framework	Date	What it brings to the gateway
NIST AI RMF 1.0	26 Jan 2023	Govern, Map, Measure, Manage
NIST AI 600-1	Jul 2024	Generative AI profile
ISO/IEC 42001	Dec 2023	AI management system
EU AI Act	in force 1 Aug 2024	Event logging (art. 12); high-risk postponed to 2 Dec 2027 (Annex III) and 2 Aug 2028 (Annex I)
LGPD	Law 13,709/2018	Security (art. 46), international transfer (33), records (37), incidents (48)

© Aridio Silva @aridiosilva Outubro 2026

Figure 8. Threat model and attack surfaces of an LLM gateway

Threats organized by trust boundary, referencing the OWASP Top 10 for LLM (2025) and real CVEs



Source: OWASP (2024); NVD; GitLab Advisory Database

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

Real vulnerabilities documented in LiteLLM

Lesson: continuous updates and defense in depth.

Class	Case	Version
SSRF and key leak	CVE-2024-6587 (NVD, 13 Sep 2024)	1.38.10
RCE	Advisory: eval in get_secret()	1.35.8
RCE	Advisory: configuration handling	1.40.12
Leak in logs	Part of the key exposed in a log file	2024
Integration secrets	Langfuse keys leaked	2024–2025
Privilege escalation	internal_user_viewer role	main-latest
SQL injection	api_key parameter in a SQL query	2024

© Aridio Silva @aridiosilva Outubro 2026

The routing algorithm itself can be attacked

~100%

success of confounder gadgets on open routers (worst case: 88%)

6–31 min

to generate a gadget (6 min on causal LLM, 31 min on SW)

>99%

detection accuracy of RerouteGuard (2026)

Cost escalation

Forces the expensive model.

Quality hijacking

Forces the weak model on complex queries.

