COVENANT AEGIS REVIEW OF ADVANCED COMPUTING VOL. 4 · NO. 1 · 2026 · PP. 25–46

Autonomous LLM Agents and Sovereign Architectures: Reasoning, Tool Use and Governance

H. Said[1], M. Lindqvist[1], P. Vasquez[2], R. A. Mendoza[3]

[1]*Covenant Aegis Technology Division, Applied AI Laboratory*

[2]*Department of Computer Science, Partner Research Network*

[3]*Center for Trustworthy Autonomous Systems*

Received 13 Jan 2026 · Revised 1 Mar 2026 · Accepted 10 Mar 2026 · REVIEW ARTICLE · DOI 10.48-CA/advcomp.2026.0402 ·
Corresponding author: h.said@covenantaegis.com

ABSTRACT

Background. Large language models (LLMs) have evolved from text predictors into the reasoning core of autonomous agents that plan, call tools, retrieve knowledge and act over multiple steps. **Objective.** To review the technical basis of LLM agents and the reliability, security and governance requirements for production deployment, and to motivate sovereign architectures. **Methods.** Narrative synthesis of the transformer, instruction-tuning, reasoning, tool-use, retrieval and agent-security literature. **Results.** Chain-of-thought and reason-and-act loops materially improve multi-step performance; retrieval grounding improves factuality; yet hallucination compounds across steps and prompt injection ranks among the top risks for LLM applications. **Conclusion.** Agentic value depends on grounding, evaluation, least-privilege security and auditable governance; sovereign, self-hosted architectures make agentic AI viable where data control and accountability are non-negotiable.

Keywords: large language models; autonomous agents; tool use; retrieval-augmented generation; prompt injection; data sovereignty; governance

1. Introduction

The transformer architecture and the scaling of self-supervised pretraining produced language models with broad few-shot capabilities [1,2]. Aligning them to follow instructions through human feedback made them controllable enough for general use [3]. The frontier now wraps such models in agentic loops that decompose goals, invoke tools and operate with limited supervision.

2. From Models to Agents

Chain-of-thought prompting elicits intermediate reasoning that improves multi-step performance [4]. The ReAct paradigm interleaves reasoning with actions, letting a model query tools, observe results and revise its plan [5]. Toolformer shows models can learn when and how to call external APIs [6]. Composed into loops with memory and planning, these techniques yield agents that browse, compute, write code and

orchestrate workflows. Figure 1 sketches the loop.

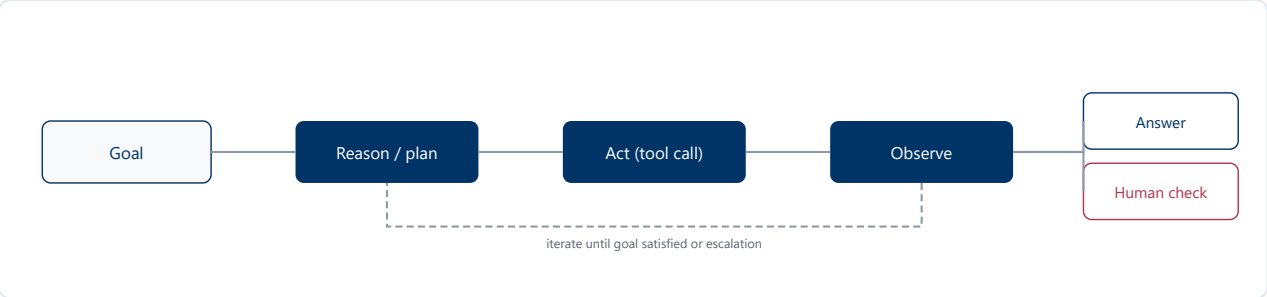


Figure 1. Reason-and-act agent loop (after ReAct [5]): the model alternates reasoning and tool actions, observing results and iterating, with human checkpoints for consequential actions.

3. Knowledge and Multi-Agent Patterns

Retrieval-augmented generation grounds outputs in an external corpus, reducing reliance on parametric memory and improving factuality and updatability [7]. Multi-agent designs assign specialised roles—planner, researcher, critic, executor—coordinating through structured messages, at the cost of latency and new failure modes.

