

Single-Turn and Multi-Turn Interactions in Frontier Generative AI Models (LLMs)

State, cost, reliability, security, and governance

SGAEIA Research Series — Article 25

Aridio Silva — Independent Researcher, Brazil
ORCID 0009-0008-2411-6995

1 November 2026 · CC BY 4.0
DOI: 10.5281/zenodo.23867630

DOI: 10.5281/zenodo.23867630

Single-Turn and Multi-Turn Interactions in Frontier Generative AI Models (LLMs)

State, cost, reliability, security, and governance

SGAEIA Research Series — Article 25
Aridio Silva — Independent Researcher, Brazil
Creator of SGAEIA — Secure Governed Autonomous Edge Intelligence Architecture
ORCID: 0009-0008-2411-6995
1 November 2026
© 2026 Aridio Silva. Text and original illustrations licensed under CC BY 4.0.

Abstract

This article examines how moving from a single-turn interaction to a multi-turn trajectory changes the state, cost, reliability, security, and governance of systems built on large language models. The analysis is a critical narrative review supported by scientific papers, official provider documentation, standards, security taxonomies, and explicitly identified original calculations. A model has no intrinsic conversational memory across independent inferences; continuity is reconstructed by the client or through provider-managed state services. When an application fully replays a history that grows at an approximately constant rate, the logical sum of input tokens grows quadratically with the number of calls, although caching, truncation, compaction, and external memory change billed cost, latency, and effective computation.

Benchmark evidence shows that requirement fragmentation, cross-turn dependencies, intent shifts, information retrieval, and reliance on previous model outputs create different failure modes; there is no single "multi-turn" condition. Evidence on explicit reasoning and memory techniques is mixed and task- and protocol-dependent. In agentic systems, accumulated history also expands gradual attacks, persistence of malicious content, error propagation, and unbounded consumption. The article relates these findings to governed routing and SGAEIA, arguing for a separation between transcript, operational state, and assurance evidence. This relationship is an architectural interpretation and validation agenda, not empirical proof of the framework.

Keywords: LLM; single-turn; multi-turn; conversational state; LLM routing; prompt caching; reliability; security; runtime governance; SGAEIA.

Contents

1. Introduction, scope and method
2. Foundations: turn, state and mechanics
3. History and evolution of methods and techniques
4. Multi-turn performance: the evidence
5. Token and credit economics
6. Security and risk in multi-turn use
7. State of the art in state management and drift mitigation
8. Compliance architecture and the SGAEIA framework
9. Information reliability, verification and audit
10. State of the art and future perspectives
11. Conclusion

1. Introduction, scope and method

Definitions. A **message** is a unit attributed to a role such as user, assistant, system, or tool. An **inference call** is one execution of the model that receives a representation of state and produces an output. An **interaction round** is, in this article, the cycle formed by a new user input and the corresponding system response, possibly including internal tool calls. In the cost equations, T denotes the number of principal calls or rounds as stated; when a benchmark uses a different convention, it is declared explicitly.

Research question. How does the choice between single-turn and multi-turn affect (a) state mechanics, (b) token and credit cost, (c) performance and reliability, and (d) security and governance in frontier LLMs?

Method and source policy. Primary sources are preferred: papers (arXiv, proceedings), official provider documentation and blogs, standards and legislation. Secondary sources are used only when flagged. Every reference was checked for authorship, date, identifier and venue; the audit result is in Section 12 (Bibliography), with **unverified** items marked as such. Prices and reasoning levels are a *snapshot* of Oct./Nov. 2026 taken from official pages, and change often.

Limitations. No provider publishes performance for each model and each reasoning level separately for single-turn and multi-turn. The cost tables use explicit assumptions and are **own calculations**, not provider data.

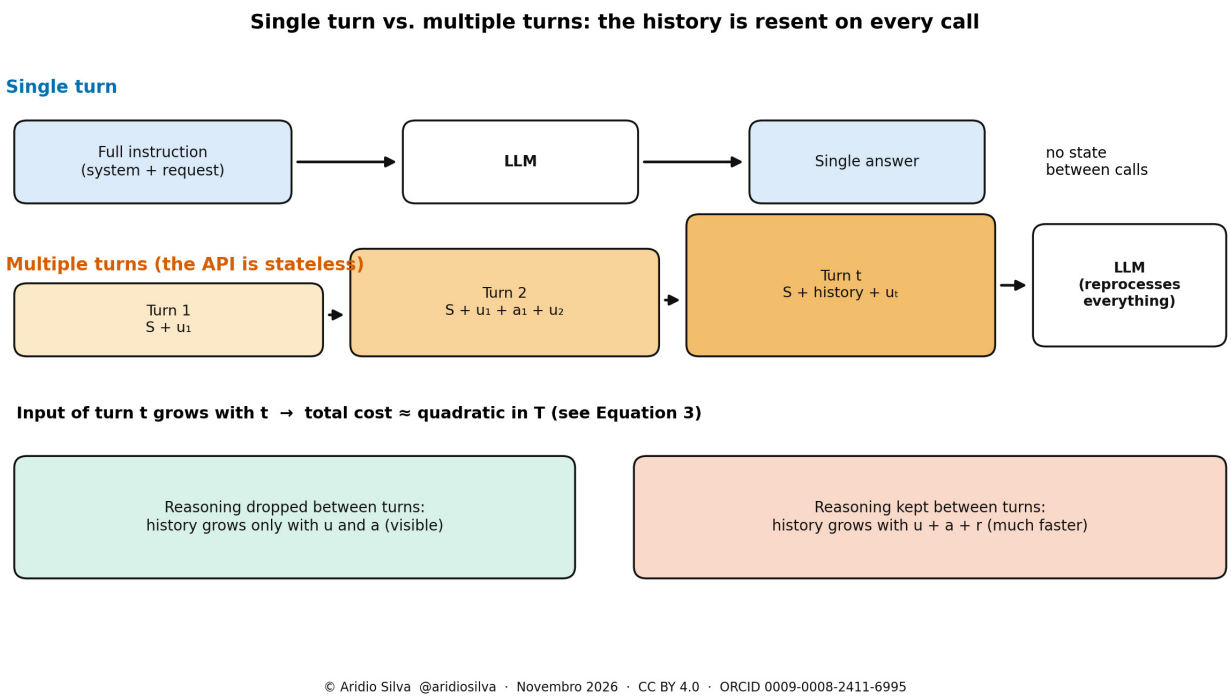
How to read. Each section ends with an analysis box. Equations are numbered images; figures carry text descriptions.

2. Foundations: turn, state and mechanics

2.1 The model has no intrinsic conversational memory

A language model does not, by itself, preserve conversational memory across independent inferences. Continuity may be reconstructed by the client, which replays messages or derived state, or by a provider service that maintains items, identifiers, or server-side compaction. In either case, the current response depends on a selected representation of the past. That representation should be treated as manipulable input subject to truncation, summarization, error, access control, and audit, rather than as an automatically trustworthy record. Three practical consequences:

- **Cost and latency** grow with each turn because the history is reprocessed, and the conversation is bounded by the context window.
- **The client controls the "past".** Earlier turns can be synthetic; for security, the history is a manipulable input surface, not a trustworthy record.
- **Multi-turn is not the same as long context.** A single 100k-token prompt is long context in one turn; multi-turn is defined by incremental disclosure of information and dependence on the model's own earlier answers.



© Aridio Silva @aridiosilva · Novembro 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Figure 1. Single turn (one call, no state) and multiple turns (history resent and growing). Author's diagram.

2.2 Single turn: isolated efficiency

In single-turn use the interaction is limited to one request and one answer, and the model has no inherent memory of earlier interactions.

- **Isolation and relative determinism.** Each call is an atomic event, which eases prompt unit testing, reproducibility and static input/output validation policies.
- **Low latency and cost.** With no accumulated history, input tokens are minimal (prompt caching aside), reducing per-inference cost and *Time to First Token*.
- **Reduced persistence across calls.** The absence of replayed history reduces persistence of malicious content across inferences, but does not eliminate indirect prompt injection in a single call containing untrusted pages, email, documents, or RAG results.

Architectural limits: no iterative refinement (any adjustment means rewriting the prompt) and unfitness for tasks needing step-by-step planning, iterative debugging or design-space exploration.

2.3 Multi-turn: state and context management

In multi-turn use, the accumulated history (user, assistant and optionally tool calls) is reprocessed or managed at each interaction. In long sessions, three strategies avoid saturating the window and degrading attention:

- **Sliding window.** Keeps only the last N interactions; simple, but suffers "catastrophic amnesia" for long-range dependencies.
- **Dynamic summarization.** A subsystem (or the LLM itself in the background) compresses old history into consolidated summaries.
- **External memory / conversational RAG.** History and artifacts are stored in vector or relational stores and retrieved by semantic search according to the current turn's intent.

A related line of research treats the context window as an operating system's main memory. MemGPT [6] proposes *virtual context management*: the main context (instructions, working context and a FIFO queue) is complemented by external storage, with paging driven by function calls.

2.4 Complexity: what is quadratic and what is not

In Transformers [1], attention costs $O(n^2 \cdot d)$ per layer (Equation 1). Quadratic growth is polynomial, not exponential. Two layers must be distinguished:

1. **Within a call**, the prefill of the context has cost quadratic in input size, and each generated token attends to the cached keys at cost linear in the context.
2. **Across the conversation**, if each round adds an approximately constant number of tokens and the application fully replays history without truncation or compaction, the logical sum of input tokens grows as $O(T^2)$ in the number of calls T (Equations 2 and 3). Caching, server-side persistence, and billing policies can make billed cost and effective computation follow different curves.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \qquad \text{cost per layer} = O(n^2\, d)$$

© 2026 Aridio Silva · @aridiosilva · November 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equation 1. Scaled dot-product attention and complexity per layer, after Vaswani et al. [1].

$$I_t = S + tu + (t-1)(a + \kappa r)$$

© 2026 Aridio Silva · @aridiosilva · November 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equation 2. Input at turn t, where S is the system prompt, u the user tokens per turn, a the visible assistant tokens, r the reasoning tokens and κ = 1 if reasoning is kept in the history (κ = 0 otherwise).

$$\sum_{t=1}^{T} I_t = TS + u\frac{T(T+1)}{2} + (a+\kappa r)\frac{T(T-1)}{2} = O(T^2)$$

© 2026 Aridio Silva · @aridiosilva · November 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equation 3. Cumulative input over T turns: quadratic in T.

2.5 Comparison matrix

Table 1

Criterion	Single-turn	Multi-turn
State complexity	Stateless: no state kept	Stateful: needs history management
Inference cost	Low and linear per request	Growing (history reprocessing; ~T^2)
Latency	Minimal	Higher (scales with context size)
Risk surface	Isolated and smaller	Cumulative (context and state pollution)
Ideal cases	Classification, extraction, translation, static pipelines	Interactive assistance, code generation, autonomous agents
Information disclosure	Complete at the start	Incremental; allows underspecification

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 1 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

2.6 Section analysis

Table 2

Dimension	Analysis
Implications	State lives in the client, not in the model. The system designer decides what to resend, summarize, drop and audit.
Strengths	Single-turn: simplicity, predictable cost, testability. Multi-turn: iterative refinement, exploring incomplete requirements, agents.
Weaknesses	Single-turn: does not cover exploratory tasks. Multi-turn: ~T^2 cost, growing latency, cumulative risk and dependence on history management.
Trade-offs	Ability to refine versus cost and risk. A sliding window is cheap and loses dependencies; summarization preserves more and may distort; external memory scales and adds complexity and attack surface.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 2 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

3. History and evolution of methods and techniques

Recent LLM evolution can be read along **seven dimensions** that influence each other. Figure 2 shows only milestones whose date was checked against a primary source (Section 12); unverified milestones were left out on purpose.

