Cognito: A Formally Verifiable, Metacognitively-Governed Architecture for Trustworthy Agentic AI

Authors: Rafael Oliveira (ORCID: 0009-0005-2697-4668)

Abstract
The rapid shift from generative to agentic Artificial Intelligence (AI) has introduced a new class of systemic risks, creating a crisis of trust and auditability that current safety paradigms are ill-equipped to handle. This paper introduces Cognito, a novel cognitive architecture for Artificial General Intelligence (AGI) designed from first principles for verifiability, safety, and accountability. Cognito operationalizes a dual-process theory of mind through a neuro-symbolic framework, governed by an explicit Metacognitive Controller (CMC) that manages a fast, intuitive sub-symbolic substrate and a slow, deliberative symbolic substrate.1 We address key criticisms of conceptual AGI frameworks by detailing: (1) a formal runtime loop and signal flow that unifies these components; (2) a novel paradigm of **Identity-Typed Programming (ITP)**, which encodes an agent's identity and permissions into its static type signature using Quantitative Type Theory (QTT), enabling compile-time verification of authorization logic [5]; (3) a hybrid verification model that combines the static guarantees of ITP with a dynamic runtime enforcement layer for contextual safety [7]; and (4) a

Computational Monad Model (CMM) for scaling, which includes a cryptographically verifiable protocol for fusing multiple specialized agents into a coherent, emergent super-agent. By integrating formal methods, metacognitive governance, and a robust identity framework, Cognito provides a technical blueprint for AGI systems that are not only powerful but also provably safe, auditable, and aligned by construction.

1. Introduction: The Crisis of Trust in Agentic AI

The current era of AI is defined by a paradigm shift from generative models that produce content to agentic systems that create outcomes in the world.[9] This transition to autonomy, while powerful, has exposed fundamental weaknesses in existing safety approaches. Large Language Model (LLM) based agents are "doomed to complete" maliciously crafted requests, and their safety "guardrails" are often just "soft boundaries" easily circumvented.[9] The resulting risks—uncontrolled autonomy, fragmented system access, and a lack of traceability—create a crisis of trust and auditability.[11] This challenge has motivated significant investment from bodies like the Defense Advanced Research Projects Agency (DARPA), whose Assured Neuro Symbolic Learning and Reasoning (ANSR) and Artificial Intelligence Quantified (AIQ) programs seek to create AI systems with guaranteed performance and robustness through the integration of symbolic reasoning and data-driven learning.[14]

Current approaches to AI safety, which often rely on post-hoc alignment or runtime monitoring, are insufficient. A more fundamental solution is required: an architecture where safety, identity, and accountability are intrinsic properties. This paper introduces Cognito, an architecture that addresses this need by operationalizing a metacognitively-governed, neuro-symbolic framework within the rigorous mathematical container of a dependently typed language. Our central thesis is that trustworthy AGI can only be achieved through a hybrid verification model that combines the *a priori* guarantees of static, type-level proofs with the contextual awareness of dynamic, runtime enforcement.

2. Related Work

Cognito builds upon three pillars of AI research: cognitive architectures, neuro-symbolic systems, and computational metacognition.

Cognitive Architectures: Frameworks like ACT-R and Soar provide blueprints for intelligence, specifying fixed structures for memory and processing.[23] ACT-R's modular design, with its declarative and procedural memory systems, informs our separation of concerns.[25] Soar's focus on problem-solving in a unified framework highlights the need for coherent goal-directed behavior.[14] More recently, the CoALA framework has attempted to bring this structural thinking to modern language agents, but it remains a descriptive blueprint rather than a prescriptive, verifiable engineering model. Cognito moves beyond these by making the architectural constraints formally verifiable.

Neuro-Symbolic (NeSy) AI: NeSy seeks to combine the strengths of neural networks (learning) and symbolic systems (reasoning). Taxonomies like Kautz's categorize integration patterns, such as pipelines (Neuro|Symbolic) or systems where logic constrains neural training (Neuro_{Symbolic}).[2] Cognito implements a highly integrated, bidirectional loop that aligns with the "learning-reasoning" model, but operationalizes it with specific, verifiable mechanisms.[45]

Computational Metacognition: Recent research has identified a "metacognitive void" in AI, with only 5% of NeSy papers addressing the topic.[46] Frameworks like SOFAI-LM and Meta-R1 propose architectures with explicit metacognitive modules that monitor and regulate a primary reasoning agent, inspired by dual-process theories.[52] These systems demonstrate the value of self-monitoring for confidence and error correction. Cognito adopts this principle but makes the metacognitive controller's logic and its governance over the other components formally verifiable.

