Runtime Governance for Security:

Microsoft Enterprise Agent Stack

Rao Mikkilineni | Ben Hendrick | Max Michaels

MINDFUL AI
FOUNDATION

Abstract. Among publicly documented enterprise AI platforms, Microsoft appears to offer the broadest and most operationally mature agent stack, spanning Microsoft Foundry, Foundry Control Plane, Agent Framework, GitHub Copilot coding agent, and Agent 365. This stack materially advances the state of the art in enterprise agent engineering by integrating model access, tool catalogs, multi-agent workflows, persistent state, observability, security, compliance, and lifecycle management. Yet its underlying architectural pattern remains one of externally governed agency: governance is added through control planes, identity services, policy layers, observability systems, and human-in-the-loop orchestration rather than being constitutive of the runtime substrate itself. This manuscript argues that the Microsoft stack should therefore be understood as the leading expression of the current paradigm, not as its final resolution.

Against this background, Mindful Machine Architecture is assessed as a more radical but earlier-stage research program. Drawing on Burgin's General Theory of Information, structural machines, digital genome concepts, and the Autopoietic and Meta-Cognitive Operating System (AMOS), it proposes that durable machine intelligence requires persistent identity, event-grounded memory, runtime invariants, policy mediation, and endogenous self-regulation. The paper offers a theoretical and practical comparison: Microsoft currently leads in ecosystem breadth, tooling, and enterprise readiness; Mindful Machines offer the stronger hypothesis about how to integrate the computer and the computed in a single constitutional substrate. The manuscript concludes that the future path for trustworthy AI likely requires combining the operational discipline of today's control-plane architectures with the deeper runtime self-modeling and governance-first principles articulated by Mindful Machines.

1. Introduction

Current discussion of advanced AI is often distorted by a false binary. On one side are triumphalist claims that contemporary agent stacks already constitute a decisive leap toward autonomous intelligence. On the other side are skeptical claims, following Floridi, that current systems amount to little more than *"agency without intelligence"* (Floridi, 2023). Both positions capture something real, but neither is sufficient for architectural assessment. The relevant question is not merely whether systems can act, but how their action is organized, governed, remembered, and corrected over time.

Among publicly documented enterprise offerings, Microsoft appears to lead the current state of the art in breadth of platform integration. Microsoft Foundry presents itself as an "AI app and agent factory" with multi-agent orchestration, tool catalogs, memory, knowledge integration, and centralized operational governance (Microsoft, 2026a). Foundry Control Plane adds cross-project fleet management, observability, compliance enforcement, and integrated security signals (Microsoft, 2026b). Agent 365 extends this pattern into an enterprise control plane for all organizational agents regardless of where they are built or acquired (Microsoft, 2026c). GitHub Copilot's coding agent adds asynchronous software execution in secure, customizable environments powered by GitHub Actions and submits pull requests for review (GitHub, 2025). Taken together, these are not thin wrappers around a language model. They are the most comprehensive public manifestation of enterprise agent engineering currently on offer.

However, leading the present state of the art is not the same as solving the deeper problem of trustworthy machine intelligence. Microsoft's own documentation emphasizes governance layers, security layers, observability layers, and maturity models for managing agent sprawl, automation complexity, and fragmented operational practices (Microsoft, 2026d). This suggests that the present paradigm is one of increasingly capable agency managed through increasingly sophisticated external control planes. By contrast, the Mindful Machine literature proposes that governance, continuity, and accountability must become intrinsic properties of the runtime substrate itself rather than administrative overlays (Mindful AI Foundation, 2026; Mikkilineni & Kelly, 2025).

This manuscript develops a theoretical and practical assessment of both trajectories. Its central claim is that Microsoft represents the most advanced expression of today's externally governed agent paradigm, whereas Mindful Machines articulate a more ambitious but less widely validated constitutional architecture for the next phase. The distinction matters because it reframes the future of AI not as a contest between more and less capable models, but as a contest between different organizational principles for machine agency.

2. Microsoft and the Current State of the Art

Microsoft's strength lies in architectural breadth. Foundry is explicitly designed for application developers, machine learning engineers, and IT administrators, and it combines model access, orchestration workflows, tool catalogs, memory, knowledge grounding, observability, centralized asset management, and enterprise controls in one platform family (Microsoft, 2026a). Foundry Control Plane centralizes management for AI agents, models, and tools, and Microsoft states that without it organizations must manage agents, models, and compliance through separate portal blades and per-project views (Microsoft, 2026b). Agent 365 then generalizes the idea into a cross-organizational control plane for governing all agents at scale (Microsoft, 2026c).