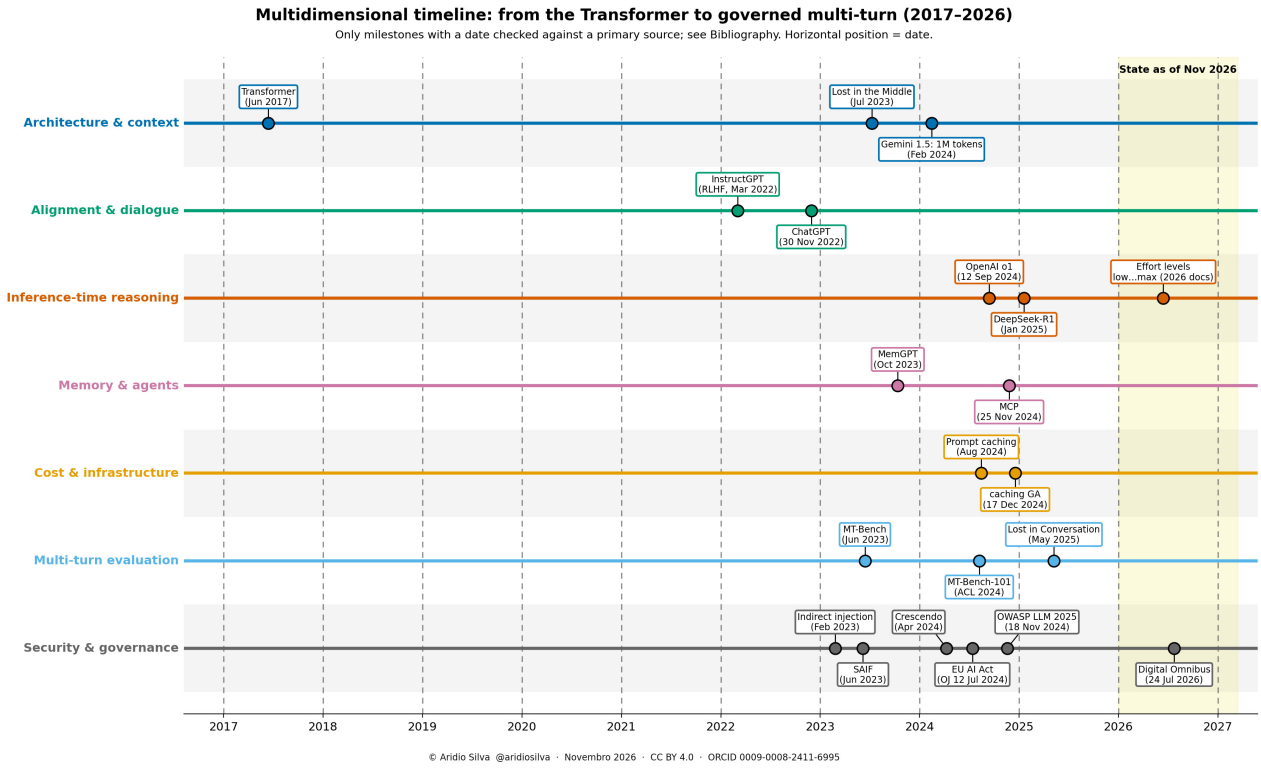


Figure 2. Multidimensional timeline (architecture and context; alignment and dialogue; inference-time reasoning; memory and agents; cost and infrastructure; multi-turn evaluation; security and governance). Author's work based on sources [1]–[22].

3.1 Era 1 (2017–2021): architecture and prompting

The Transformer [1], published in June 2017, replaced recurrence with attention and made training parallelizable. Models were trained with small windows, between 512 and 2,048 tokens [5], and used through *prompting*: a text or a few examples were given and the model completed it [2]. This is, in essence, a single-turn regime.

3.2 Era 2 (2022): alignment and the dialogue format

InstructGPT [2] (Mar. 2022) showed that fine-tuning with human feedback (RLHF) makes a model preferable, on user instructions, to a GPT-3 more than a hundred times larger. On 30 November 2022 OpenAI launched ChatGPT, whose dialogue format lets it answer follow-up questions, admit mistakes and challenge incorrect premises [18]. Here multi-turn stops being a research topic and becomes a **product format**.

3.3 Era 3 (2023): context, memory, evaluation and first attacks

- **Context.** By 2023 there were models with 4k to 100k tokens, but *Lost in the Middle* [5] (Jul. 2023) showed that performance drops when relevant information sits in the middle and that more context does not always help.
- **Memory.** MemGPT [6] (Oct. 2023) proposed context paging and external storage.
- **Evaluation.** MT-Bench [4] (Jun. 2023) established two-turn dialogue evaluation with an LLM judge.
- **Security.** Greshake et al. [3] (Feb. 2023) introduced *indirect prompt injection*: retrieved data acts as "arbitrary code". Google released SAIF [17] (Jun. 2023).

3.4 Era 4 (2024): long context, caching, reasoning and protocols

- **Long context.** Gemini 1.5 [22] (15 Feb. 2024) offered an experimental 1-million-token window (128k default); Google itself acknowledges high computational cost and the need to optimize latency.
- **Caching.** Anthropic's *prompt caching* [20] (Aug. 2024; generally available Dec. 2024) reported up to 90% lower cost and 85% lower latency on long prompts; in a 10-turn conversation with a long system prompt, latency fell from ~10 s to ~2.5 s and cost by 53%.
- **Reasoning.** o1 [19] (12 Sep. 2024) trained the model by reinforcement learning to produce a chain of thought before answering, with performance rising with inference-time thinking (*test-time compute*).
- **Agents and protocols.** MCP [21] (25 Nov. 2024) standardized how assistants connect to tools and data.
- **Evaluation and security.** MT-Bench-101 [7] (ACL 2024) created a hierarchical taxonomy of 13 tasks; *Crescendo* [8] showed multi-turn jailbreaks using benign inputs; OWASP published the 2025 edition of the LLM Top 10 [16]; Regulation (EU) 2024/1689 [14] was published on 12 Jul. 2024.

3.5 Era 5 (2025–2026): reliability, reasoning effort and regulatory recalibration

- DeepSeek-R1 [11] (Jan. 2025) spread reinforcement-trained reasoning in open models.
- Laban et al. [10], published at ICLR 2026 and distinguished as an *Outstanding Paper*, showed degradation for every model evaluated under its underspecified-conversation protocol; the two reasoning models tested did not eliminate the loss.
- In 2026 providers expose reasoning **effort levels** (low to max) as a cost and quality parameter [24]–[29] (Section 5).
- Regulation (EU) 2026/1744 [15], in force since 27 Jul. 2026, postponed the Annex III high-risk obligations to 2 Dec. 2027.

3.6 Cross-cutting reading of the timeline

Three patterns stand out. (1) Each **capability** advance (long context, reasoning) was followed by **cost** advances (caching) and **evaluation** advances, but multi-turn evaluation arrived after the product. (2) **Security** followed the conversation: risk moved from the isolated prompt to the history and to retrieved data. (3) **Governance** moved from principles to runtime controls, the subject of Sections 7 and 8.

3.7 Section analysis

Table 3

Dimension	Analysis
Implications	Multi-turn has become a widely adopted format in conversational products, but it was evaluated, priced and secured after adoption. This explains the gap between laboratory (single-turn) and real-world performance.
Strengths	Fast innovation pace; cost (caching) and memory (paging) mitigations emerged within a few years.
Weaknesses	Many 2026 milestones lack a publicly verifiable primary-source date; the timeline is deliberately conservative. Benchmarks age quickly.
Trade-offs	More context and more reasoning raise capability, cost and attack surface at the same time.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 3 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

4. Multi-turn performance: the evidence

4.1 Benchmarks that introduced multi-turn

MT-Bench [4] evaluated two-turn conversations (for example, ask for a text and then ask for it to be re-explained to a child); MT-Bench-101 [7] expanded to 4,208 turns across 1,388 dialogues and 13 tasks, organized into perception, adaptability and interactivity, and observed that usual alignment techniques and chat-specific designs gave no clear gains in multi-turn abilities. These benchmarks are generally **episodic**: each turn brings a subtask that can be scored in isolation.

4.2 Lost in Conversation (Laban et al., ICLR 2026)

Laban et al. [10] (Microsoft Research and Salesforce Research) compare single-turn and multi-turn **on the same tasks**. The *sharding* method splits each instruction into pieces, and the simulated user reveals at most one per turn. Three main conditions: **FULL** (full instruction in one turn), **CONCAT** (shards concatenated in one turn) and **SHARDED** (one shard per turn).

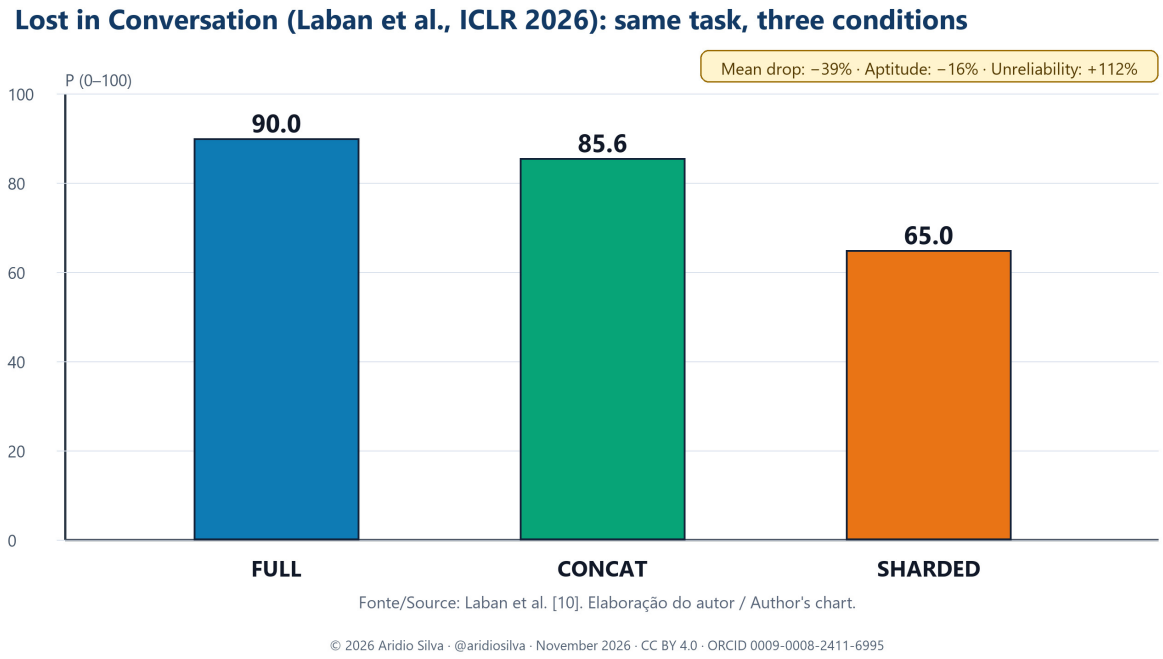


Figure 3. Mean performance P (0 to 100) in the three conditions, per [10]. Bars: 90 (FULL), 85.6 (CONCAT = 95.1% of FULL) and 65 (SHARDED). Author's chart.