3. The Cognito Architecture: A Technical Deep Dive

Cognito is a metacognitively-governed, neuro-symbolic architecture designed for verifiability. It operationalizes dual-process theory by instantiating System 1 and System 2 as distinct computational substrates, governed by a third component, the Metacognitive Controller (CMC).

3.1. Core Components

1. **Sub-Symbolic Substrate (S1):** This is the fast, intuitive pattern-recognition engine. To handle the long-range dependencies required for robust intuition with high efficiency, this substrate is implemented using **Structured State Space Models (SSMs)** like Mamba, which offer near-linear complexity and faster inference compared to Transformers.[54] Its function is to process raw perceptual data and generate probabilistic hypotheses (e.g.,
{'concept': 'cat', 'confidence': 0.98}).
2. **Symbolic Substrate (S2):** This is the slow, deliberative reasoning engine. It is built around a **Neuro-Symbolic Program Synthesizer** that, given a specification, generates an executable program in a formal Domain-Specific Language (DSL). The output is not an opaque neural state but an auditable, verifiable program.
3. **Bidirectional Translation Bus:** This is the critical interface that addresses the semantic translation problem. It operates in two directions:
    - **Neural → Symbolic:** It translates the probabilistic output of S1 into a symbolic representation. This is not a direct mapping. It uses frameworks like **Logic Tensor Networks (LTNs)** to represent the neural output as a fuzzy logical assertion, preserving uncertainty.[57] An S1 output
{'concept': 'cat', 'confidence': 0.98} becomes a fuzzy logical term Cat(x) with a truth value of 0.98.
    - **Symbolic → Neural:** It translates feedback from the symbolic system (e.g., a proof failure or a logical constraint) into a differentiable loss signal that can guide the S1

substrate, a mechanism we term **Proof-Guided Learning (PGL)**.

4. **Metacognitive Controller (CMC):** This is the "conscious core" or "inner voice" of the system, responsible for functional self-awareness and governance. It is not a metaphor but an operationalized control loop. Its key functions are:
   - **Confidence Monitoring:** It continuously assesses the confidence scores from S1. If confidence falls below a threshold τ_c, it triggers a deeper verification by S2.
   - **Logical Consistency Auditing:** It maintains a knowledge base of core axioms. Using LTNs, it continuously calculates the satisfaction degree of the system's current belief state against these axioms. If the satisfaction score falls below a threshold τ_s, it signals a logical contradiction.[58]
   - **Strategic Triage:** It routes incoming tasks based on a risk assessment. Low-risk, perceptual tasks are handled by S1. High-risk or logic-heavy tasks are sent directly to S2. Ambiguous tasks follow a hybrid path.

3.2. The Runtime Loop: Operationalizing the "Inner Voice"

The interaction between these components is governed by a formal, recursive runtime loop, which constitutes the agent's "thought process."

Algorithm 1: Cognito Metacognitive Runtime Loop

```
function handle_task(task):
 // 1. GENERATE (S1 - Fast Intuition)
 hypothesis, confidence = S1.propose(task)

 // 2. METACOGNITIVE AUDIT (CMC)
 if confidence < τ_c OR task.is_high_risk():
   // Escalate to slow, deliberative reasoning
   return verify_and_execute(task, hypothesis)
 else:
   // Fast path for high-confidence, low-risk tasks
   return execute(hypothesis)

function verify_and_execute(task, hypothesis):
 // 3. VERIFY (S2 - Slow Reasoning)
 program = S2.synthesize_program(hypothesis)
```

```
 verification_result = S2.verify(program)

 if verification_result.is_success():
   // 4a. EXECUTE
   return execute(program)
 else:
   // 4b. CORRECT (Recursive Self-Correction)
   error_feedback = TranslationBus.proof_failure_to_signal(verification_result.error)
   S1.update_with_feedback(error_feedback) // Proof-Guided Learning
   return handle_task(task) // Recursive call with new constraints
```

This loop provides the formal signal flow and recursive self-correction mechanism that addresses key architectural critiques. A verification failure is not an error; it is a productive signal that drives learning and adaptation.