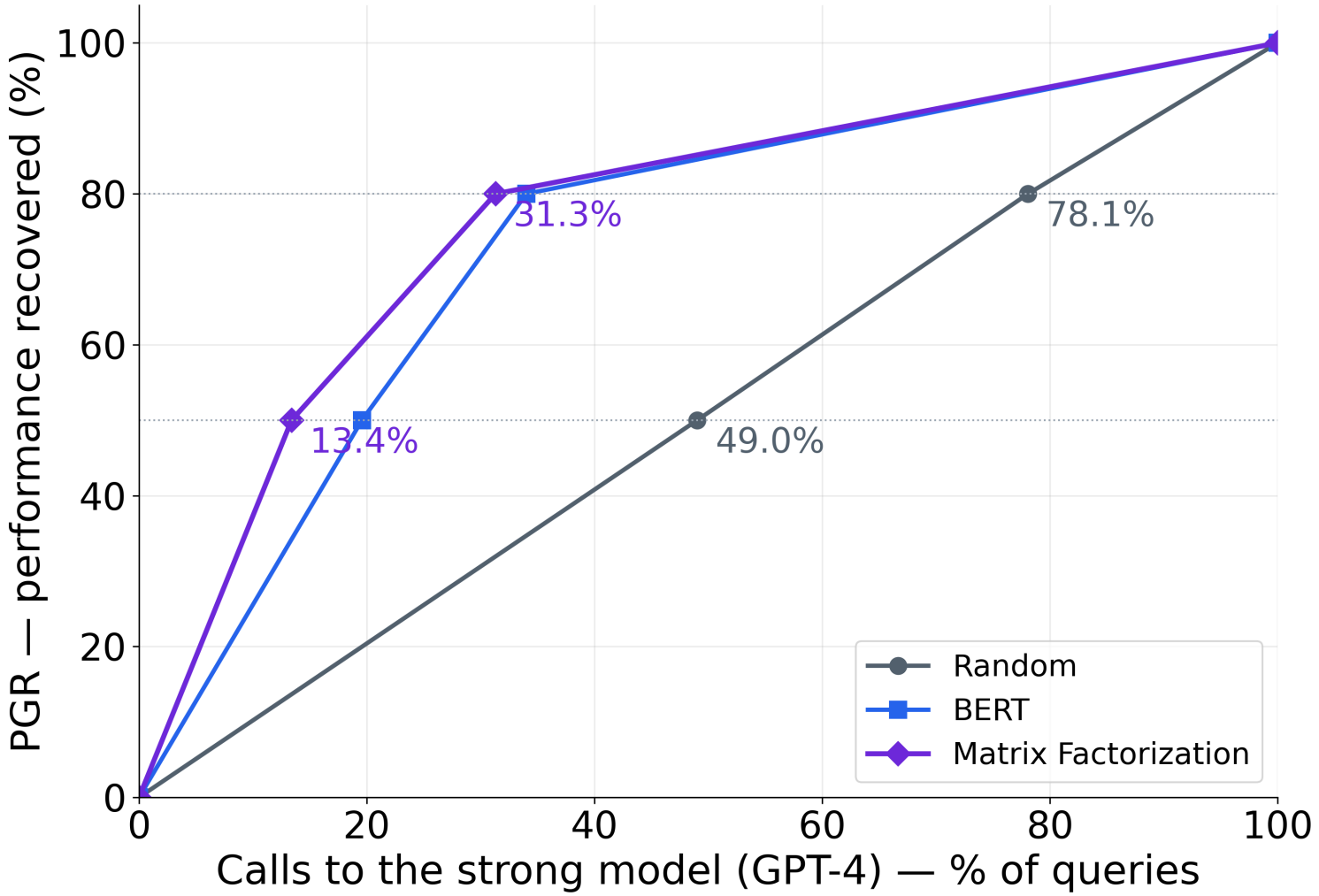
Safety bypass

Diverts harmful queries to a weaker model.

© Aridio Silva @aridiosilva Outubro 2026

Figure 4. Cost to recover performance on MT Bench (RouteLLM)

CPT(50%) and CPT(80%) points reported by Ong et al. (2025, Table 1); segments between points are illustrative interpolation



Reading

To recover 50% of the gap between Mixtral and GPT-4, the MF router sends 13.4% of the queries to GPT-4; the random one, 49.0%.

For 80%: 31.3% versus 78.1%.

Source: Ong et al., RouteLLM, ICLR 2025, Table 1.

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

RouteLLM on MT Bench: cost to recover performance

Training data	Router	CPT(50%)	CPT(80%)	APGR
—	Random	49.03%	78.08%	0.500
Arena	Matrix Factorization	25.32%	74.26%	0.580
Arena	SW Ranking	37.85%	58.99%	0.610
Arena + GPT-4 judge	BERT	19.58%	34.02%	0.751
Arena + GPT-4 judge	Causal LLM	31.50%	48.75%	0.679
Arena + GPT-4 judge	Matrix Factorization	13.40%	31.31%	0.802

More data matters more than architecture: the best result is MF with augmented data.

© Aridio Silva @aridiosilva Outubro 2026

Three benchmarks, three points of view

Criterion	RouterBench	RouteLLM	LLMRouterBench
Scale	405k samples, 11 models, 8 datasets	GPT-4 / Mixtral-8x7B pair	33 models, 21 datasets, ~1.8B tokens
Trade-off KPI	AIQ (NDCH)	APGR, CPT	PerfGain, CostSave, ParetoDist
Baselines	Zero router, Oracle	Random	Random, Best Single, Oracle
Latency	Not measured	Not central	Approximate analysis
Model generation	2023	2023–2024	2025

© Aridio Silva @aridiosilva Outubro 2026

LLMRouterBench: the critical counterpoint

33 · 21

models · datasets

+4%

maximum gain over the Best Single

−24.7%

OpenRouter router vs. Best Single (GPT-5)