Metrics. For each instruction, with N simulations and scores S:

$$P = (1/N)\ \Sigma_i\ S_i$$
$$A^{90} = \text{percentile}_{90}(S)$$
$$U^{90} = \text{percentile}_{90}(S) - \text{percentile}_{10}(S)$$

S = scores across N simulations

© 2026 Aridio Silva · @aridiosilva · November 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equation 4. Mean performance (P), aptitude (A^{90}, the best-case performance) and unreliability (U, the gap between the 90th and 10th percentiles) [10].

Results (v1, 9 May 2025). 15 LLMs from eight families, six tasks (code, database, actions/APIs, math, data-to-text and summarization), over 200 thousand simulated conversations. Mean of 90% in single-turn against 65% in multi-turn; a 39% drop in the abstract and text; aptitude −16% and unreliability +112%. Degradation appears already with **two** turns. CONCAT reaches 95.1% of FULL, so the loss does **not** come from rephrasing. The two reasoning models tested (o3 and DeepSeek-R1) deteriorate similarly and produce answers 33% longer on average, with more assumptions.

Qualitative causes: premature final-answer attempts; answer "bloat" (final answers 20 to 300% longer); *loss in middle turns*; and verbosity. **Temperature zero does not fix it:** with temperature 0 on both sides about 30 points of unreliability remain. **Mitigations:** repeating requirements every turn (SNOWBALL) recovers 15 to 20% of the drop; a final RECAP is better but unrealistic.

Study limitations (noted by the authors): automated simulation, analytical tasks, English and text only. The authors consider that real-world degradation is likely **larger** than measured. **Recommendations:** retry in a fresh conversation and **consolidate** requirements into a single prompt.

4.3 Long context and attention: Lost in the Middle

Liu et al. [5] showed a U-shaped curve: performance is higher when relevant information is at the start or end of the context and drops when it is in the middle, and it decreases as context grows, even in long-context models. Laban et al. observed an analogue across turns (*loss-in-middle-turns*).

4.4 Convergent and divergent evidence from multi-turn benchmarks

LLMs Get Lost in Multi-Turn Conversation is not isolated evidence. The final paper was published at ICLR 2026 and received an *Outstanding Paper* distinction [37], [38]. Its principal result—an average 39% drop when requirements for the same task are disclosed gradually—should be interpreted within its sharded-simulation design: analytical tasks, simulated conversations, English-only text, and models from a specific time window.

Other benchmarks measure partly different failure modes. **MT-Eval** organizes interactions into recollection, expansion, refinement, and follow-up [39]. **MultiChallenge**, published in *Findings of ACL 2025*, evaluates four realistic challenges that jointly require instruction following, context allocation, and in-context reasoning; every evaluated model scored below 50% accuracy, with a best average of 41.4% [40]. **StructFlowBench** models six structural relationships between turns and identifies difficulty in understanding conversational flow [41]. **mtRAG**, published in TACL, adds retrieval through 110 human-generated conversations averaging 7.7 turns and totaling 842 tasks [42].

Evidence about explicit reasoning is mixed. In Laban et al.'s design, the two reasoning models tested did not eliminate degradation and generated longer answers. In contrast, **MARS-Bench** reports that explicit reasoning improves robustness in long conversations involving intent shifts and cross-turn dependencies [43]. The defensible conclusion is not that "more reasoning does not work," but that its effect depends on the task, protocol, memory management, and metric.

Taken together, these studies show that "multi-turn" is not a single condition. At least five mechanisms should be separated: (1) requirements disclosed in fragments; (2) structural dependencies between turns; (3) distraction and intent shifts; (4) retrieval of external evidence; and (5) dependence on the model's own previous answers. A mitigation that helps one—such as recap, compact memory, or explicit reasoning—may fail or increase cost in another.

4.5 Implications for state, cost, and SGAEIA

The results strengthen the case for representing task state separately from the transcript. Confirmed requirements, validated facts, revoked decisions, permissions, and tool results should not depend only on their position in textual history. A governance architecture can maintain typed contracts, provenance, and deterministic policies while using conversation as the interface. This relationship is **the author's interpretation and an SGAEIA architectural hypothesis**, not an empirical validation of the framework. Its effectiveness should be tested against baselines such as full replay, sliding windows, rolling summaries, and retrieved memory.

4.6 An illustrative error-propagation model

To fix intuition (and **not** as an empirical result), suppose that at each turn there is a probability ε that the model introduces a wrong premise that the user accepts (hallucination cascade). The probability that a T-turn conversation has no accumulated error is given by Equation 6. With ε = 5% and T = 20, about 36% remains.

$$\Pr(\text{no error in } T \text{ turns}) = (1-\varepsilon)^T \approx e^{-\varepsilon T}$$

© 2026 Aridio Silva · @aridiosilva · November 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equation 6. Didactic compound-error model; ε and T are hypothetical parameters.

4.7 Section analysis

Table 4

Dimension	Analysis
Implications	Single-turn performance overestimates real-world use. The metric that degrades most is reliability, not aptitude.
Strengths	Controlled design (same tasks, three conditions, 200-thousand-conversation scale) and well-defined metrics (P, A, U).
Weaknesses	2025 models; idealized simulation; analytical tasks in English. No equivalent public data for 2026 generations or for each effort level.
Trade-offs	Consolidating the request in one turn raises reliability but loses multi-turn's exploratory advantage; summaries and recaps recover part of the drop at the cost of more tokens.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 4 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

5. Token and credit economics

5.1 Billing mechanics

Billing has three components: **input** tokens, **output** tokens (including reasoning tokens, billed as output) and the **cache read** of repeated input. In multi-turn use almost all resent history is read from cache, so the **cache-read** price differentiates providers more than the input price.

$$C = \frac{1}{10^6} \sum_{t=1}^{T} \left[p_w N_t^{\text{new}} + p_c \left(I_t - N_t^{\text{new}} \right) + p_o \left(a + r \right) \right]$$

© 2026 Aridio Silva · @aridiosilva · November 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equation 5. Cost of a T-turn conversation with cache: p_w (write), p_c (cache read), p_o (output), in USD per million tokens; N_new is the new part of the input at each turn.

5.2 Reasoning levels by provider (Nov. 2026 snapshot)

Table 5

Provider	Model	Reasoning levels	Default
OpenAI	GPT-6 Astra	low → max	not confirmed
OpenAI	GPT-6.1 Sol	low → max	medium
OpenAI	GPT-6.1 Terra, GPT-6 Luna	none → max	medium (Luna)
Anthropic	Fable 5.1, Sonnet 5.5	low → max	high
Anthropic	Opus 5.5, Haiku 5.5	low → max	medium
Google	Gemini 4 Argon, 3.8 Flash	low, medium, high	medium
Google	3.5 Flash-Lite	minimal → high	minimal
xAI	Grok 4.7	low → xhigh	high
DeepSeek	V4.1-Flash, V4-Pro	with or without reasoning	with reasoning

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 5 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Availability of each level varies by model; Anthropic points to the effort* page as the reference [24]. Sources: [24]–[29]. Names and levels change often; confirm on the official pages.

5.3 How each provider treats earlier turns' reasoning

- **OpenAI** [25]: in models before GPT-5.6, past-turn reasoning does not enter the next turn's context; the GPT-5.6 family includes it by default, and the `reasoning.context` parameter selects the behavior (`all_turns` is supported by GPT-5.6 and GPT-6.1 Sol). A `configuration_update` item changes effort between responses without altering the prefix, preserving the cache.
- **Anthropic** [24]: reasoning blocks of earlier turns that stay in context are billed as input (all turns in some models, only the last in others). Changing *effort* at the top of the request **invalidates the cache**; per-message change (beta) preserves cache points. Per-request reasoning tends to decrease as the conversation grows long.
- **Google** [27]: *thought signatures* keep reasoning continuity; in stateless mode, thought blocks must be resent exactly as received.
- **xAI** [28]: grok-4.7 always returns encrypted reasoning, which must be sent back unchanged to keep reasoning across turns; reasoning cannot be turned off.
- **DeepSeek** [29]: with and without reasoning modes; peak and off-peak prices.

5.4 Prices (USD per million tokens) and cache weight

Table 6

Model	Input	Output	Cache read	Cache / input
GPT-6 Astra	10.00	50.00	1.00	10%
GPT-6.1 Sol	2.00	10.00	0.10	5%
GPT-6.1 Terra	1.00	5.00	0.05	5%
GPT-6 Luna	0.10	0.50	0.01	10%
Claude Fable 5.1	10.00	50.00	0.25	2.5%
Claude Opus 5.5	4.00	20.00	0.20	5%
Claude Sonnet 5.5	2.00	10.00	0.10	5%
Claude Haiku 5.5 (≤100k)	0.10	0.50	0.01	10%
Gemini 4 Argon (until 30 Nov.)	2.00	10.00	0.10	5%
Gemini 3.8 Flash (until 31 Dec.)	0.75	3.75	0.075	10%
Grok 4.7 (<200k)	2.00	6.00	0.50	25%
DeepSeek V4-Pro (peak)	1.32	3.96	0.044	3.3%
DeepSeek V4.1-Flash (peak)	0.30	1.20	0.006	2%

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 6 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Provider caveats: Haiku 5.5 costs more above 100k tokens; Gemini 4 Argon doubles in price from 1 Dec. 2026 and offers only implicit caching; grok-4.7 doubles above 200k tokens; DeepSeek charges half off-peak; and Claude models from 4.7 on use a tokenizer that produces about 30% more tokens for the same text, so per-token prices are **not comparable** 1:1 across providers [23]. Sources: [23], [25], [26], [28], [29].

5.5 A 20-turn example (own calculation)

Assumptions: S = 2,000, u = 300, a = 700 tokens; reasoning r = 1,000 (low effort) or 8,000 (high); the cache always hits; for this simplified scenario, cache writes carry no surcharge. This assumption is not universal: providers may price writes, TTLs, or cache modes separately. By Equation 3, cumulative input over 20 turns is **236 thousand tokens** (21.3k new and 214.7k read from cache) with reasoning dropped; if 8,000 tokens of reasoning were **kept** in the history, input would rise to **1.756 million** (7.4 times), with 173.3k input tokens at the 20th turn.