4. Identity-Typed Programming: A Foundation for Verifiable Agents

To provide mathematical guarantees about an agent's behavior, particularly its authorization logic, we introduce **Identity-Typed Programming (ITP)**. ITP elevates an agent's identity and permissions from runtime data to static, compile-time properties.

4.1. From Runtime Checks to Compile-Time Proofs

ITP is built on **Quantitative Type Theory (QTT)**, the foundation of languages like Idris 2.[5] QTT allows types to depend on values and tracks the usage of variables with multiplicities (e.g., a linear type with multiplicity

1 must be used exactly once). In ITP, we model permissions as linear resources. An action that requires a permission "consumes" the type-level proof of that permission, ensuring it cannot be used again if it is single-use.

4.2. Identity Models and Type-Level Claims

An agent's identity is established via cryptographic credentials, which can be centralized (e.g., a JWT from AWS Cognito) or decentralized (W3C Verifiable Credentials with DIDs).[64] Upon initialization, the agent validates these credentials and ingests the claims into a type-level construct, the

VerifiedContext.

Snippet de código

```
-- Psy Language Example
namespace Auth

 -- A type representing a cryptographically verified claim
 data Claim : (key : String) -> (val : String) -> Type where
   MkClaim : (1 _ : VerifiedProof) -> Claim key val

 -- A function requiring a specific permission, enforced by the type checker
 deleteFile : (1 ctx : VerifiedContext) -> (path : String) ->
              {auto prf : HasClaim "scope" "files.delete" ctx} ->
              IO ()
```

The deleteFile function requires an implicit proof (prf) that the claim "scope": "files.delete" exists in the context. If this proof cannot be constructed at compile-time, the program is rejected. This statically prevents the agent from even being compiled with code that attempts an action for which it lacks the necessary identity-based permissions.

5. Hybrid Verification: The Synthesis of Static and Runtime Guarantees

While ITP provides powerful static guarantees, it cannot reason about dynamic, context-dependent properties. To address this, Cognito employs a hybrid verification model that integrates a runtime enforcement layer.

5.1. Runtime Enforcement with AgentSpec

We adopt a framework inspired by **AgentSpec**, a DSL for specifying and enforcing runtime constraints on LLM agents.[7] The CMC integrates an AgentSpec-like monitor that intercepts actions

after they have been statically verified but *before* they are executed. This monitor checks the action against dynamic constraints (e.g., "Is the trade volume for this stock within today's risk limits?").

5.2. The Integrated Verification Loop

The two verification layers work in concert, governed by the CMC:

1. **Static Verification (Compile-Time):** ITP ensures the agent's code is compliant with its static identity and permissions. An agent without the files.delete permission can never have a valid deleteFile function in its compiled binary.
2. **Dynamic Verification (Runtime):** Before executing a statically-valid action, the AgentSpec monitor checks it against dynamic rules. The deleteFile action might be blocked if the file is currently locked by another process.
3. **Metacognitive Feedback:** A runtime violation is fed back to the CMC, which can trigger the self-correction loop. This might involve synthesizing a new plan that respects the dynamic constraint or escalating to a human operator.

This hybrid model provides defense-in-depth, combining the absolute guarantees of formal methods with the contextual flexibility of runtime monitoring.

6. The Computational Monad Model: Scaling Cognition and Memory

To address scalability and the "Monad" metaphor, we introduce the **Computational Monad Model (CMM)**, a framework for composing verified agents.

6.1. From Single Agent to Agent Families

In the CMM, a single Cognito agent is a "Monad." A "Soul Family" is a decentralized network of specialized agents (e.g., a perception agent, a logic agent, an ethics agent) operating on a distributed runtime like **Parallax**.[72]

6.2. The Monadic Fusion Protocol

The CMM's core innovation is a verifiable protocol for fusing multiple agents into a temporary "Super-Agent" to solve complex problems.

1. **Knowledge Fusion:** The agents' symbolic knowledge bases are merged. The type system verifies the logical consistency of the combined knowledge.
2. **Model Fusion:** The neural models are combined using techniques like Mixture-of-Experts (MoE), where the Super-Agent's CMC learns to route sub-tasks to the most appropriate neural "fragment."
3. **Verifiable Fusion with Zero-Knowledge Proofs (ZKPs):** The integrity of the fusion process is guaranteed by ZKPs. An agent can generate a zk-SNARK to prove it has correctly incorporated another agent's knowledge without revealing the underlying (potentially proprietary) data.[82] This makes the fusion process itself an auditable, trustless computation.