This stack is reinforced by Microsoft Agent Framework, which provides session-based state management, multi-agent workflows, filters, telemetry, and long-running stateful execution patterns (Microsoft, 2026e). Durable Task support makes the point even more clearly: human-in-the-loop workflows, pauses, resumes, and checkpointed long-running execution are treated as first-class platform concerns rather than ad hoc application patches (Microsoft, 2026f). GitHub Copilot's coding agent shows the same pattern in software engineering: assign a task, spin up a secure development environment, explore the repository, make code changes, run in the background, and produce a pull request for human review (GitHub, 2025).

In practical terms, Microsoft's stack therefore advances the state of the art in five important ways. First, it unifies model, tool, and workflow composition. Second, it operationalizes enterprise identity and governance for agents. Third, it makes observability and security explicit parts of the platform. Fourth, it provides concrete developer pathways from low-code to pro-code agents. Fifth, it recognizes that long-running agent systems are not merely model inference endpoints but software systems requiring lifecycle discipline.

For these reasons, it is fair to say that Microsoft currently leads the publicly documented enterprise state of the art. The company is not simply shipping a model or a chatbot. It is offering a layered agent ecosystem: development substrate, runtime substrate, observability substrate, governance substrate, and security substrate. That is a major advance over isolated copilots.

3. The Managerial Recursion Problem in External Control Planes

Yet Microsoft's own documentation also reveals the limitation of the current paradigm. Foundry Control Plane is needed because agents, models, tools, and compliance otherwise remain fragmented across views and teams (Microsoft, 2026b). Microsoft's maturity guidance warns specifically about "operations silos" and "automation complexity," acknowledging the risk of brittle systems that become difficult to understand or modify as governance and operations are increasingly automated (Microsoft, 2026d). Its observability guidance states that traditional observability no longer suffices because AI systems are probabilistic rather than deterministic; visibility must therefore expand beyond conventional logs, metrics, and traces into AI-native evaluation and governance signals (Microsoft, 2026g). In short, the stack is becoming governable by adding more management layers above increasingly complex agent systems.

This is the managerial recursion problem: ***who manages the managers?*** Microsoft's answer is pragmatic and sophisticated. Agent 365 extends identity, governance, and security down into agents; Foundry Control Plane centralizes fleet management; Entra, Defender, Purview, and admin center tooling extend enterprise control; and human-in-the-loop review is imposed where deterministic oversight is required (Microsoft, 2026c; Microsoft, 2026h). But these mechanisms remain primarily exogenous to the application's own constitutive order. They supervise, constrain, and observe the system from above or around it. They do not amount to a documented system-wide self-model in which application identity, commitments, state transitions, and corrective policies are unified as endogenous runtime properties.

That distinction is crucial. A persistent enterprise identity for an agent is not yet a persistent application-level sense of self. Cross-project visibility is not the same as intrinsic continuity of commitments across changing execution contexts. Human-in-the-loop checkpoints improve safety, but they are orchestrator-enforced pauses rather than evidence that the system itself carries a constitutional account of what it must preserve. Microsoft's paradigm

is therefore best described as externally governed agency at scale. It is the leading operational answer to cloud-native agent complexity, but it still inherits the core cloud-era pattern in which IaaS, PaaS, tool planes, policy planes, and observability planes must be composed and managed together.

Cockshott, Mackenzie, and Michaelson (2012) describe computation as the modeling of one part of physical reality by another, and Turing's own introduction of oracles underscores that classical formal models did not settle every question about how computation might be extended when fixed procedures encounter uncomputable or open-ended structure (Hodges, 2024). Read in that broader frame, Microsoft's current evolution can be seen as an intelligent bottom-up attempt to compensate for a deeper separation between the computer and the computed. The system acquires more surrounding governance because the governance is not yet constitutive of the application organism itself.

4. Mindful Machine Architecture and the AMOS Alternative

Mindful Machine Architecture begins from a different premise. The 2021 Burgin and Mikkilineni paper argues that knowledge processing requires means of working with knowledge representations rather than data alone, because knowledge is an abstract structure and intelligence requires operations over such structures (Burgin & Mikkilineni, 2021). The 2022 paper extends this claim by arguing that if digital machines are to mimic the sentient, resilient, and intelligent behaviors of living organisms, then they must be infused with autopoietic and cognitive behaviors; current symbolic and sub-symbolic systems, by contrast, depend on external entities to find resources, deploy, configure, monitor, and manage them (Mikkilineni, 2022).