31.7%

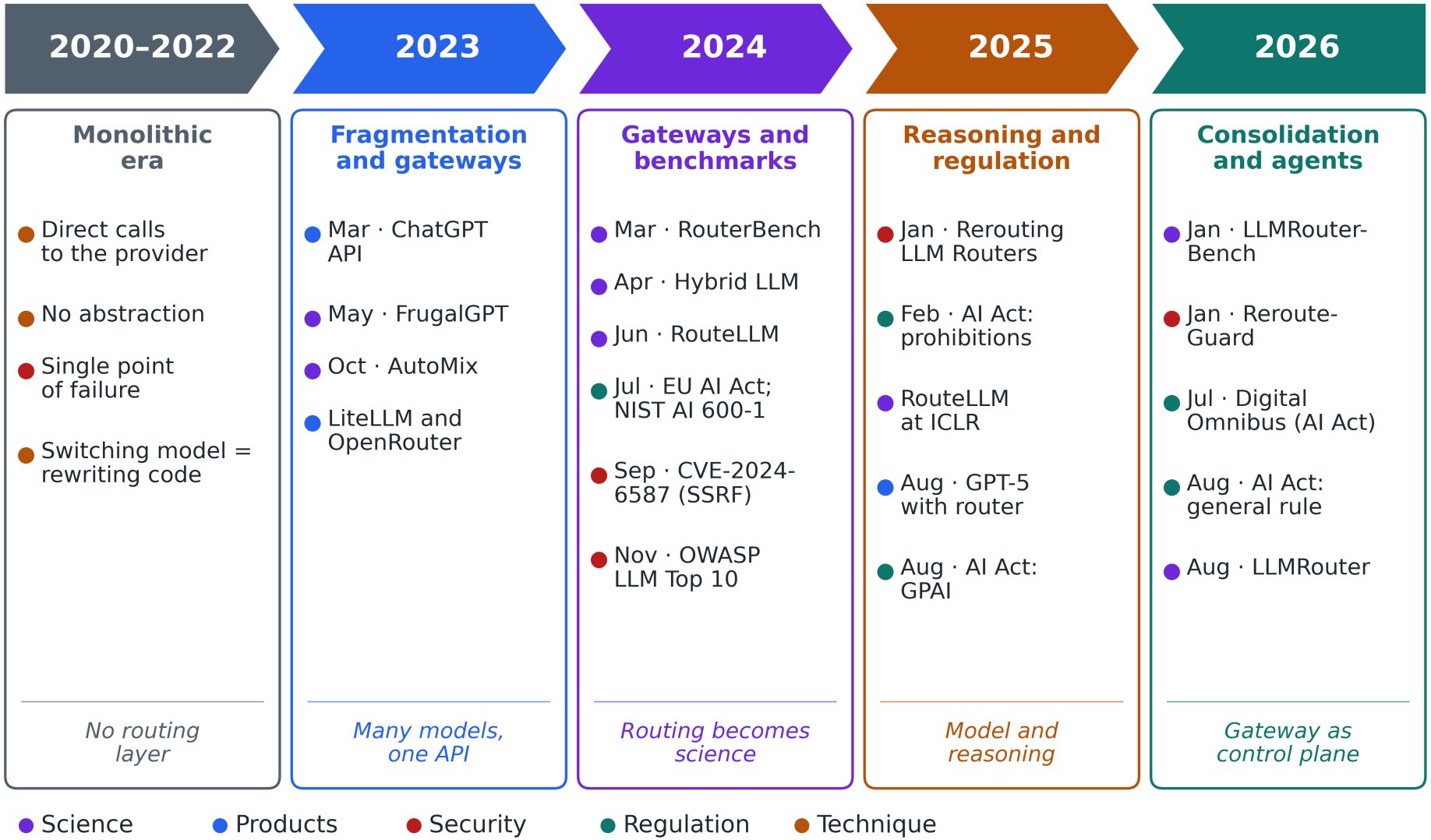
largest saving without losing accuracy

- Most methods converge to similar performance.
- Several do not reliably beat a simple baseline.
- Large gap to the Oracle, due to model-recall failures.
- The embedding model has little impact.