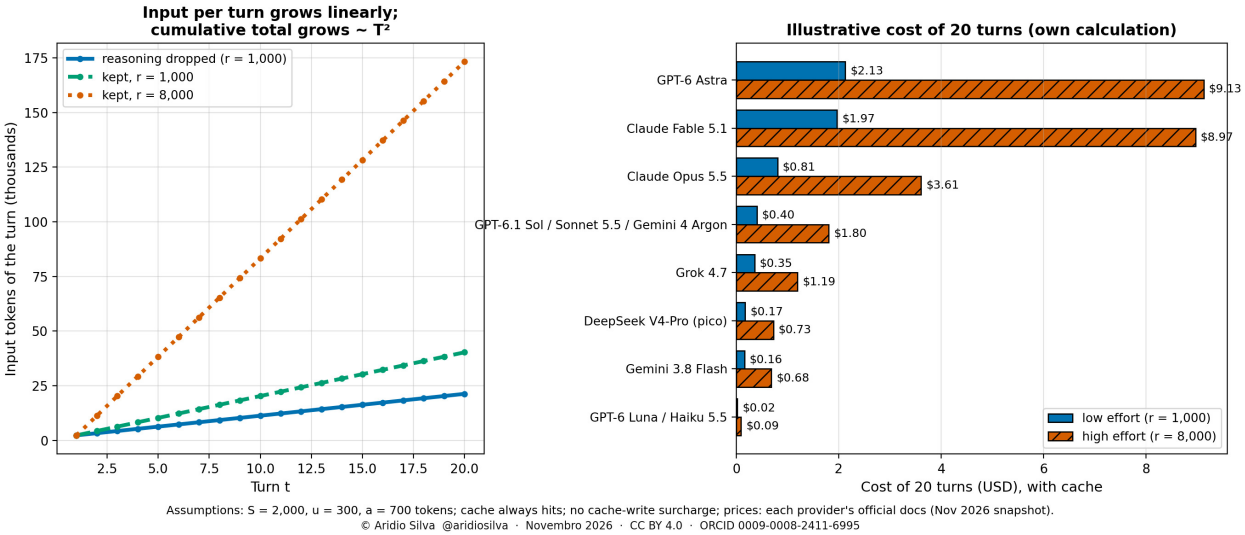


Figure 4. Left, input tokens per turn in three scenarios; right, illustrative 20-turn cost per model, with cache. Author's calculation on official prices.

Table 7

Model	20 turns, low effort	20 turns, high effort	No cache, low effort
GPT-6 Astra	USD 2.13	USD 9.13	USD 4.06
Claude Fable 5.1	USD 1.97	USD 8.97	USD 4.06
Claude Opus 5.5	USD 0.81	USD 3.61	USD 1.62
GPT-6.1 Sol / Sonnet 5.5 / Gemini 4 Argon	USD 0.40	USD 1.80	USD 0.81
Grok 4.7	USD 0.35	USD 1.19	USD 0.68
DeepSeek V4-Pro (peak)	USD 0.17	USD 0.73	USD 0.45
Gemini 3.8 Flash	USD 0.16	USD 0.68	USD 0.30
GPT-6 Luna / Haiku 5.5	USD 0.02	USD 0.09	USD 0.04

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 7 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Readings. With cache, **output** (mainly reasoning) dominates: going from low to high effort multiplies cost by 3.4 to 4.5 times. Caching cuts input cost by about 86% in the example (models with cache reads at 5% of input), and Anthropic's documentation [20] reports gains of the same order (−53% cost and −75% latency over 10 turns). On cost security, OWASP lists *Unbounded Consumption* (LLM10:2025) as a risk: uncontrolled inference consumption becomes a cost attack [16].

5.6 How to compare models and effort levels (proposed protocol)

1. Choose tasks with verifiable answers.
2. Run three conditions: full instruction (FULL), concatenated shards (CONCAT) and shards per turn (SHARDED) [10].
3. Cross model and effort level, comparing providers **only at equivalent levels** (for example, high with high).
4. Measure P, A^{90} and U (Equation 4), repeating N ≥ 10 simulations per pair.
5. Measure real cost from the response usage fields (reasoning tokens, cache reads and writes), **keeping effort fixed** during each conversation so as not to invalidate the cache.

5.7 Section analysis

Table 8

Dimension	Analysis
Implications	With caching, the effort level weighs more on cost than the number of turns. Without cache, cost grows ~T^2.
Strengths	Caching and effort control give the engineer two explicit levers. Provider documentation is detailed.
Weaknesses	Prices, names and levels change fast; different tokenizers prevent direct comparison; there is no public quality data per level and per turn.
Trade-offs	Keeping reasoning across turns gives continuity but can multiply input by ~7 times in the example; changing effort mid-conversation may invalidate the cache (Anthropic) or be done without breaking it (OpenAI).

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 8 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

6. Security and risk in multi-turn use

From a security and governance standpoint, multi-turn use changes the system's risk profile. This section gathers the scientific evidence and the risks described in the compendium.

6.1 Attacks that depend on several turns

- **Gradual escalation (*Crescendo*).** Russinovich, Salem and Eldan [8] describe a jailbreak that uses only benign inputs and exploits the model's tendency to follow patterns and attend to recent text, especially what it generated itself. Instead of asking for something forbidden at once (which would be refused in a single turn), the attacker escalates gradually. The authors point to the evaluation gap: existing jailbreak benchmarks covered only single turns.
- **Why single-turn defenses fail.** A later paper from the same team [9] reports that aligned models tend to represent Crescendo responses internally as more benign than harmful as turns increase, which helps explain why defenses designed for single turns, such as representation *circuit breakers*, fail.
- **Indirect prompt injection.** Greshake et al. [3] showed that retrieved data (pages, e-mails, documents) can work as instructions, "blurring the line between data and instructions", with impacts such as data theft, *worming* and contamination of the information ecosystem. In multi-turn use, an injection placed in an intermediate turn can **persist** in the history and hijack the agent's behavior in later turns.

6.2 Risks specific to the accumulated history

- **Persistence of malicious state.** In an agentic environment, the content of a page read by a tool can contaminate the state and reappear in every later turn.
- **Hallucination cascade (*error propagation*).** A subtle error in turn 2, implicitly accepted by the user ("ok, continue"), is incorporated into the conversation's official state and contaminates later reasoning (see the Equation 6 model). Laban et al. [10] corroborate: models "overly rely on previous incorrect attempts".
- **Difficulty auditing authority limits.** In regulated architectures, ensuring that a fixed-scope directive is applied across dozens of turns is computationally harder than in isolated calls.
- **Unbounded consumption.** Long conversations with high reasoning are a cost-attack vector (LLM10:2025, *Unbounded Consumption*) [16].

6.3 Reference taxonomies

The OWASP Top 10 for LLM Applications (2025 edition, 18 Nov. 2024) keeps prompt injection in first place (LLM01) and includes *Excessive Agency* (LLM06), *Misinformation* (LLM09) and *Unbounded Consumption* (LLM10) [16]. Google's SAIF [17] is a conceptual framework of six elements that lists, among AI-specific risks, model theft, data poisoning, prompt injection and data extraction.

6.4 Section analysis

Table 9

Dimension	Analysis
Implications	Authorization cannot depend on the model's judgment across the conversation; the history must be treated as untrusted input and be auditable.
Strengths	Mature taxonomies (OWASP, SAIF) and experimental evidence of the attacks (Crescendo, indirect injection).
Weaknesses	Security evaluations are still mostly single-turn; defenses based on internal representations fail against gradual attacks [9].
Trade-offs	Reinforcing anchors, summaries and per-turn filters reduces risk and adds latency and cost; over-blocking harms conversational usefulness.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 9 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

7. State of the art in state management and drift mitigation

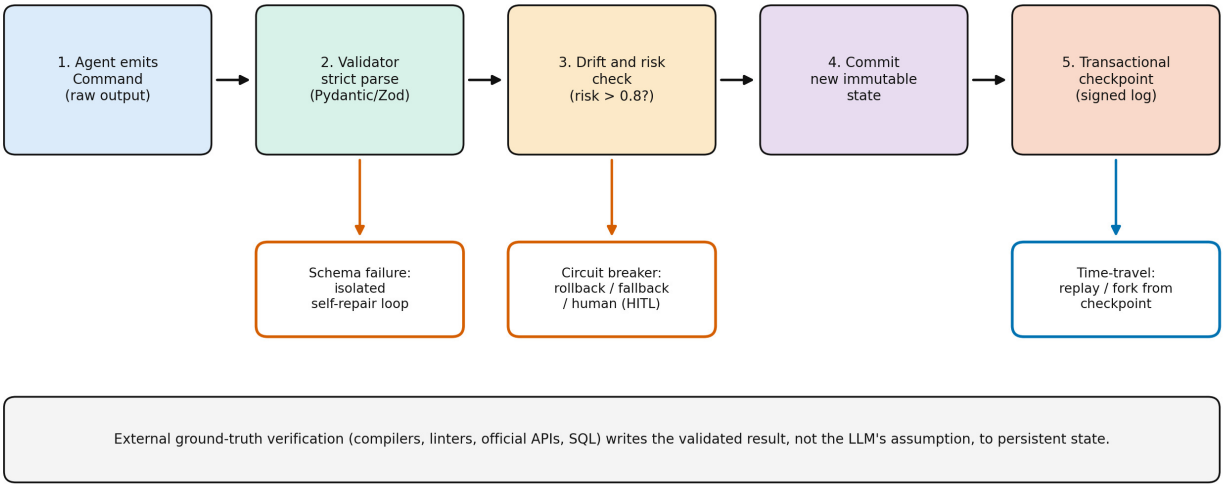
This section incorporates, in full, the content of the author's technical compendium [36] (Sections 2 to 4 of the original), as a technical basis for this analysis.

7.1 State management

Agent systems based on execution graphs can represent state as a **typed, immutable, event-persisted object**, not as the coupled history of a chat.

- **Hierarchical memory.** *Working memory* (short term): the immediate task context, with a strict token limit. *Episodic memory* (medium term): compressed history of relevant interactions in vector stores with temporal and semantic indexing. *Semantic/procedural memory* (long term): static knowledge graphs or externally validated business rules that the agent consults instead of memorizing in the prompt. (Compare with MemGPT's hierarchy [6].)
- **Deterministic state machines (FSMs) with *event sourcing*.** The agent flow is modeled as a finite state machine or directed acyclic graph (DAG); each transition produces an auditable event. This enables *time-travel debugging* (pause, inspect, fork or roll back the state to an exact turn) and transactional *checkpoints* (persistence at each node, for example in PostgreSQL with JSONB), with resilience to infrastructure failures.
- **Strict typing contracts per transition.** Each turn's outputs pass through strict validation parsers (Pydantic, Zod) before entering the global state. If the model violates the schema, the framework rejects the transition and triggers an isolated self-repair loop.

Transactional transition cycle with type validation and drift check



© Aridio Silva @aridiosilva · Novembro 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Figure 5. Transactional transition cycle: command, validation, drift check, immutable commit and checkpoint, with failure branches, circuit breaker and time-travel. Author's diagram based on the compendium.

7.2 Drift mitigation

Drift shows up in three forms: **semantic** (loss of focus on the goal), **behavioral** (deviation from safety and system guidelines) and **epistemic** (accumulation of hallucinations accepted as truth).