6.3. Memory Compression as Principled Generalization

The CMM also addresses the challenge of memory compression. The agent's long-term memory is not an ever-expanding log of raw experiences. Instead, the ethical and cognitive evolution process we have described acts as a form of principled, lossy compression. The neuro-symbolic loop analyzes specific experiences ("case law") to abstract them into more general rules ("statutes"), which are then periodically reviewed by human governance to refine core principles ("constitution"). This hierarchical process distills vast amounts of experience into a compact, robust, and generalizable knowledge structure.

7. Discussion and Future Work

The Cognito architecture presents a comprehensive but challenging roadmap.

- **Computational Overhead:** The primary challenge is the overhead of formal verification and ZKP generation. We propose mitigating this through asynchronous audits, risk-based verification, and leveraging the heterogeneous, distributed nature of the Parallax runtime to parallelize symbolic and neural workloads.[72]
- **Specification Complexity:** The "proof is only as good as the spec" problem remains.[101] Formally specifying complex ethical and cultural norms is a profound socio-technical challenge that requires participatory governance models.
- **Consciousness:** Cognito aims for functional, auditable self-awareness, not phenomenal consciousness. The "hard problem" remains a philosophical and scientific frontier.

Future work will focus on implementing a prototype of the ITP paradigm in a language like Idris 2, developing the ZKP-based fusion protocol, and refining the metrics for evaluating metacognitive governance in complex, multi-agent scenarios.

8. Conclusion

The emergence of agentic AI necessitates a paradigm shift from post-hoc alignment to verifiable, by-design safety. The Cognito architecture provides a technical blueprint for this shift. By operationalizing a metacognitively-governed neuro-symbolic loop within a framework of Identity-Typed Programming and hybrid verification, it addresses the core challenges of safety, auditability, and identity that plague current systems. The Computational Monad Model further provides a scalable path toward emergent, collective intelligence. While the engineering and philosophical challenges are immense, we argue that this principled, formally-grounded approach is essential for building an AGI that society can trust.

Referências citadas

1. Unlocking the Potential of Generative AI through Neuro-Symbolic Architectures – Benefits and Limitations - arXiv, acessado em outubro 2, 2025, https://arxiv.org/html/2502.11269v1
2. Neuro-symbolic AI - Wikipedia, acessado em outubro 2, 2025, https://en.wikipedia.org/wiki/Neuro-symbolic_AI
3. Neuro-Symbolic AI: Let's go back to the start | by Eric Papenhausen | Medium, acessado em outubro 2, 2025, https://medium.com/@epapenha_40736/neuro-symbolic-ai-lets-go-back-to-the-

start-cca9be15002
4. Language Models Coupled with Metacognition Can Outperform Reasoning Models - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2508.17959
5. Idris 2: Quantitative Type Theory in Practice - DROPS, acessado em outubro 2, 2025, https://drops.dagstuhl.de/storage/00lipics/lipics-vol194-ecoop2021/LIPIcs.ECOOP.2021.9/LIPIcs.ECOOP.2021.9.pdf
6. Idris 2: Quantitative Type Theory in practice - University of St Andrews Research Portal, acessado em outubro 2, 2025, https://research-portal.st-andrews.ac.uk/en/publications/07daeee4-0629-4afb-aadb-7da20f1ba416
7. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2503.18666v1.pdf?ref=applied-gai-in-security.ghost.io
8. AgentSpec: Customizable Runtime Enforcement for Safe and ... - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2503.18666
9. AI Security Trends from Black Hat 2025 | ActiveFence, acessado em outubro 2, 2025, https://www.activefence.com/blog/black-hat-2025-ai-security-takeaways/
10. Blackhat and Def Con 2025 Thoughts - IDPro, acessado em outubro 2, 2025, https://idpro.org/blackhat-and-def-con-2025-thoughts/
11. Seizing the agentic AI advantage - McKinsey, acessado em outubro 2, 2025, https://www.mckinsey.com/capabilities/quantumblack/our-insights/seizing-the-agentic-ai-advantage
12. Industry News 2025 The Growing Challenge of Auditing Agentic AI - ISACA, acessado em outubro 2, 2025, https://www.isaca.org/resources/news-and-trends/industry-news/2025/the-growing-challenge-of-auditing-agentic-ai
13. AI trends 2025: Adoption barriers and updated predictions - Deloitte, acessado em outubro 2, 2025, https://www.deloitte.com/us/en/services/consulting/blogs/ai-adoption-challenges-ai-trends.html
14. Soar (cognitive architecture) - Wikipedia, acessado em outubro 2, 2025, https://en.wikipedia.org/wiki/Soar_(cognitive_architecture)
15. Gradient aims to bring decentralization to AI - CryptoRank, acessado em outubro 2, 2025, https://cryptorank.io/news/feed/bda12-gradient-network-10m-ai-infrastructure
16. Information Innovation Office - DARPA, acessado em outubro 2, 2025, https://www.darpa.mil/about/offices/i2o
17. Programs - DARPA, acessado em outubro 2, 2025, https://www.darpa.mil/research/programs
18. AI Forward | DARPA, acessado em outubro 2, 2025, https://www.darpa.mil/research/programs/ai-forward
19. ANSR: Assured Neuro Symbolic Learning and Reasoning - DARPA, acessado em outubro 2, 2025, https://www.darpa.mil/research/programs/assured-neuro-symbolic-learning-and-