4. Reliability and Evaluation

LLMs can produce fluent but incorrect or fabricated content; surveys catalogue the causes and mitigations of hallucination [8]. Agents compound the risk because errors propagate across steps (Figure 2). Robust deployment requires task-specific evaluation, verification of tool outputs, constrained action spaces and human checkpoints for consequential operations.

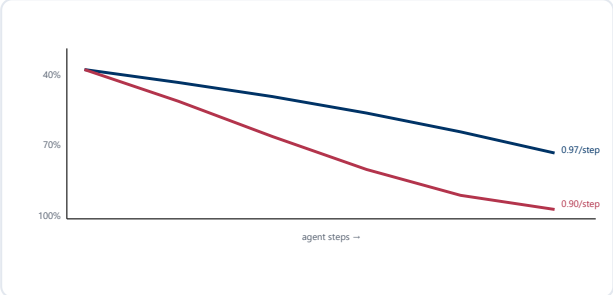


Figure 2. End-to-end task success decays geometrically with the number of dependent steps at fixed per-step reliability—motivating verification, grounding and checkpoints. Illustrative (0.97 and 0.90 per-step success).

5. Security

Agents that read untrusted content and use tools are vulnerable to prompt injection, in which adversarial text hijacks the model's instructions; the OWASP Top 10 for LLM applications ranks prompt injection and insecure tool/plugin use among the leading risks [9]. Defenses include input/output mediation, least-privilege tool scopes, content provenance and sandboxing—security cannot be an afterthought.

6. Sovereign Architectures

For regulated and security-sensitive organisations, sending data to third-party providers is often unacceptable. Sovereign

architectures self-host open-weight models and keep retrieval indices, logs and fine-tuning data within controlled infrastructure, providing auditability, data residency and independence from external availability. The trade-off is operational burden and tracking a fast-moving open-model ecosystem.

7. The Covenant Aegis Approach

We build agents with explicit tool permissions, retrieval grounding, structured evaluation harnesses, comprehensive audit logging and human-in-the-loop checkpoints for high-impact actions, deployable on sovereign infrastructure. Governance—who can do what, observed and reversible—is a first-class design concern.

8. Discussion and Conclusion

LLM agents extend language models into capable autonomous systems, but their value depends on grounding, evaluation, security and governance. Benchmarks for agentic reliability remain immature, and security is adversarial and evolving. Sovereign, auditable architectures make agentic AI viable where data control and accountability are paramount.

References

1. Vaswani A, et al. Attention is all you need. *NeurIPS*. 2017.
2. Brown TB, et al. Language models are few-shot learners. *NeurIPS*. 2020.
3. Ouyang L, et al. Training language models to follow instructions with human feedback. *NeurIPS*. 2022.
4. Wei J, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*. 2022.
5. Yao S, et al. ReAct: synergizing reasoning and acting in language models. *ICLR*. 2023.
6. Schick T, et al. Toolformer: language models can teach themselves to use tools. *NeurIPS*. 2023.
7. Lewis P, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS*. 2020.
8. Ji Z, et al. Survey of hallucination in natural language generation. *ACM Comput Surv*. 2023;55(12):1–38.
9. OWASP Foundation. OWASP Top 10 for Large Language Model Applications. 2023–2025.

9. The Transformer and Scaling Laws

The transformer's self-attention mechanism enabled parallel training on massive corpora and became the substrate for modern language models [A1]. Empirical scaling laws relate loss to model size, data and compute, and the compute-optimal analysis showed that many models were undertrained relative to their parameter count, shifting practice toward more tokens per parameter [A2]. These regularities explain why capability has tracked scale, while also signalling diminishing returns that motivate algorithmic and data-quality improvements.

Table A1. Capability stages from base model to deployed agent.

Stage	Technique	Adds
Pretraining	Self-supervision	Broad knowledge
Instruction tuning	SFT + RLHF	Controllability
Reasoning	Chain-of-thought	Multi-step problem solving
Tool use	ReAct, function calls	Action, grounding
Agent	Planning, memory	Autonomy

10. Alignment and Instruction Following

Reinforcement learning from human feedback (RLHF) aligns model behaviour with human preferences and made general-purpose assistants usable [A3]. Constitutional and AI-feedback methods reduce reliance on human labels for harmlessness. Alignment is imperfect: models can be sycophantic, refuse incorrectly, or be jailbroken, and reward models can be gamed. For agents, alignment failures compound because the model both decides and acts.