Later papers translate this theoretical move into architectural terms. The 2024 article on digital genome and self-regulating distributed software architecture proposes a cloud-agnostic distributed application architecture that uses a digital genome to express and enforce form and function across heterogeneous resources (Mikkilineni, 2024). The 2025 Computers article then presents Mindful Machines as distributed AI systems organized around a persistent digital genome and executed through AMOS, an Autopoietic and Meta-Cognitive Operating System (Mikkilineni & Kelly, 2025). The Mindful AI Foundation's public framing sharpens the intended contrast with current AI: preserve identity, enforce invariants, track commitments, recover from drift,

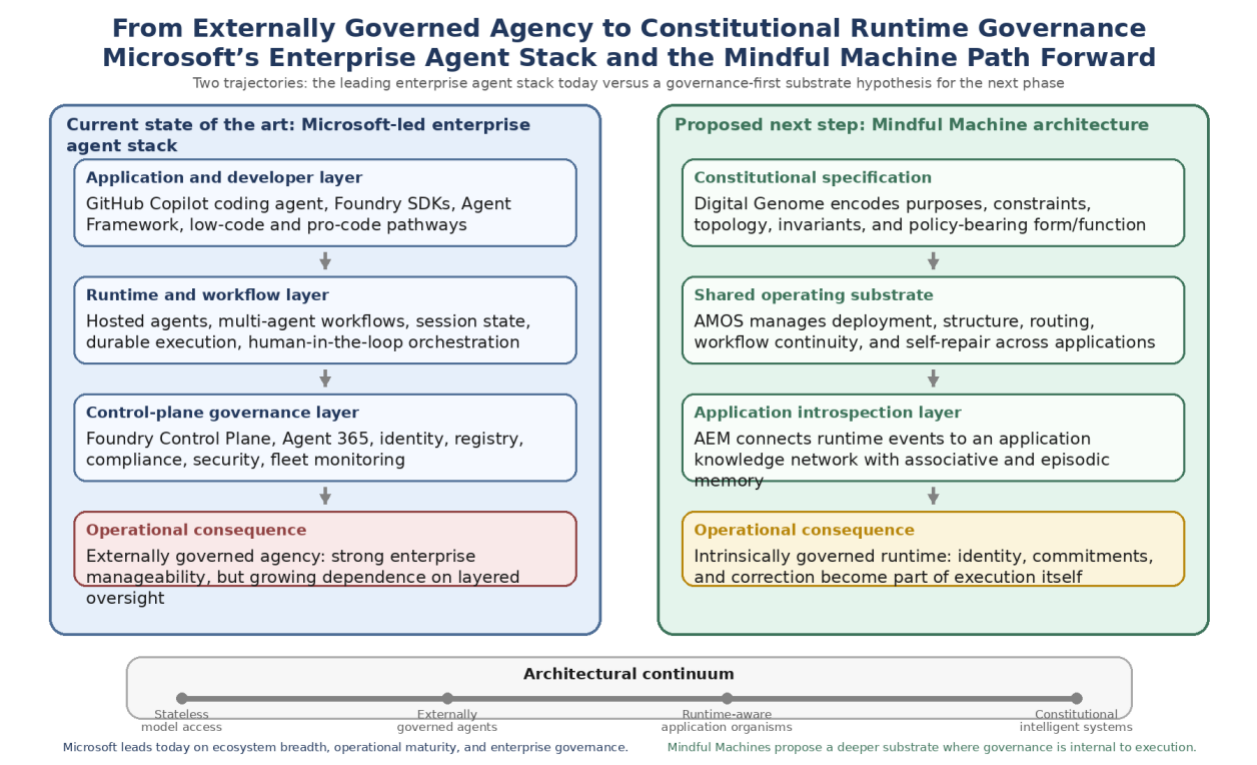
and remain accountable over time; governance must be embedded in the substrate, not layered on top (Mindful AI Foundation, 2026).