reasoning
20. Assured Neuro Symbolic Learning and Reasoning (DARPA) [SRI funding: 5.4M], acessado em outubro 2, 2025, https://nusci.csl.sri.com/project/darpa-ansr/
21. U.S. DOD DARPA - Artificial Intelligence Quantified (AIQ) - Memento EPFL, acessado em outubro 2, 2025, https://memento.epfl.ch/event/us-dod-darpa-artificial-intelligence-quantified-ai/
22. ANSRs to Hard AI Questions - DARPA, acessado em outubro 2, 2025, https://www.darpa.mil/news/2023/ansrs-ai-questions
23. Cognitive architecture - Wikipedia, acessado em outubro 2, 2025, https://en.wikipedia.org/wiki/Cognitive_architecture
24. ACT-R - Wikipedia, acessado em outubro 2, 2025, https://en.wikipedia.org/wiki/ACT-R
25. An overview of ACT- R's modules and their dependent buffers. - ResearchGate, acessado em outubro 2, 2025, https://www.researchgate.net/figure/An-overview-of-ACT-Rs-modules-and-their-dependent-buffers_fig1_268520256
26. Cognitive architectures: Research issues and challenges - GitHub, acessado em outubro 2, 2025, https://raw.githubusercontent.com/SoarGroup/website-downloads/main/pubs/cogarch.cogsys08.pdf
27. Cognitive Architectures: Research Issues and Challenges - Vanderbilt University, acessado em outubro 2, 2025, https://cdn-dev.vanderbilt.edu/t2-my-dev/wp-content/uploads/sites/3191/2021/02/CognitiveArchitecture-copy.pdf
28. 1562 PDFs | Review articles in ACT-R - ResearchGate, acessado em outubro 2, 2025, https://www.researchgate.net/topic/ACT-R/publications
29. How does the ACT-R model compare to other cognitive architectures in decision-making?, acessado em outubro 2, 2025, https://consensus.app/search/how-does-the-act-r-model-compare-to-other-cognitiv/8KOMbEz5QWWsremiqA1tvA/
30. (PDF) An Analysis and Comparison of ACT-R and Soar, acessado em outubro 2, 2025, https://www.researchgate.net/publication/358148660_An_Analysis_and_Comparison_of_ACT-R_and_Soar
31. Comparing LLMs for Prompt-Enhanced ACT-R and Soar Model Development: A Case Study in Cognitive Simulation - Penn State, acessado em outubro 2, 2025, https://acs.ist.psu.edu/papers/wuSRLip.pdf
32. A Survey of Cognitive Architectures - CMU School of Computer Science, acessado em outubro 2, 2025, https://www.cs.cmu.edu/~bej/CognitiveModelingForUIDesign/2b_Architecture_Survey.pdf
33. ACT-R: A cognitive architecture for modeling cognition - PubMed, acessado em outubro 2, 2025, https://pubmed.ncbi.nlm.nih.gov/30536740/
34. Integrated cognitive architectures: A survey - InK@SMU.edu.sg, acessado em outubro 2, 2025,