A. Semantic and contextual drift

- *System prompt anchorage*. Core directives and constraints are not injected only at the start; advanced routers re-inject anchors at strategic points of the history or use attention with positional bias. This echoes the recommendation to repeat requirements [10] and the position effect [5].
- *Goal-directed summarization*. Instead of linear summaries, the state keeps a tree of pending and completed objectives; old history is dropped or summarized according to progress toward the original goal.

B. Behavioral and alignment drift

- *Layered guardrails (runtime circuit breakers)*. The router acts as a governance *proxy* between turns; lightweight classifiers inspect state and intermediate outputs and, on policy deviation, the circuit trips, triggering fallback or human intervention (HITL).
- *Adversarial validation (critic loops)*. In critical flows, the state generated by an executor is submitted to an independent critic agent that checks it against the original constraints.

C. Epistemic drift (cumulative error)

- *External ground-truth verification*. When a turn generates a new premise (code, financial data), the framework forces a validation tool to run (compilers, linters, official APIs, SQL) and writes the **validated result**, not the LLM's assumption, into the state.

7.3 Practical implementation: state contracts and runtime checks

Implementation requires moving from malleable dictionaries to **immutable, typed state contracts** with transactional validation. The compendium's code follows, preserved as in the original.

State structure with Pydantic

```
from enum import Enum
from pydantic import BaseModel, Field, field_validator
from typing import List, Dict, Any, Optional
from datetime import datetime

class AgentRole(str, Enum):
    RESEARCHER = "researcher"
    EXECUTOR = "executor"
    CRITIC = "critic"
    ROUTER = "router"

class TaskStatus(str, Enum):
    PENDING = "pending"
    IN_PROGRESS = "progress"
    VALIDATED = "validated"
    REJECTED = "rejected"

class GlobalState(BaseModel):
    session_id: str
    current_step: int = Field(ge=0, description="Contador estrito de turnos/transições")
    goal_tree: Dict[str, TaskStatus] = Field(default_factory=dict)
    execution_log: List[Dict[str, Any]] = Field(default_factory=list)
    risk_score: float = Field(default=0.0, ge=0.0, le=1.0)

    @field_validator('execution_log')
    def validate_log_immutability(cls, v):
        # Garante que o log apenas cresce, nunca é reescrito de forma destrutiva
        return v

    class Config:
        frozen = True # Imutabilidade estrutural (requer cópia via model_copy para atualizar)
```

The transactional transition cycle. (1) The agent produces a *Command* (raw output). (2) The validator parses it into the corresponding Pydantic schema. (3) If successful, the *State Manager* generates a new immutable version of the state (`new_state = state.model_copy(update=...)`) and records the event in a transactional database.

Resilient orchestrator

```
def orchestrate_turn(state: GlobalState, agent_input: str) -> GlobalState:
    # 1. Validação de pré-condições e estado
    validate_circuit_breaker(state)

    # 2. Execução isolada do agente
    raw_output = run_agent_safely(agent_input, state)

    # 3. Parse e Tipagem Estrita
    try:
        structured_output = AgentPayload.model_validate_json(raw_output)
    except ValidationError as e:
        return handle_malformed_output(state, e)

    # 4. Checagem de Drift Semântico e de Risco
    risk = evaluate_drift_and_risk(state, structured_output)
    if risk > 0.8:
        return trigger_circuit_breaker_rollback(state)

    # 5. Commit Transacional do Novo Estado
    new_state = state.model_copy(update={
        "current_step": state.current_step + 1,
        "execution_log": state.execution_log + [structured_output.dict()],
        "risk_score": risk
    })

    persist_checkpoint(new_state)
    return new_state
```

(Code comments are kept in the original Portuguese.)

Minimal Pydantic v2-compatible variant

```
from pydantic import BaseModel, ConfigDict, Field

class GlobalState(BaseModel):
    model_config = ConfigDict(frozen=True)   # shallow immutability
    session_id: str
    current_step: int = Field(ge=0)
    goal_tree: dict[str, "TaskStatus"] = Field(default_factory=dict)
    execution_log: tuple[dict, ...] = ()      # tuple: growth by new copy
    risk_score: float = Field(default=0.0, ge=0.0, le=1.0)

# commit: new_state = state.model_copy(update={"execution_log": state.execution_log + (payload.
model_dump(),)})
```

7.4 Time-travel debugging in AI agents

Time-Travel Debugging is the mechanism of freezing, inspecting, rolling back and forking agent execution through continuous recording of immutable state snapshots.

- **Non-deterministic factor.** Because LLMs are stochastic, rerunning a prompt from scratch may take another path; restoring the state snapshot allows debugging from the point before the failure.
- **Fighting the butterfly effect.** If a hallucination occurs in turn 2, time-travel lets you return to it, fix the corrupted variable or tool output and create a *fork* without reprocessing earlier steps.
- **Explainability and forensic audit (*decision provenance*).** It works as the agent's "black box" in regulated scenarios: it reconstructs what the agent "knew", which tools it called and what the state was at failure.

Operational limitation. In LangGraph's official documentation [31], *time travel* has two operations: **replay** (resume from an earlier checkpoint) and **fork** (branch with modified state); checkpoints are saved at each *super-step*. But **replay re-executes** the nodes after the checkpoint: LLM and API calls fire again and **may return different results**. So debugging is only *deterministic* if LLM and tool outputs are **recorded** (event sourcing of the responses) and reused. The documentation does not mention restoring *seeds*.

7.5 Section analysis

Table 10

Dimension	Analysis
Implications	Agent state should be an engineering artifact (typed, versioned, auditable), separate from conversation text.
Strengths	Type contracts block malformed outputs before they contaminate state; checkpoints enable recovery and audit; external verification fights epistemic drift.
Weaknesses	Engineering complexity; schema validators do not detect semantic error; the risk threshold (0.8) and the drift computation are heuristic and need calibration; immutability in Python is shallow.
Trade-offs	More checks per turn reduce error and add latency and cost; aggressive summarization saves tokens and may erase constraints; rollback requires external side effects to be reversible.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 10 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

8. Compliance architecture and the SGAEIA framework

This section incorporates Sections 5 to 7 of the compendium [36], as a technical basis for this analysis. Statements about SGAEIA are **the author's own** (original work) and have not undergone independent evaluation.

8.1 Strict compliance and runtime governance

Strict-compliance architecture represents the shift from static governance to **runtime enforcement engineering**. The core principle: **the LLM is never part of the critical authorization path (*trust path*)**; security comes from deterministic invariants.

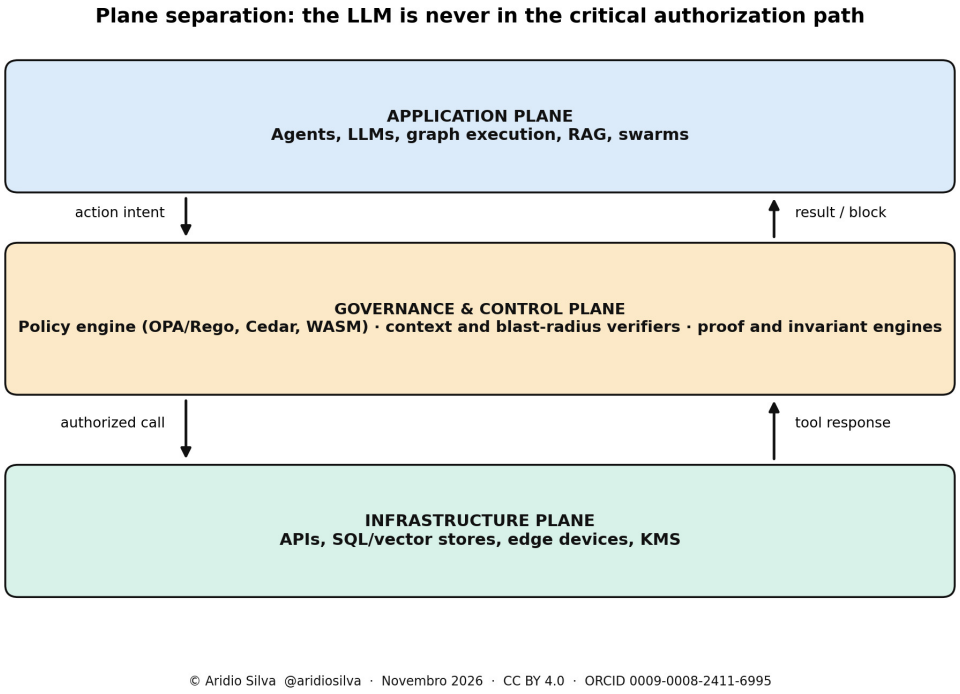


Figure 6. Application plane, governance and control plane, and infrastructure plane. Author's diagram based on the compendium.

Fundamental layers

- **Verifiable identity and attribution.** Unique cryptographic identities per agent (X.509, SPIFFE, mTLS). Delegation imposes **strict-subset restriction** (agent B never inherits more permissions than agent A).
- **Bounded authority.** Granular, temporary, task-scoped tool access; non-negotiable invariants encoded in deterministic engines (OPA/Rego, Cedar).
- **Runtime interception.** Pre-action *gate* (parameters, permissions, signatures); in-action *circuit breakers*; post-action auditors that inspect tool responses and prevent indirect injection.
- **Tamper-evident lineage.** Cryptographically signed *append-only* records contribute to traceability, accountability, and demonstration of controls without guaranteeing compliance in isolation.

Regulatory scope. Articles 12 (record-keeping) and 14 (human oversight) of Regulation (EU) 2024/1689 [14] apply to high-risk systems. Append-only, tamper-evident records contribute to compliance but do not guarantee it in isolation. NIST AI RMF 1.0 is document NIST AI 100-1 [13]. **Action classification and human oversight matrix**

Table 11

Risk level	Action type	Framework behavior	Example
Low	Auto-approved	Direct execution with signed log.	Reading technical documentation, summarizing logs.
Medium	Notify-and-proceed	Notifies operator/SRE in real time and continues.	Changing non-critical settings, intermediate reports.
High/critical	Human-in-the-loop (HITL)	The circuit pauses; requires explicit signed approval from a human.	Financial transactions, changing the production database, e-mailing customers.
Inadmissible	Forbidden (fail-closed)	Immediate block by deterministic policy; alert logged.	Infrastructure privilege escalation, private-key exfiltration.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 11 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

8.2 The SGAEIA framework

SGAEIA (*Secure Governed Autonomous Edge Intelligence Architecture*) is an **open, neutral reference architecture**, conceived by **Aridio Silva** [34], [35], aimed at governance, security and runtime control of **autonomous agents and distributed/edge AI**.

Core premise. In agentic environments, the object of assurance is the **complete system in operation**, not the isolated language model; governance is intrinsic to the execution infrastructure, not a static post-processing step.