Public implementation materials give the AMOS claim additional specificity. A public AMOS implementation transcript describes four core managers in the operating substrate: the Autopoietic Process Manager (APM), Structural Event Manager (SEM), Cognitive Network Manager (CNM), and Service Workflow Manager (SWM), along with an Application Event Manager (AEM) that keeps the graph database updated with real-time application status and interactions (Mikkilineni, 2025b). In this decomposition, APM instantiates service entities, SEM connects them according to valid schema-defined paths, CNM adapts network behavior, SWM manages anomaly-triggered workflow correction, and AEM links operating-system events to an application knowledge network represented in a graph database. The same public material demonstrates at least two concrete applications-a video-on-demand system with failover and a credit-default system with event timelines and auditability-under the same operating substrate (Mikkilineni, 2025b).

The scholarly significance of this architecture is not that it has already achieved universal validation. It has not. The public evidence is still early-stage, concentrated in peer-reviewed papers, foundation materials, and public demonstrations rather than broad third-party replication. Its importance lies elsewhere: it offers a coherent answer to the question Microsoft's stack still leaves open. Instead of thickening the control plane around increasingly autonomous agents, Mindful Machines propose that the system's identity, event memory, policy mediation, and continuity constraints be constituted in the runtime substrate itself. In that sense, AMOS is not merely an orchestrator. It is a constitutional operating layer.

Figure 1

Two architectural trajectories: Microsoft-led enterprise AI and the Mindful Machine path forward



Note. Figure 1 clarifies the core architectural contrast. The left side depicts the Microsoft-led state of the art as a layered control-plane ecosystem: powerful, enterprise-ready, and operationally disciplined, but reliant on external governance surfaces to manage agent behavior at scale. The right side depicts the Mindful Machine hypothesis: a digital genome instantiated through a shared operating substrate (AMOS), connected through AEM to an application knowledge network and policy-oracle layer, so that memory, invariants, continuity, and repair are not administrative afterthoughts but runtime properties. The figure should not be read as claiming that one side is finished and the other is proven. Rather, it shows two trajectories: one mature in enterprise operations, the other deeper in constitutional ambition.

5. Comparative Assessment

A fair comparison must separate operational maturity from architectural depth. Microsoft is ahead on operational maturity. It offers the most visible and comprehensive integration of agent building, orchestration, control-plane governance, security, and developer tooling presently available in public documentation. For enterprises that need agents today, this matters enormously. Real organizations need identity, lifecycle management,

observability, exception handling, policy enforcement, and auditability now, not only as future theory.

Mindful Machines are stronger on architectural depth. The literature does not merely add guardrails to stochastic or workflow-based systems; it proposes a different ontological basis for machine organization. GTI and structural machines widen the frame from token processing to knowledge processing; the digital genome introduces an executable constitutional specification; AMOS introduces autopoietic and meta-cognitive management; and the application knowledge network introduces event-grounded associative and episodic memory. If this architecture matures, it could address the problem of system-wide self-modeling, not only fleet-wide observability.

Each side also has clear limitations. Microsoft's stack, despite its sophistication, still appears to externalize governance. Its own guidance emphasizes maturity models, control layers, and administrative coordination, which suggests that complexity is being managed rather than dissolved. Mindful Machines, by contrast, have not yet demonstrated the independent ecosystem maturity, standardized benchmarks, or broad industrial replication that would make their claims incontestable. The public demos are promising, but they are not the same as industry-wide validation.

The strongest conclusion is therefore synthetic rather than sectarian. Microsoft's stack is the current best expression of externally governed enterprise agency. Mindful Machines are the strongest public articulation, in this discourse, of a move toward intrinsically governed machine organization. The future of trustworthy AI likely requires a transition from the former toward the latter, while preserving the hard-won operational lessons of the former.

Table 1

Fair comparison of Microsoft's current enterprise stack and Mindful Machine architecture

Dimension	Microsoft-led state of the art	Mindful Machine architecture	Assessment
Primary organizing principle	Externally governed agency at scale	Constitutional runtime governance	Different architectural depths, not just different feature sets
Core runtime pattern	Models, tools, workflows, control planes, admin surfaces	Digital genome, AMOS substrate, application knowledge network	Microsoft leads in platformization; Mindful Machines lead in substrate hypothesis
Identity	Enterprise identity for agents and registries	Persistent system identity bound to genome, topology, and events	Agent identity is necessary but not yet a full self-model
Memory	Context memory, session state, observability history	Associative and episodic memory in graph-based knowledge network	Mindful Machines propose stronger event-grounded continuity
Governance	Policies, compliance, security, human checkpoints imposed through control planes	Policies and invariants mediated within the runtime substrate	External vs intrinsic governance is the decisive difference
Fault handling	Recovery, durability, and HITL through orchestration and platform services	Autopoietic repair and workflow redirection via substrate managers	Both address reliability; only one treats it as constitutive
Operational maturity	High and rapidly growing	Early, prototype and literature centered	Microsoft clearly leads current enterprise readiness
Evidence base	Official documentation, broad product ecosystem, production-oriented guidance	Peer-reviewed papers, foundation materials, public demos	Breadth favors Microsoft; conceptual

			novelty favors Mindful Machines
Main risk	Managerial recursion and layered complexity	Validation gap and limited third-party replication	Each paradigm has a different failure mode
Best current reading	State of the art in present enterprise agent systems	Promising path toward next-generation governed machine systems	They should be compared as successive paradigms, not direct market substitutes