https://ink.library.smu.edu.sg/context/sis_research/article/6223/viewcontent/Integrated_cognitive_architectures_A_survey.pdf
35. About - ACT-R - Carnegie Mellon University, acessado em outubro 2, 2025, http://act-r.psy.cmu.edu/about/
36. Introduction to the Soar Cognitive Architecture, acessado em outubro 2, 2025, https://acs.ist.psu.edu/misc/nottingham/pst.w.9/hungry-thirsty/full.html
37. Introduction to the Soar Cognitive Architecture, acessado em outubro 2, 2025, https://acs.ist.psu.edu/misc/nottingham/pst2/hungry-thirsty/full.html
38. an operator transforms a state (makes changes to the representation); and a goal is a desired outcome of the problem-solving activity. - Soar Cognitive Architecture, acessado em outubro 2, 2025, https://soar.eecs.umich.edu/soar_manual/02_TheSoarArchitecture/
39. (PDF) An introduction to Soar as an agent architecture - ResearchGate, acessado em outubro 2, 2025, https://www.researchgate.net/publication/228398068_An_introduction_to_Soar_as_an_agent_architecture
40. Introduction to the Soar Cognitive Architecture1 - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2205.03854
41. Neuro Symbolic Taxonomies - Lab groups within ASU Engineering, acessado em outubro 2, 2025, https://labs.engineering.asu.edu/labv2/wp-content/uploads/sites/82/2022/09/NSRtaxonomies.pdf
42. Evaluating Neuro-Symbolic AI Architectures: Design Principles, Qualitative Benchmark, Comparative Analysis and Results - OpenReview, acessado em outubro 2, 2025, https://openreview.net/pdf?id=yCwcRijfXz
43. Towards Cognitive AI Systems: a Survey and Prospective on Neuro-Symbolic AI - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2401.01040
44. Towards Cognitive AI Systems: Workload and Characterization of Neuro-Symbolic AI - Zishen Wan, acessado em outubro 2, 2025, https://zishenwan.github.io/publication/ISPASS24_NSAI.pdf
45. Neuro-symbolic artificial intelligence | European Data Protection Supervisor, acessado em outubro 2, 2025, https://www.edps.europa.eu/data-protection/technology-monitoring/techsonar/neuro-symbolic-artificial-intelligence_en
46. Neuro-Symbolic AI in 2024: A Systematic Review - arXiv, acessado em outubro 2, 2025, https://arxiv.org/html/2501.05435v1
47. [2501.05435] Neuro-Symbolic AI in 2024: A Systematic Review - arXiv, acessado em outubro 2, 2025, https://arxiv.org/abs/2501.05435
48. Neuro-Symbolic AI in 2024: A Systematic Review - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2501.05435
49. Neuro-Symbolic AI in 2024: A Systematic Review - CEUR-WS, acessado em outubro 2, 2025, https://ceur-ws.org/Vol-3819/paper3.pdf
50. [Literature Review] Neuro-Symbolic AI in 2024: A Systematic Review - Moonlight, acessado em outubro 2, 2025, https://www.themoonlight.io/en/review/neuro-symbolic-ai-in-2024-a-systematic-