© Aridio Silva @aridiosilva Outubro 2026

Figure 9. Evolution of LLM routers by phase (2020–2026)

Scientific, product, security and regulatory milestones of each phase



Source: arXiv, ICLR, NeurIPS, ACL, NVD, OWASP, NIST, EUR-Lex

© Aridio Silva @aridiosilva — Outubro 2026

5 phases

2020–22: monolithic era

2023: gateways

2024: benchmarks and CVEs

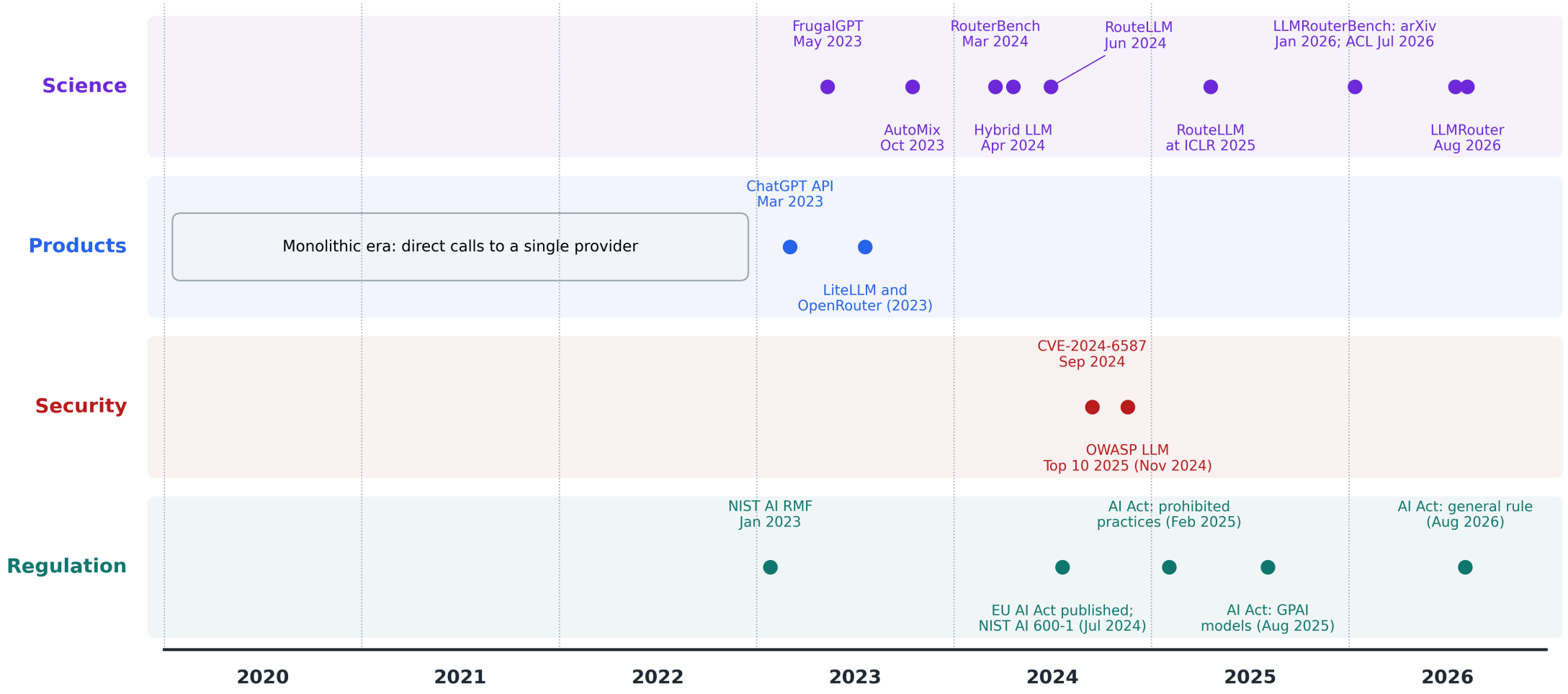
2025: reasoning and regulation

2026: consolidation and agents

© Aridio Silva @aridiosilva Outubro 2026

Figure 10. Timeline by topic lane (2020–2026)

Scientific, product, security and regulatory milestones on a linear time scale



© Aridio Silva @aridiosilva — Outubro 2026

Figure 11. Maturity of LLM routers (Oct 2026)

Author's assessment per dimension, based on the evidence in Section 14 (scale from 1 to 5)