6. Discussion and Research Agenda

Three research and engineering priorities follow from this comparison.

First, the field needs better ways to evaluate system-wide self-modeling. Current benchmarks largely measure task performance, response quality, latency, cost, or safety-policy compliance. They do not adequately measure persistence of commitments, continuity of identity under failure, invariance preservation during adaptation, or application-level introspective coherence. If Mindful Machine claims are to be tested fairly, these dimensions need benchmarkable operational definitions.

Second, enterprise AI needs convergence between control-plane excellence and runtime constitutionalism. Microsoft's stack shows what mature operational discipline looks like: security, observability, lifecycle management, and managed execution. Mindful Machines show what a deeper substrate ambition looks like: identity and invariance as engineering primitives, event-grounded memory, and governance-first architecture. The future is unlikely to reward either pure externalism or pure conceptual novelty alone. It will reward systems that can combine both.

Third, the discussion of AI must move beyond the shorthand of "agents" and "wrappers." Current enterprise systems are already more than wrappers, but they are still not obviously self-governing organisms. Conversely, governance-first architectures cannot remain at the level of metaphor. They must continue to demonstrate multi-application portability, fault tolerance, cross-cloud execution, cost discipline, and third-party reproducibility. The path forward is empirical, architectural, and comparative.

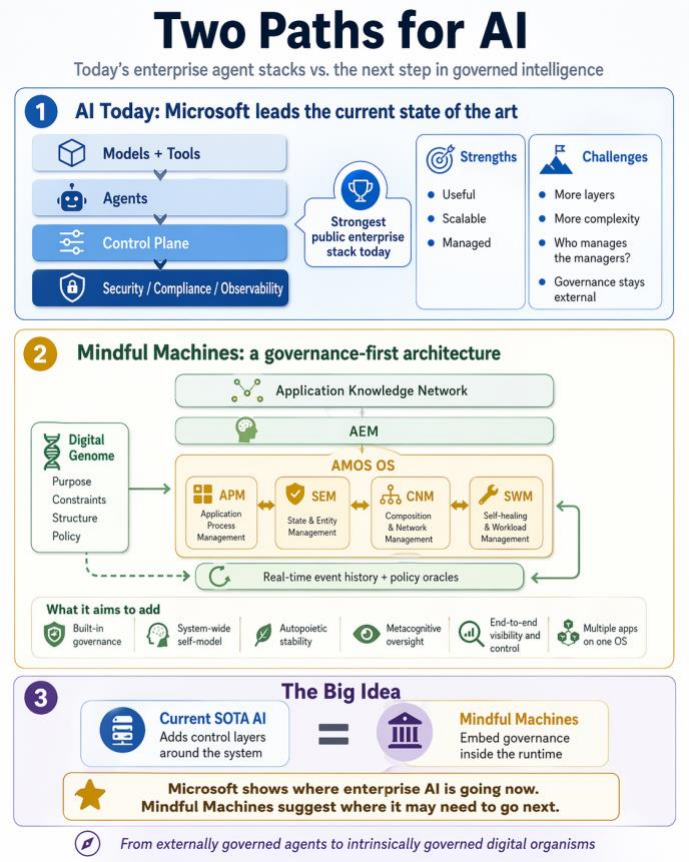
In that sense, the field may be entering a productive transition. The Microsoft-led state of the art demonstrates how far externally governed agency can be industrialized. Mindful Machines demonstrate why that might still be insufficient. Taken together, they improve the discussion: one by showing what can already be built at scale, the other by clarifying what the next paradigm would have to solve.

7. Conclusion

Floridi was right to warn that contemporary AI often displays agency without intelligence. Microsoft complicates that diagnosis but does not overturn it. Its 2025-2026 stack shows that agency can be made persistent, observable, secure, and enterprise-manageable across increasingly complex workflows. That is a major achievement. Yet the architecture remains, on the public record, one of externally governed agency. Governance is thickened around the system by control planes, security products, registries, observability layers, and human checkpoints.