review
51. acessado em dezembro 31, 1969, https.arxiv.org/abs/2501.05435
52. Language Models Coupled with Metacognition Can Outperform Reasoning Models - arXiv, acessado em outubro 2, 2025, https://arxiv.org/html/2508.17959v1
53. Meta-R1: Empowering Large Reasoning Models with Metacognition - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2508.17291
54. New AI Holographic Cognitive Fields vs Traditional LLM Process - Meaning Spark Labs, acessado em outubro 2, 2025, https://www.meaningspark.com/blog/ai-holographic-cognitive-fields-vs-traditional-llm-process
55. From S4 to Mamba: A Comprehensive Survey on Structured State Space Models - arXiv, acessado em outubro 2, 2025, https://www.arxiv.org/pdf/2503.18970
56. From S4 to Mamba: A Comprehensive Survey on Structured State Space Models - arXiv, acessado em outubro 2, 2025, https://arxiv.org/abs/2503.18970
57. Neurosymbolic AI: Bridging Neural Networks and Symbolic Reasoning for Smarter Systems, acessado em outubro 2, 2025, https://www.netguru.com/blog/neurosymbolic-ai
58. Logic Tensor Networks for Semantic Image Interpretation - IJCAI, acessado em outubro 2, 2025, https://www.ijcai.org/proceedings/2017/0221.pdf
59. Logic Tensor Networks - Sony AI, acessado em outubro 2, 2025, https://ai.sony/publications/Logic-Tensor-Networks/
60. Logic tensor network-enhanced generative adversarial network - University of Edinburgh Research Explorer, acessado em outubro 2, 2025, https://www.research.ed.ac.uk/files/539098517/UpretiandBelleICLP2025LogicTensorNetwork-Enhanced.pdf
61. [PDF] Logic Tensor Networks - Semantic Scholar, acessado em outubro 2, 2025, https://www.semanticscholar.org/paper/Logic-Tensor-Networks-Badreddine-Garcez/76db1e8479ae91590bb6c9ef21e347775771978b
62. [PDF] Idris 2: Quantitative Type Theory in Practice | Semantic Scholar, acessado em outubro 2, 2025, https://www.semanticscholar.org/paper/Idris-2%3A-Quantitative-Type-Theory-in-Practice-Brady/d670ad0f4a9448d3c0869a1519fed7fc97be60a2
63. Multiplicities — Idris2 0.0 documentation, acessado em outubro 2, 2025, https://idris2.readthedocs.io/en/latest/tutorial/multiplicities.html
64. Verifiable Credentials: A Deep Dive for the Agentic AI Era ..., acessado em outubro 2, 2025, https://shankarkumarasamy.blog/2025/02/28/verifiable-credentials-a-deep-dive-for-the-agentic-ai-era/
65. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2503.18666?
66. arxiv.org, acessado em outubro 2, 2025, https://arxiv.org/html/2503.18666v1
67. (PDF) AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents, acessado em outubro 2, 2025, https://www.researchgate.net/publication/390142942_AgentSpec_Customizable_Runtime_Enforcement_for_Safe_and_Reliable_LLM_Agents

68. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents - arXiv, acessado em outubro 2, 2025, https://arxiv.org/abs/2503.18666
69. [Literature Review] AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents - Moonlight, acessado em outubro 2, 2025, https://www.themoonlight.io/en/review/agentspec-customizable-runtime-enforcement-for-safe-and-reliable-llm-agents
70. \tool: Customizable Runtime Enforcement for Safe and Reliable LLM Agents - arXiv, acessado em outubro 2, 2025, https://arxiv.org/html/2503.18666v2
71. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents - ChatPaper, acessado em outubro 2, 2025, https://chatpaper.com/chatpaper/paper/123315
72. Parallax - A New Operating System Prototype Demonstrating Service Scaling and Service Self-Repair in Multi-core Servers - ResearchGate, acessado em outubro 2, 2025, https://www.researchgate.net/publication/221015688_Parallax_-_A_New_Operating_System_Prototype_Demonstrating_Service_Scaling_and_Service_Self-Repair_in_Multi-core_Servers
73. (PDF) Parallax - A New Operating System for Scalable, Distributed, and Parallel Computing, acessado em outubro 2, 2025, https://www.researchgate.net/publication/224257210_Parallax_-_A_New_Operating_System_for_Scalable_Distributed_and_Parallel_Computing
74. Parallax: Managing Storage for a Million Machines - USENIX, acessado em outubro 2, 2025, https://www.usenix.org/legacyurl/parallax-managing-storage-million-machines
75. The Race to Build a Distributed GPU Runtime - Voltron Data, acessado em outubro 2, 2025, https://voltrondata.com/blog/the-race-to-build-a-distributed-gpu-runtime
76. Gradient Network Complete Analysis | Review, Rating & Stats - Coinlaunch, acessado em outubro 2, 2025, https://coinlaunch.space/projects/gradient-network/
77. Gradient Network, acessado em outubro 2, 2025, https://gradient.network/
78. A Letter to Our Community: The Gradient Roadmap, acessado em outubro 2, 2025, https://gradient.network/blog/community-letter-roadmap
79. Parallax | Gradient, acessado em outubro 2, 2025, https://docs.gradient.network/research/the-gradient-stack/parallax
80. Introducing Parallax: The World Inference Engine - Gradient Network, acessado em outubro 2, 2025, https://gradient.network/blog/parallax-world-inference-engine
81. [Literature Review] Parallax: A Compiler for Neutral Atom Quantum Computers under Hardware Constraints - Moonlight, acessado em outubro 2, 2025, https://www.themoonlight.io/en/review/parallax-a-compiler-for-neutral-atom-quantum-computers-under-hardware-constraints
82. ZKTorch: Compiling ML Inference to Zero-Knowledge Proofs ... - arXiv, acessado em outubro 2, 2025, https://arxiv.org/pdf/2507.07031
83. (PDF) Circom: A Circuit Description Language for Building Zero ..., acessado em