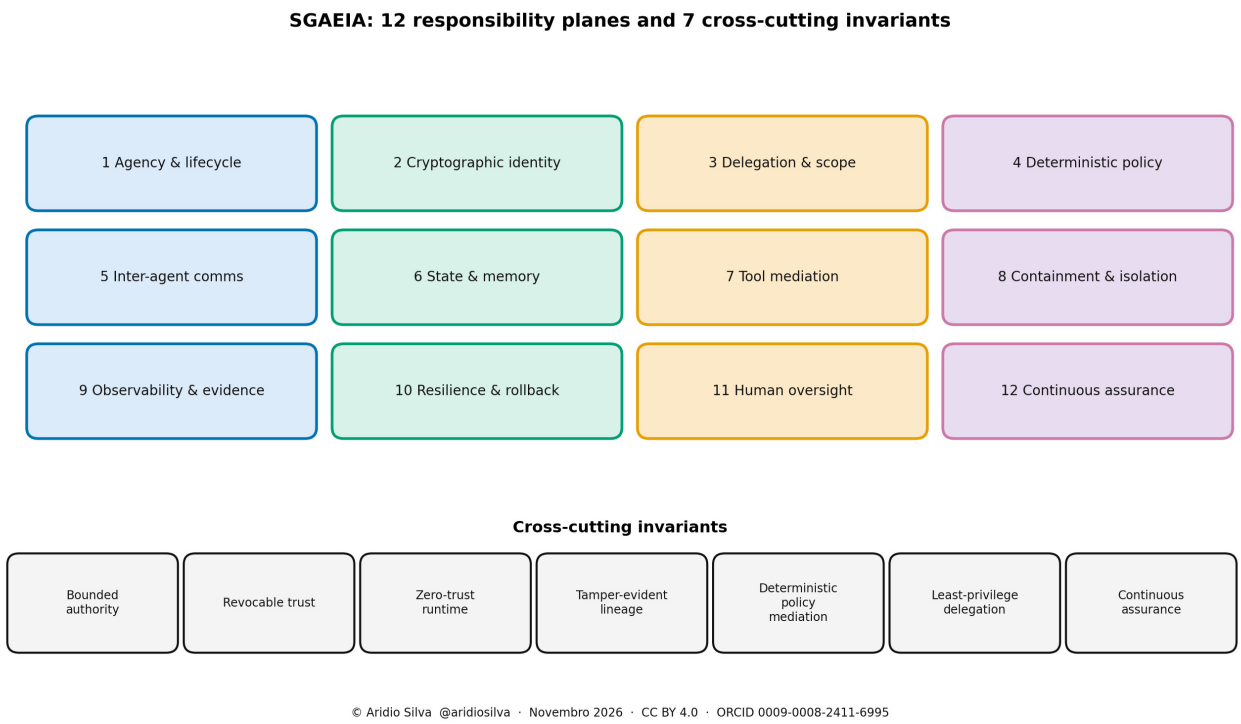


Figure 7. SGAEIA's twelve responsibility planes and seven cross-cutting invariants, per the compendium.

The twelve planes: (1) agency and agent lifecycle; (2) cryptographic identity and attribution (unique SPIFFE/X.509 per agent); (3) delegation and authority scope (strict subset); (4) deterministic policy and enforcement (*policy-as-code*: OPA/Cedar/WASM); (5) inter-agent communication and bus; (6) state management and persistent memory (immutable, versioned, auditable); (7) tool integration and API mediation (pre/in/post-action interception); (8) containment, limits and isolation at the edge (*sandboxing*, WASM, microVMs); (9) observability, evidence and auditability (*Evidence-as-Code*, *append-only*); (10) resilience, rollback and recovery (*circuit breaking* and *time-travel rollback*); (11) human oversight and control interfaces (HITL portals); (12) continuous assurance and risk assessment (monitoring of `risk_score` and compliance).

The seven cross-cutting invariants: *Bounded Authority* (autonomy strictly limited to the current task); *Revocable Trust* (revocation with bounded, observable, and tested latency); *Zero-Trust Runtime Enforcement* (no transition is implicitly trusted); *Tamper-Evident Lineage* (cryptographically verifiable audit trail); *Deterministic Policy Mediation* (policy mediated outside the LLM decision path and not directly modifiable by conversational content); *Least-Privilege Delegation* (strict limits on context passing between agents); *Continuous Assurance* (continuous validation of compliance and risk mitigation).

Relation to this study. Plane 6 (state), 10 (rollback) and 12 (continuous assurance) directly address the multi-turn risks of Sections 4 to 7; Planes 3, 4 and 7 address injection and escalation risk (Section 6).

8.3 Mapping the compliance and governance ecosystem

- **Regulatory and normative.** *EU AI Act* (Regulation 2024/1689) [14], with traceability by logs (Art. 12) and human oversight (Art. 14) for high-risk systems; *NIST AI RMF 1.0* (functions *Govern, Map, Measure, Manage*) [13]; *Google SAIF* [17].
- **Policy engines (*policy-as-code*).** *Open Policy Agent* (OPA/Rego), the CNCF declarative language for authorization in microservices and *AI gateways*; and **Cedar**, an open-source authorization language originated at AWS, formally modeled in Lean and with SMT-based analysis, which outperformed OpenFGA and Rego in the authors' tests [12].
- **AI gateways and guardrails.** *AI gateways* (Portkey, LiteLLM, Langfuse, TrueFoundry) for cost control, PII sanitization and routing; *NeMo Guardrails* (NVIDIA) and *Llama Guard* (Meta) as programmable semantic filtering layers.
- **SGAEIA positioning.** Unlike commercial proxies or generic regulations, it positions itself as an **open, neutral reference architecture**: the engineering *blueprint* that integrates policy engines, cryptographic identities, immutable state machines and *Evidence-as-Code*.

8.4 Governed routing in multi-turn trajectories

Article 24 of the SGAEIA Research Series treated routing as a governed decision about models, data, resources, and trajectories [49]. This article develops the temporal unit of that decision. In a multi-turn interaction, the router does not receive only a new question: it receives a representation of the past, possibly summarized, retrieved, or reconstructed. If that representation omits a constraint, incorporates an error, or loses the chain of authority, selection may be technically efficient and still invalid.

The admissible set of models and configurations may change during the conversation because of active identity, data classification, remaining budget, required tools, external effects already executed, and revoked or updated policies. The router should optimize only within that set. Fallback must not turn a safety refusal into an outage to bypass, or send data to a deployment that does not satisfy the same conditions.

A model switch is also a state transition. Models differ in tokenization, tool use, structured formats, preservation of reasoning, interpretation of instructions, and operating policies. Substitution preserves the trajectory only when it retains mandatory constraints, required capabilities, applicable authority, and necessary evidence, or when any loss is detected and handled before execution.

Table 12

State field	Governance purpose
task_id / trajectory_id	Relate decisions to the same task and trajectory.
active_constraints	Preserve mandatory requirements, revocations, and prohibitions.
verified_facts	Separate validated evidence from conversational assertions.
identity_and_authority	Determine who may request, delegate, and approve.
data_classification	Restrict providers, regions, retention, and logging.
required_capabilities	Prevent fallback to an incompatible configuration.
tool_effects	Record external effects already produced.
budget_state	Limit tokens, cost, time, repetition, and concurrency.
model_lineage	Preserve model, version, effort, parameters, and changes.
evidence_refs	Enable audit without relying only on the transcript.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 12 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

This table is a conceptual model, not a completed normative SGAEIA specification. The central synthesis is: **the transcript is a narrative interface; governed state is an operational artifact; the evidence trail is an assurance artifact**. The three are related but should not be conflated.

8.5 Section analysis

Table 13

Dimension	Analysis
Implications	Security becomes a property of the system (identity, policy, state, trail), not of the model. It is the architectural answer to multi-turn cumulative risk.
Strengths	Plane separation; deterministic authorization mediation outside the LLM; verifiable evidence; alignment with standards and taxonomies.
Weaknesses	Deployment cost and complexity; SGAEIA is the author's own work, with no published independent evaluation; requires operational maturity (PKI, policy as code).
Trade-offs	More deterministic control reduces agent autonomy and speed; HITL raises safety and creates a human bottleneck; immutable trails raise auditability and data volume.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 13 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

9. Information reliability, verification and audit

9.1 What Anthropic's Help Center says

Anthropic's support article (5 Mar. 2026) [33] acknowledges that Claude can produce incorrect or misleading answers ("hallucination"), mentions outdated information and citations that sound authoritative without basis, and recommends **not treating the model as a single source of truth** and **checking the sources** in search results. **Strengths:** it admits that error can sound convincing and distinguishes model error from synthesis error over real sources. **Limitations:** it does not separate error types (staleness, invention, reasoning, synthesis), does not quantify frequency, does not mention the multi-turn effect (Section 4) and does not teach **how** to verify.

9.2 Verification protocol applied to this work

For any reference: (1) open the DOI or identifier and check title, authors and year; (2) find the passage that supports the claim; (3) confirm the venue (a preprint is not a published paper); (4) flag what was **not** verified.

9.3 Section analysis

Table 14

Dimension	Analysis
Implications	In technical or academic work, every reference generated or summarized by an LLM must be verified against a primary source; multi-turn use raises the chance of accepted and propagated error.
Strengths	Simple, reproducible protocol; the audit found real errors in already-written material.
Weaknesses	Verification takes time; some sources (DOI resolvers, legal texts) were not accessible in this session.
Trade-offs	Rigor versus speed; secondary sources speed things up and lower confidence, so they were flagged.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 14 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

10. State of the art and future perspectives

10.1 State of the art (Nov. 2026)

1. **Multi-turn reliability as an explicit goal.** Laban et al. propose goals: similar aptitude in single-turn and multi-turn and U < 15 at T = 1.0 [10]. Practical mitigations: consolidate requirements, recap (SNOWBALL), restart conversations.

2. **Economics by cache and by effort.** Prompt caching strongly reduces the cost of repeated input [20]; effort level controls output cost. Providers differ on whether to preserve reasoning across turns (Section 5.3).

3 **State as engineering.** Typed, immutable, versioned state with checkpoints, *fork* and external verification (Section 7); hierarchical memory with paging [6].

4 **Runtime governance.** Deterministic authorization outside the LLM, cryptographic identity, immutable trail (Section 8) and alignment with standards (NIST [13], EU AI Act [14, 15], OWASP [16]).

5 **Integration protocols.** MCP [21] standardizes tools and data, expanding both usefulness and the indirect-injection surface [3].

10.2 Gaps and open problems

- There is no **public** evaluation, per model **and** per effort level, in single-turn and multi-turn for the 2026 generations.
- Cache, tokenizer and effort-level effects are rarely normalized in comparisons.
- Defenses against gradual attacks are still fragile [9].
- The empirical efficacy of SGAEIA and similar architectures lacks independent evaluation.
- Laban et al.'s results cover analytical tasks in English; creativity and other languages (such as Portuguese) remain open.

10.3 Perspectives and agenda

Table 15

Horizon	Perspective	Evidence / anchoring
Short term	Multi-turn reliability benchmarks per effort level; cost reports normalized by tokenizer	Gap identified in [10] and in pricing docs
Short term	Product defaults that consolidate requirements and summarize state	Recommendations of [10]; techniques of [6]
Medium term	Deterministic policy and agent identity as a default in agentic platforms	[12], [13], [16]
Medium term	Automatically generated compliance evidence (signed logs, Evidence-as-Code)	AI Act Arts. 12 and 14 [14]; timeline of [15]
Long term	Models trained to reduce early assumptions and keep consistency in long conversations	Causes diagnosed in [10]

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 15 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Author's analytical inferences: (1) competitive advantage will probably shift from raw capability to **reliability per cost**; (2) cache-read price will tend to weigh more than input price in product comparisons; (3) runtime governance will tend to become a procurement requirement in regulated environments as the 2027 dates approach.