	1 Research	2 Prototype	3 Early adoption	4 Wide production	5 Commo-dity	Evidence
Gateway / unified API				4		LiteLLM, OpenRouter
Fallback and limits				4		Native in gateways
Keys, budget, logs				4		Native; CVEs require updates
Rule-based routing				4		Hard-to-beat baseline
Learned binary			3			RouteLLM; does not beat the Best Single
Learned multi-model		2				Far from the Oracle
In mass-market product			3			GPT-5 (Aug 2025)
Agentic / multi-turn		2				Active research
Latency-aware	1					Not optimized
Adversarial robustness	1					Rerouting up to ~100%
Standardized evaluation			3			Benchmarks still fragmented

Two maturities

Control plane: level 4

Intelligent decision: levels 2–3

Latency and adversarial robustness: level 1

Source: Hu et al. (2024); Ong et al. (2025); Li et al. (2026); Shafran et al. (2025); OpenAI (2025)

© Aridio Silva @aridiosilva — Outubro 2026

© Aridio Silva @aridiosilva Outubro 2026

Overhead and throughput: measure in your own environment

Router	req/s	US$ / million
SW ranking (CPU)	2.9	39.26
Matrix Factorization	155.16	3.32
BERT	69.62	3.19
Causal LLM	42.46	5.23

RouteLLM algorithms, measured by the authors.

Competitor measurements differ and have a conflict of interest.

≤ 0.4%

maximum router cost over GPT-4 generation

1,170 req/s

LiteLLM, 4 instances (official documentation)

8 ms

gateway p95 overhead in that test

© Aridio Silva @aridiosilva Outubro 2026

Positive and negative implications

Positive

- Savings of over 2× up to 3.66×
- Automatic fallback across providers
- Single governance point
- Model switching by configuration

Negative

- Uncertain algorithm gain
- New point of failure and attack
- Extra latency and recalibration
- User may not know which model answered

© Aridio Silva @aridiosilva Outubro 2026

Recommendations by scenario

Scenario	Recommendation
Production with sensitive data	Self-hosted gateway in a VPC, virtual keys, DLP, auditing and updates against advisories
Cut cost at scale	Start with rules; adopt a learned router only if it beats the baseline on real traffic
Prototypes and startups	Managed aggregator with fallback; data passes through the vendor
Multi-agent	Model and reasoning per role, queues, token bucket and per-run budget
Compliance	Data residency, event logging and control–requirement mapping

© Aridio Silva @aridiosilva Outubro 2026

Conclusion and future work

Conclusion

- The problem admits precise formalization (PGR, APGR, CPT).
- The control plane is mature; the decision algorithm must prove its gain.
- Latency and adversarial robustness are the gaps.

Future work

- Joint routing of model and reasoning effort
- Multi-turn and agentic routing with budgets
- Evaluation under attack
- Benchmarks with 2026 models
- Governance metrics

© Aridio Silva @aridiosilva Outubro 2026

Main sources

- Ong et al. (2025). RouteLLM. ICLR 2025. arXiv:2406.18665
- Li et al. (2026). LLMRouterBench. Findings of ACL 2026. arXiv:2601.07206
- LLMRouter (2026). arXiv:2608.06867
- Chen, Zaharia, Zou (2023). FrugalGPT. arXiv:2305.05176
- Hu et al. (2024). RouterBench. arXiv:2403.12031
- Ding et al. (2024). Hybrid LLM. ICLR 2024. arXiv:2404.14618
- Aggarwal et al. (2024). AutoMix. NeurIPS 2024. arXiv:2310.12963
- Shafran et al. (2025). Rerouting LLM Routers. arXiv:2501.01818
- RerouteGuard (2026). arXiv:2601.21380

43 audited references in the full document, between 28 and 29 Oct 2026.

© Aridio Silva @aridiosilva Outubro 2026

Key messages

Control plane

Gateway, fallback, keys, budgets and auditing are already in wide production.

Intelligent decision

The algorithm's gain must be proven against a simple baseline.

Security

The gateway concentrates keys and data; routing itself can be attacked.

Aridio Silva · Independent Researcher · @aridiosilva · ORCID 0009-0008-2411-6995

© Aridio Silva @aridiosilva Outubro 2026