outubro 2, 2025, https://www.researchgate.net/publication/366676429_Circom_A_Circuit_Description_Language_for_Building_Zero-knowledge_Applications
84. Introducing Lurk: A programming language for recursive zk-SNARKs, acessado em outubro 2, 2025, https://filecoin.io/blog/posts/introducing-lurk-a-programming-language-for-recursive-zk-snarks/
85. An Exploration of Zero-Knowledge Proofs and zk-SNARKs - Jerome Fisher Program in Management & Technology, acessado em outubro 2, 2025, https://fisher.wharton.upenn.edu/wp-content/uploads/2020/09/Thesis_Terrence-Jo.pdf
86. GENES: An Efficient Recursive zk-SNARK and Its Novel Application in Blockchain - MDPI, acessado em outubro 2, 2025, https://www.mdpi.com/2079-9292/14/3/492
87. Towards a zk-SNARK compiler for Wolfram language - arXiv, acessado em outubro 2, 2025, https://arxiv.org/html/2401.02935v1
88. Scaling Zero Knowledge Proofs Through Application and Proof System Co-Design - UC Berkeley EECS, acessado em outubro 2, 2025, https://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-32.pdf
89. How to construct a circuit in zkSNARK - Cryptography Stack Exchange, acessado em outubro 2, 2025, https://crypto.stackexchange.com/questions/87371/how-to-construct-a-circuit-in-zksnark
90. [2502.02387] SoK: Understanding zk-SNARKs: The Gap Between Research and Practice, acessado em outubro 2, 2025, https://arxiv.org/abs/2502.02387
91. Idea behind Zksnark protocols - Stack Overflow, acessado em outubro 2, 2025, https://stackoverflow.com/questions/75990098/idea-behind-zksnark-protocols
92. SoK: Understanding zk-SNARKs: The Gap Between Research and ..., acessado em outubro 2, 2025, https://arxiv.org/pdf/2502.02387
93. [2401.02935] Towards a zk-SNARK compiler for Wolfram language - arXiv, acessado em outubro 2, 2025, https://arxiv.org/abs/2401.02935
94. zkSNARKs Libraries for Blockchains: a Comparative Study - CEUR-WS, acessado em outubro 2, 2025, https://ceur-ws.org/Vol-3791/paper7.pdf
95. Zk-SNARKs: Under the Hood - Medium, acessado em outubro 2, 2025, https://medium.com/@VitalikButerin/zk-snarks-under-the-hood-b33151a013f6
96. [2202.06877] A Review of zk-SNARKs - arXiv, acessado em outubro 2, 2025, https://arxiv.org/abs/2202.06877
97. (PDF) Zk-SNARKs As A Cryptographic Solution For Data Privacy And Security In The Digital Era - ResearchGate, acessado em outubro 2, 2025, https://www.researchgate.net/publication/373794436_Zk-SNARKs_As_A_Cryptographic_Solution_For_Data_Privacy_And_Security_In_The_Digital_Era
98. Accelerating zk-SNARKs - MSM and NTT algorithms on FPGAs with Hardcaml, acessado em outubro 2, 2025, https://blog.janestreet.com/zero-knowledge-fpgas-hardcaml/
99. iden3/snarkjs: zkSNARK implementation in JavaScript & WASM - GitHub, acessado em outubro 2, 2025, https://github.com/iden3/snarkjs

100. Accelerating Zero-Knowledge Proofs Through Hardware-Algorithm Co-Design - People | MIT CSAIL, acessado em outubro 2, 2025, https://people.csail.mit.edu/devadas/pubs/micro24_nocap.pdf
101. Dependent types can prove your code is correct up to a specification. But how do you prove the specification is correct? - Stack Overflow, acessado em outubro 2, 2025, https://stackoverflow.com/questions/33065965/dependent-types-can-prove-your-code-is-correct-up-to-a-specification-but-how-do