Mindful Machine Architecture offers a different wager. It argues that trustworthiness cannot be added late. It must be built into the substrate through digital genome, knowledge-network execution, event-grounded memory, and autopoietic and meta-cognitive management. The public evidence for this paradigm is still early, but the theoretical contribution is clear and the implementation direction is now concrete enough to merit serious attention.

The most productive path for the field is therefore neither dismissal of the present nor premature triumphalism about the future. Microsoft's stack should be studied as the leading operational form of the current paradigm. Mindful Machines should be studied as a substantive candidate for what comes next. The future of AI will depend on whether we can move from managing intelligent-seeming systems from the outside toward constituting governed intelligent systems from within.



Keywords: enterprise AI, agentic AI, Microsoft Foundry, Microsoft Agent 365, GitHub Copilot, Mindful Machines, AMOS, General Theory of Information, digital genome, AI governance

References

Burgin, M., & Mikkilineni, R. (2021). From data processing to knowledge processing: Working with operational schemas by autopoietic machines. Big Data and Cognitive Computing, 5(1), 13. https://doi.org/10.3390/bdcc5010013

Cockshott, P., Mackenzie, L. M., & Michaelson, G. (2012). Computation and its limits. Oxford University Press.

Floridi, L. (2023). AI as agency without intelligence: On ChatGPT, large language models, and other generative models. Philosophy & Technology, 36, Article 15. https://doi.org/10.1007/s13347-023-00621-y

GitHub. (2025, May 19). GitHub Copilot: Meet the new coding agent. The GitHub Blog. https://github.blog/news-insights/product-news/github-copilot-meet-the-new-coding-agent/

Hodges, A. (2024). Alan Turing. In E. N. Zalta & U. Nodelman (Eds.), The Stanford Encyclopedia of Philosophy (Summer 2024 ed.). https://plato.stanford.edu/entries/turing/

Microsoft. (2026a). What is Microsoft Foundry? Microsoft Learn. https://learn.microsoft.com/en-us/azure/foundry/what-is-foundry

Microsoft. (2026b). What is Microsoft Foundry Control Plane? Microsoft Learn. https://learn.microsoft.com/en-us/azure/foundry/control-plane/overview

Microsoft. (2026c). Microsoft Agent 365 documentation. Microsoft Learn. https://learn.microsoft.com/en-us/microsoft-agent-365/

Microsoft. (2026d). Agentic AI maturity model - AI governance and security. Microsoft Learn. https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/maturity-model-security-governance

Microsoft. (2026e). Microsoft Agent Framework overview. Microsoft Learn. https://learn.microsoft.com/en-us/agent-framework/overview/

Microsoft. (2026f). Durable Task extension for Microsoft Agent Framework. Microsoft Learn. https://learn.microsoft.com/en-us/azure/durable-task/sdks/durable-agents-microsoft-agent-framework

Microsoft. (2026g). Observability for generative AI and agentic AI systems. Microsoft Learn. https://learn.microsoft.com/en-us/security/zero-trust/sfi/observability-ai-systems

Microsoft. (2026h). Secure autonomous agentic AI systems. Microsoft Learn. https://learn.microsoft.com/en-us/security/zero-trust/sfi/secure-agentic-systems

Mindful AI Foundation. (2026). Mindful AI Foundation. https://mindfulai.foundation/

Mikkilineni, R. (2022). A new class of autopoietic and cognitive machines. Information, 13(1), 24. https://doi.org/10.3390/info13010024

Mikkilineni, R. (2024). Digital genome and self-regulating distributed software architecture. Computers, 13(9), 220. https://doi.org/10.3390/computers13090220

Mikkilineni, R. (2025b). AMOS: Autopoietic operating system for distributed applications [LinkedIn post]. LinkedIn. https://www.linkedin.com/posts/raomikkilineni_amos-an-autopoietic-and-meta-cognitive-activity-7404658862231994368-UrPW

Mikkilineni, R., & Kelly, W. P. (2025). From static prediction to mindful machines: A paradigm shift in distributed AI systems. Computers, 14(12), 541. https://doi.org/10.3390/computers14120541

Turing, A. M. (1939). Systems of logic based on ordinals. Proceedings of the London Mathematical Society, s2-45(1), 161-228.

*****

MINDFUL AI
FOUNDATION