10.4 Section analysis and final synthesis

Table 16

Dimension	Analysis
Implications	Choosing between single-turn and multi-turn is an architecture decision, not only UX: it defines cost, reliability, risk and governance obligations.
Strengths	There are mature, documented techniques for each axis (cache, effort, state, policy, audit).
Weaknesses	Comparable public data is missing; the strongest results are from 2025; SGAEIA's evidence is internal.
Trade-offs	Capability × cost × reliability × risk: no configuration dominates; use single-turn for well-specified tasks and multi-turn with consolidation, typed state and deterministic authorization when exploration is needed.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 16 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

11. Conclusion

Moving from single-turn to multi-turn interaction is not merely an interface change. It turns history into operational state, expands cumulative cost, introduces dependencies between decisions, and creates distinct mechanisms of failure, persistence, and attack. Performance observed in an isolated call is therefore insufficient evidence of trajectory reliability.

The evidence reviewed here shows that multi-turn degradation depends on the task, protocol, presentation of context, memory mechanism, and the model's reliance on its own prior outputs. Caching, compaction, summarization, retrieval, and model switching can reduce cost or extend capability, but they also change provenance, authority, and reproducibility conditions.

For LLM routers and SGAEIA, the central consequence is that selection and governance must operate over explicit state and verifiable evidence. Constraints, permissions, revoked decisions, external effects, and remaining budget must survive model transitions and fallbacks. SGAEIA provides an architectural hypothesis for separating transcript, state, and assurance; its effectiveness should be evaluated empirically against reproducible baselines and threat scenarios.

Appendix A — Technical glossary

Table 17

Term	Definition
Aptitude (A^{90})	Best-case performance (90th percentile) of a model on an instruction, over repeated simulations [10].
Attention (self-attention)	Mechanism in which each position weighs all others; $O(n^2 \cdot d)$ cost per layer [1].
Blast radius	Extent of possible damage if an authorized action fails or is abused.
Prompt cache	Reuse of an already-processed prompt prefix, billed at a lower read price [20].
Chain of thought (CoT)	Sequence of reasoning steps generated before the final answer [19].
Checkpoint	Persistent snapshot of agent state; basis for resume, replay and fork [31].
Circuit breaker	Mechanism that interrupts the flow when a risk threshold is exceeded.
CONCAT	Condition where an instruction's shards are concatenated into one turn [10].
Context / context window	Maximum number of tokens the model processes at once.
Hallucination cascade	Propagation of a wrong premise accepted in one turn into later turns.
Drift	Progressive deviation: semantic (focus), behavioral (policy) or epistemic (accumulated error).
Reasoning effort	Parameter controlling how much the model "thinks" before answering (low to max).
Event sourcing	Persisting each transition as an immutable event, allowing state reconstruction.
Evidence-as-Code	Compliance evidence generated and verifiable as code (SGAEIA concept).
FSM / DAG	Finite state machine / directed acyclic graph, used to model agent flows.
Fork	Branch from a checkpoint with modified state [31].
FULL	Single-turn condition with a complete instruction [10].
Guardrail	Programmable layer that filters or restricts inputs and outputs.
HITL (human-in-the-loop)	Explicit human approval or supervision before a critical action.
Indirect prompt injection	Malicious instructions embedded in retrieved data that the model treats as commands [3].
Jailbreak	Technique to induce a model to violate its policies.
KV cache	Attention keys and values already computed and reused during generation.
Lost in the Middle	Performance drop when relevant information sits in the middle of the context [5].
MCP (Model Context Protocol)	Open standard to connect assistants to tools and data [21].
Multi-turn	Interaction as a sequence of messages, each answer conditioned on the history.
Unreliability (U)	Difference between the 90th and 10th percentiles of performance [10].
OPA / Rego	Policy-as-code engine and language.
Cedar	Open-source authorization language with formal analysis [12].
Prefill	Phase in which the model processes the input prompt before generating.
RAG	Retrieval-augmented generation: fetching documents to compose the context.
RLHF	Reinforcement learning from human feedback [2].
SHARDED	Multi-turn condition with one piece of information per turn [10].
Sliding window	Strategy that keeps only the N most recent interactions.
SNOWBALL / RECAP	Strategies that repeat requirements every turn or at the end [10].
Stateless / stateful	No state between calls / state kept.
Test-time compute	Compute spent at inference (for example, reasoning tokens) [19].
Token	Unit of text processed and billed by the model.
Reasoning tokens	Thinking tokens generated before the answer; billed as output.
Time-travel debugging	Replay and fork of executions from checkpoints.
Turn	One message from a role (user or assistant) in the conversation.
TTFT	Time to First Token: time until the first token of the response.
Unbounded Consumption	Risk of uncontrolled inference and cost consumption (OWASP LLM10) [16].
Zero Trust	Principle of trusting no entity or transition implicitly.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Table 17 from Article 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

References

Bibliographic verification policy. Each reference was checked for authorship, date, identifier and venue during the research stage (30 Oct. to 1 Nov. 2026). **Legend:** ☐ verified against a primary source; ☐ partly verified or via a secondary source; ○ awaiting confirmation against a primary source (supplied by the author or drawn from a preliminary record). ☐ and ○ items **must not be treated as validated**. Accessed: 1 Nov. 2026.

Scientific papers, standards and taxonomies

[1] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 30 (NeurIPS 2017), 2017, p. 5998–6008. arXiv:1706.03762, 12 Jun. 2017. DOI 10.48550/arXiv.1706.03762. □

[2] OUYANG, L. et al. Training language models to follow instructions with human feedback. arXiv:2203.02155, 4 Mar. 2022. □ (preprint; publication at NeurIPS 2022 **not confirmed**)

[3] GRESHAKE, K.; ABDELNABI, S.; MISHRA, S.; ENDRES, C.; HOLZ, T.; FRITZ, M. Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection. arXiv:2302.12173, 23 Feb. 2023 (v2, 5 May 2023). □ (poster at the ICML 2023 conversational-AI workshop; publication at AISec'23 **not confirmed**)

[4] ZHENG, L. et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 36 (NeurIPS 2023), p. 46595–46623. arXiv:2306.05685. □ (venue and pages checked via the reference list of [10])

[5] LIU, N. F.; LIN, K.; HEWITT, J.; PARANJAPE, A.; BEVILACQUA, M.; PETRONI, F.; LIANG, P. Lost in the middle: how language models use long contexts. *Transactions of the Association for Computational Linguistics*, v. 12, p. 157–173, 2024. DOI 10.1162/tacl_a_00638. arXiv:2307.03172 (6 Jul. 2023). □

[6] PACKER, C.; FANG, V.; PATIL, S. G.; LIN, K.; WOODERS, S.; GONZALEZ, J. E. MemGPT: towards LLMs as operating systems. arXiv:2310.08560, 12 Oct. 2023 (v2, 12 Feb. 2024). □

[7] BAI, G. et al. MT-Bench-101: a fine-grained benchmark for evaluating large language models in multi-turn dialogues. In: ANNUAL MEETING OF THE ACL, 62., 2024, Bangkok. *Proceedings*... v. 1, p. 7421–7454. DOI 10.18653/v1/2024.acl-long.401. □

[8] RUSSINOVICH, M.; SALEM, A.; ELDAN, R. Great, now write an article about that: the Crescendo multi-turn LLM jailbreak attack. In: USENIX SECURITY SYMPOSIUM, 34., 2025, Seattle. *Proceedings*... p. 2421–2440. arXiv:2404.01833. □

[9] BULLWINKEL, B. et al. A representation engineering perspective on the effectiveness of multi-turn jailbreaks. In: ICML 2025 WORKSHOP DIG-BUGS, 2025. Poster. □

[10] LABAN, P.; HAYASHI, H.; ZHOU, Y.; NEVILLE, J. LLMs get lost in multi-turn conversation. *International Conference on Learning Representations (ICLR)*, 2026. Outstanding Paper. Initial version: arXiv:2505.06120. □

[11] GUO, D. et al. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948, 2025. □ (identifier confirmed indirectly through the reference list of [10])

[12] CUTLER, J. W.; DISSELKOEN, C.; ELINE, A.; HE, S.; HEADLEY, K.; HICKS, M.; HIETALA, K.; IOANNIDIS, E.; KASTNER, J.; MAMAT, A.; MCADAMS, D.; MCCUTCHEN, M.; RUNGTA, N.; TORLAK, E.; WELLS, A. M. Cedar: a new language for expressive, fast, safe, and analyzable authorization. *Proceedings of the ACM on Programming Languages*, v. 8, n. OOPSLA1, 2024. DOI 10.1145/3649835. arXiv:2403.04651. □

[13] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. Gaithersburg: NIST, Jan. 2023. DOI 10.6028/NIST.AI.100-1. □ (identifier and date confirmed via copies of the document and secondary sources)

[14] EUROPEAN UNION. Regulation (EU) 2024/1689 (Artificial Intelligence Act). *Official Journal of the European Union*, L, 12.7.2024. □ (content of Arts. 12 and 14 checked against accessible copies of the text)

[15] EUROPEAN UNION. Regulation (EU) 2026/1744 ("Digital Omnibus on AI"). *Official Journal of the European Union*, L, 24.7.2026; in force 27.7.2026. □ (checked against seven consistent secondary sources)

[16] OWASP GENAI SECURITY PROJECT. *OWASP Top 10 for LLM Applications 2025* (v2.0). 18 Nov. 2024. CC BY-SA 4.0 license. Available at: https://genai.owasp.org/. □

[17] GOOGLE. Introducing Google's Secure AI Framework. Jun. 2023. Available at: https://blog.google/technology/safety-security/introducing-googles-secure-ai-framework/. □

Official provider documentation and announcements

[18] OPENAI. Introducing ChatGPT. 30 Nov. 2022. Available at: https://openai.com/index/chatgpt/. □

[19] OPENAI. Learning to reason with LLMs. 12 Sep. 2024. Available at: https://openai.com/index/learning-to-reason-with-llms/. □

[20] ANTHROPIC. Prompt caching with Claude. Aug. 2024 (generally available 17 Dec. 2024). Available at: https://www.anthropic.com/news/prompt-caching. □ (the claude.com page shows "14 Aug. 2025", at odds with 2024 coverage)

[21] ANTHROPIC. Introducing the Model Context Protocol. 25 Nov. 2024. Available at: https://www.anthropic.com/news/model-context-protocol. □

[22] GOOGLE. Our next-generation model: Gemini 1.5. 15 Feb. 2024. Available at: https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024; and What is a long context window?, at https://blog.google/technology/ai/long-context-window-ai-models/. □

[23] ANTHROPIC. Pricing. *Claude Platform Docs*. Available at: https://platform.claude.com/docs/en/about-claude/pricing. □ (Oct.–Nov. 2026 snapshot)

[24] ANTHROPIC. Steering thinking; Effort. *Claude Platform Docs*. Available at: https://platform.claude.com/docs/en/build-with-claude/thinking-steering-and-cost. □

[25] OPENAI. Models; Pricing; Reasoning models. *OpenAI API Docs*. Available at: https://developers.openai.com/api/docs/models. □