11. Retrieval-Augmented Generation

RAG grounds generation in retrieved documents, improving factuality, enabling source attribution and allowing knowledge updates without retraining [A4]. Quality depends on retrieval relevance, chunking, and the model's faithfulness to retrieved context; poor retrieval yields confident but unsupported answers. Figure A1 illustrates the factuality gain from grounding relative to closed-book generation.

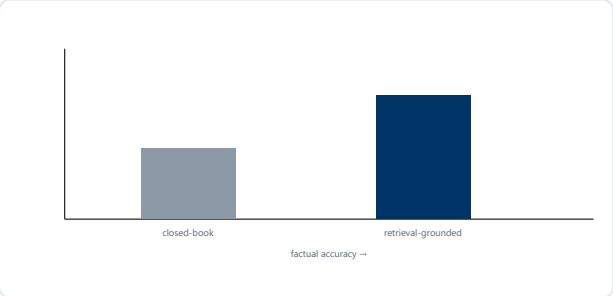


Figure A1. Schematic factuality improvement from retrieval grounding on knowledge-intensive tasks [A4]. Conceptual; magnitude varies by task and corpus.

12. Evaluation of Agents

Evaluating agents is harder than evaluating single responses: success is multi-step, stochastic and tool-dependent. Benchmarks span tool use, web navigation and software tasks, but reproducibility and contamination are concerns. Robust practice combines task-success metrics, trajectory inspection, adversarial/red-team tests and human review for consequential domains. Because per-step errors compound (Section 4), even modest per-step failure rates produce large end-to-end failure on long tasks.

13. Security: Prompt Injection and Tool Abuse

Agents that ingest untrusted content face prompt injection, where adversarial text overrides intended instructions; indirect injection via retrieved web pages or documents is especially dangerous for tool-using agents [A5]. The OWASP Top 10 for LLM applications ranks prompt injection, insecure output handling and excessive agency among the leading risks [A6]. Defenses include privilege separation between trusted instructions and untrusted data, least-privilege tool scopes, output validation, human confirmation for high-impact actions and sandboxing—no single control suffices.

14. Sovereign Deployment and Governance

Sovereign architectures self-host open-weight models and keep retrieval indices, logs and fine-tuning data within controlled infrastructure, providing data residency, auditability and independence from external providers—often required in regulated sectors and addressed by emerging governance frameworks [A7]. The trade-off is operational burden and the need to track a fast-moving open-model ecosystem. Governance—scoped permissions, comprehensive logging, reversibility and human checkpoints—is treated as a first-class design requirement.

15. Future Directions

Priorities include reliable long-horizon planning, verifiable tool use, robust defenses against injection, faithful uncertainty and provenance, and standardised agent evaluation. The viability of agentic AI in high-stakes settings will be determined by grounding, security and governance rather than raw capability.

Extended References

10. Vaswani A, et al. Attention is all you need. *NeurIPS*. 2017.
11. Hoffmann J, et al. Training compute-optimal large language models (Chinchilla). *NeurIPS*. 2022.
12. Ouyang L, et al. Training language models to follow instructions with human feedback. *NeurIPS*. 2022.
13. Lewis P, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS*. 2020.
14. Greshake K, et al. Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection. *ACM AISec*. 2023.
15. OWASP Foundation. OWASP Top 10 for Large Language Model Applications. 2023–2025.
16. NIST. AI Risk Management Framework (AI RMF 1.0). 2023.

Author contributions: HS led the review; ML and PV contributed reasoning and retrieval sections; RAM supervised governance. All authors revised. **Funding:** Covenant Aegis Technology Division. **Competing interests:** Authors are affiliated with Covenant Aegis.

© 2026 Covenant Aegis Technology Division. This review synthesises publicly available peer-reviewed and community literature; figures are illustrative. For professional and educational use.