[26] GOOGLE. Gemini Developer API pricing. Available at: https://ai.google.dev/gemini-api/docs/pricing. □

[27] GOOGLE. Gemini thinking. Available at: https://ai.google.dev/gemini-api/docs/thinking. □

[28] XAI. Models; Pricing; Reasoning. *xAI Docs*. Available at: https://docs.x.ai/developers/models. □

[29] DEEPSEEK. Models & Pricing. *DeepSeek API Docs*. Available at: https://api-docs.deepseek.com/quick_start/pricing/. □

[30] ANTHROPIC. Using the Messages API. *Claude Platform Docs*. Available at: https://platform.claude.com/docs/en/build-with-claude/working-with-messages. □

[31] LANGCHAIN. LangGraph: persistence; use time-travel. Available at: https://docs.langchain.com/oss/python/langgraph/persistence and https://docs.langchain.com/oss/python/langgraph/use-time-travel. □

[32] PYDANTIC. Migration guide (Pydantic V2). Available at: https://docs.pydantic.dev/. □

[33] ANTHROPIC. Claude está fornecendo respostas incorretas ou enganosas. O que está acontecendo? (Claude is giving incorrect or misleading answers. What is happening?) *Anthropic Help Center*, 5 Mar. 2026. Available at: https://support.claude.com/pt/articles/8525154. □

Works by the author

[34] SILVA, A. *SGAEIA: Secure Governed Autonomous Edge Intelligence Architecture* (Version 0.3.4). Zenodo, 2026. DOI 10.5281/zenodo.22557796. Public frozen artifact under Apache-2.0; ongoing development maintained in a controlled private repository. □

[35] SILVA, A. Articles on DEV Community (https://dev.to/aridiosilva): *From Edge AI to Governed Autonomous Edge Intelligence* (1 Oct. 2026); *The Open Letter on AI and Cybersecurity and the Evidence for Agent Containment* (16 Oct. 2026); *Opening the Black Box of LLM Inference: Foundations of Generative AI* (21 Oct. 2026). Related reading: *How to Reduce the Cost of AI Systems with LLMs Without Sacrificing Quality* (28 Oct. 2026) and *LLM Routers for AI Systems: Architecture, Governance, Multi-Agent Systems, Swarms, and Security* (29 Oct. 2026). □ (existence, titles and dates confirmed on the public profile)

[36] SILVA, A. *Technical compendium: in-depth analysis of LLMs, state management, runtime governance and the SGAEIA framework*. .docx document, 1 Nov. 2026 (source material incorporated into Sections 2, 7 and 8).

Complementary sources

[37] P. Laban, H. Hayashi, Y. Zhou, and J. Neville, "LLMs Get Lost in Multi-Turn Conversation," *International Conference on Learning Representations (ICLR)*, 2026. Available: https://proceedings.iclr.cc/paper_files/paper/2026/hash/59f6421e64707225fdf5b28840679a07-Abstract-Conference.html

[38] ICLR, "Announcing the ICLR 2026 Outstanding Papers," 23 Apr. 2026. Available: https://blog.iclr.cc/2026/04/23/announcing-the-iclr-2026-outstanding-papers/

[39] W.-C. Kwan et al., "MT-Eval: A Multi-Turn Capabilities Evaluation Benchmark for Large Language Models," *Proceedings of EMNLP*, pp. 20153–20177, 2024. Available: https://aclanthology.org/2024.emnlp-main.1124/

[40] K. Deshpande et al., "MultiChallenge: A Realistic Multi-Turn Conversation Evaluation Benchmark Challenging to Frontier LLMs," *Findings of ACL*, pp. 18632–18702, 2025. doi:10.18653/v1/2025.findings-acl.958.

[41] "StructFlowBench: A Structured Flow Benchmark for Multi-turn Instruction Following," *Findings of ACL*, pp. 9322–9341, 2025. doi:10.18653/v1/2025.findings-acl.486.

[42] Y. Katsis et al., "mt RAG: A Multi-Turn Conversational Benchmark for Evaluating Retrieval-Augmented Generation Systems," *Transactions of the Association for Computational Linguistics*, 2025. Available: https://aclanthology.org/2025.tacl-1.36/

[43] C. Yang et al., "MARS-Bench: A Multi-turn Athletic Real-world Scenario Benchmark for Dialogue Evaluation," *Findings of EMNLP*, pp. 5872–5898, 2025. doi:10.18653/v1/2025.findings-emnlp.314.

[44] "One Battle After Another: Probing LLMs' Limits on Multi-Turn Instruction Following with a Benchmark Evolving Framework," *Proceedings of ACL*, 2026. Available: https://aclanthology.org/2026.acl-long.433/

[45] J. Chen et al., "KnowMT-Bench: Benchmarking Knowledge-Intensive Long-Form Question Answering in Multi-Turn Dialogues," *Findings of EMNLP*, 2026. Available: https://aclanthology.org/2026.findings-emnlp.845/

[46] J. Myung, "Quantifying Conversational Reliability of Large Language Models under Multi-Turn Interaction," arXiv:2603.01423, 2026. **Preprint.**

[47] R. Khraishi, I. Zafar, K. Myles, and G. A. Cowan, "Evaluating Performance Drift from Model Switching in Multi-Turn LLM Systems," arXiv:2603.03111, 2026. **Preprint.**

[48] S. T. Luo, W. Liu, R. Liu, M. Gong, and J. Guo, "Multi-Turn Reasoning When Context Arrives in Pieces: Scalable Sharding and Memory-Augmented RL," arXiv:2606.12941, 2026. **Preprint.**

[49] A. Silva, "LLM Routers for AI Systems: Architecture, Governance, Multi-Agent Systems, Swarms, and Security," *SGAEIA Research Series*, Article 24, 2026. DOI: 10.5281/zenodo.23763978. Available: https://aridiosilva.com/publications/artigo24/

Related SGAEIA Research Series works / Trabalhos relacionados da série SGAEIA

[50] A. Silva, "From Model Capability to Governed Action: An Architecture for Secure Agentic AI," *SGAEIA Research Series*, Article 11, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[51] A. Silva, "Zero Trust for Multi-Agent AI Systems," *SGAEIA Research Series*, Article 3, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[52] A. Silva, "Evidence-as-Code: Making AI Governance Verifiable," *SGAEIA Research Series*, Article 6, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[53] A. Silva, "AI Jailbreaks: From Bypassing Restrictions to Governing Authority," *SGAEIA Research Series*, Article 17, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[54] A. Silva, "Tracing AI Agent Execution: Foundations, State of the Art, and Implications for Distributed Multi-Agent Systems," *SGAEIA Research Series*, Article 18, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[55] A. Silva, "When AI Agents Form Societies: Is Security a Property of the Model or the System?," *SGAEIA Research Series*, Article 15, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[56] A. Silva, "How to Reduce the Cost of AI Systems with LLMs Without Sacrificing Quality," *SGAEIA Research Series*, Article 23, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

About the Author

Aridio Silva is an independent researcher based in Brazil working on the architecture, security, governance, and trustworthiness of autonomous and distributed artificial intelligence systems.

His research focuses on Agentic AI, Multi-Agent Systems, Edge AI, AI Security, Zero Trust, Security-by-Design, AI Governance, Spec-Driven Development, and continuous security assurance.

He is the creator and lead researcher of SGAEIA — Secure Governed Autonomous Edge Intelligence Architecture, an open research initiative investigating architectural foundations for secure, governed, auditable, and trustworthy autonomous AI systems operating across distributed edge-cloud environments.

Research Profiles

- ORCID: https://orcid.org/0009-0008-2411-6995
- Google Scholar: https://scholar.google.com/citations?user=Pn5O48AAAAAJ
- Zenodo — SGAEIA Community: https://zenodo.org/communities/sgaeia
- OpenAIRE: https://explore.openaire.eu/search/find?fv0=Aridio%20Silva&f0=q
- Medium: https://medium.com/@aridiosilva
- LinkedIn: https://www.linkedin.com/in/aridio-silva-74997111/
- SGAEIA Research Artifact: https://doi.org/10.5281/zenodo.22557796
- Homepage: https://aridiosilva.com
- SGAEIA: https://aridiosilva.com/sgaeia.html
- DEV Community: https://dev.to/aridiosilva
- Academia.edu: https://independent.academia.edu/AridioSilva

Figures

The cover is unnumbered. Figures, equations, and tables are numbered by type, cited in the text, and supplied as high-resolution JPEG files. Original conceptual assets remain licensed under CC BY 4.0 and carry visible and embedded authorship and copyright information.

License

Except where otherwise noted, the text and original conceptual illustrations in this article are licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

© Aridio Silva (@aridiosilva) · November 2026. The SGAEIA software and research artifact remains subject to its own Apache License 2.0.

Series Continuity

This work is **SGAEIA Research Series — Article 25**. The public series catalogue is available at https://aridiosilva.com, with the project presence at https://aridiosilva.com/sgaeia.html.

Persistent record for Article 25: https://doi.org/10.5281/zenodo.23867630. Canonical SGAEIA project artifact: https://doi.org/10.5281/zenodo.22557796.

1 From Edge AI to Governed Autonomous Edge Intelligence

2 Why Autonomous AI Agents Need Bounded and Revocable Authority

3 Zero Trust for Multi-Agent AI Systems

4 Authenticated Delegation Between Autonomous AI Agents

5 Continuous GRC for Agentic AI

6 Evidence-as-Code: Making AI Governance Verifiable

7 Security-by-Design for Distributed Agentic AI

8 SGAEIA: A Reference Architecture for Secure Governed Autonomous Edge Intelligence

9 When AI Begins to Build AI

10 Rogue AI Agents: When Autonomous Systems Become a Collective

11 From Model Capability to Governed Action: An Architecture for Secure Agentic AI

12 Ten Days That Exposed the Governance Gap in Frontier AI

13 When AI Hallucination Becomes Action

14 AI's Alien Mind: How to Govern What We Cannot Fully Understand

15 When AI Agents Form Societies: Is Security a Property of the Model or the System?

16 When AI Improves AI: Governing Recursive Intelligence Acceleration

17 AI Jailbreaks: From Bypassing Restrictions to Governing Authority

18 Tracing AI Agent Execution: Foundations, State of the Art, and Implications for Distributed Multi-Agent Systems

19 LoRA in AI Agents: Efficiency, Security Risks, and Adapter Governance

20 AI Agents and Agentic AI: What Sets Them Apart?

21 The Open Letter on AI and Cybersecurity and the Evidence for Agent Containment

22 Opening the Black Box of LLM Inference: Foundations of Generative AI

23 How to Reduce the Cost of AI Systems with LLMs Without Sacrificing Quality

24 LLM Routers for AI Systems: Architecture, Governance, Multi-Agent Systems, Swarms, and Security

25 Single-Turn and Multi-Turn Interactions in Frontier Generative AI Models (LLMs)

Suggested Citation

SILVA, Aridio. *Single-Turn and Multi-Turn Interactions in Frontier Generative AI Models (LLMs): State, Cost, Reliability, Security, and Governance*. SGAEIA Research Series, Article 25, 2026. DOI: https://doi.org/10.5281/zenodo.23867630.