EIGENVECTOR RESEARCH • Inholland University of Applied Sciences

Patternomics: A Formal Theory of Execution Pattern Optimization in Enterprise Agentic AI Systems

Marco H.K. van Hurne

EIGENVECTOR RESEARCH

Inholland University of Applied Sciences

marco.vanhurne@eigenvector.eu • marco.vanhurne@inholland.nl

May 2026

Author Note

Marco H.K. van Hurne is affiliated with EIGENVECTOR RESEARCH and the Research and Innovation Centre Techniek, Ontwerpen en Informatica at Inholland University of Applied Sciences, Haarlem, Netherlands.

This paper builds on prior work published by EIGENVECTOR RESEARCH, including the Process Automation Suitability Framework and Design Engine (PASF-PADE, 2026), the Ontological Compliance Gateway (OCG, 2026), the PASF Mapping and Agentification Risk study (2026), and the Technical Debt-Aware Prompting Framework (2025).

Correspondence concerning this article should be addressed to Marco H.K. van Hurne, EIGENVECTOR RESEARCH. E-mail: marco.vanhurne@eigenvector.eu

Abstract

Contemporary enterprise AI discourse remains predominantly model-centric, focusing on parameter counts, token throughput, and benchmark capabilities. However, empirical evidence from production deployments indicates that the primary driver of system-level cost, latency, and reliability is not the underlying model, but the execution structure—the topology and orchestration pattern of the multi-agent system. This paper introduces *Patternomics*, a formal theory of execution pattern optimization for enterprise agentic AI. Building upon the Process Automation Suitability Framework (PASF), the Ontological Compliance Gateway (OCG), and the Agentification Factory model, we synthesize findings from 106 primary sources and 177 enterprise deployments to demonstrate that pattern-level optimization yields cost reductions of up to 97.9% and latency improvements of 62.5%. We formalize the Pattern Calculus, introduce a Tri-Objective Cost Model encompassing inference, coordination, and governance, and define the Pattern Efficiency Frontier (PEF). The paper concludes by positioning Patternomics as the necessary theoretical foundation for the next phase of enterprise AI maturity, moving from ad-hoc agent design to deterministic, governed, and economically optimized pattern engineering.

Keywords: Patternomics, agentic AI, multi-agent systems, enterprise AI governance, execution pattern optimization, PASF, OCG, pattern efficiency frontier, tri-objective cost model, pattern calculus

The transition of Artificial Intelligence from experimental sandboxes to enterprise production environments has exposed a structural flaw in how the industry evaluates AI systems. The prevailing paradigm is overwhelmingly model-centric: optimization efforts focus on parameter counts, context windows, and token generation speed [1]. However, as organizations deploy complex, multi-step workflows using Agentic AI, the binding constraints on deployment are rarely the cognitive limits of the underlying Large Language Models (LLMs). Instead, the primary drivers of system failure, cost overruns, and latency are architectural [2].

This reality is already visible in enterprise process automation. As demonstrated in the Process Automation Suitability Framework (PASF) and the Process Automation Design Engine (PADE), the viability of an automated process depends heavily on its execution modality and governance requirements, not just model capability [3] [15] [25]. Furthermore, the deployment of autonomous agents in regulated environments requires deterministic, pre-execution validation—a requirement addressed by the Ontological Compliance Gateway (OCG) [4] [21].

Despite these practical advancements, the field lacks a unified theoretical framework to describe, measure, and optimize the structures that connect agents together. We propose that the execution pattern itself—the topology, the routing logic, and the coordination mechanism—must be treated as the primary unit of economic and architectural optimization. We term this discipline **Patternomics**.

This paper formalizes Patternomics by synthesizing empirical evidence from recent multi-agent system (MAS) research and enterprise deployments, including the Agentification Factory model [16] and the Agentic Success Pattern Framework (ASPF) [17]. We introduce a formal Pattern Calculus, a Tri-Objective Cost Model, and the concept of the Pattern Efficiency Frontier (PEF), providing a rigorous foundation for enterprise agentic AI design.

The necessity of Patternomics is not theoretical; it is an empirical reality emerging simultaneously across multiple sub-disciplines of AI research. A systematic review of 106 primary sources published between 2023 and 2026 reveals that the field is already optimizing execution patterns, albeit without a unifying nomenclature [2]. We categorize this empirical foundation into five pillars.

2.1 Topology-Dependent Error Amplification

The assumption that adding more agents to a system linearly increases performance is demonstrably false. Research indicates that multi-agent systems exhibit topology-dependent error amplification. For instance, Kim et al. (2025) demonstrated that decentralized, independent agent topologies can amplify baseline hallucination rates by a factor of 17.2x, whereas centralized hub-and-spoke models limit amplification to 4.4x [6].

Furthermore, MAS architectures only provide positive returns when the single-agent baseline performance exceeds a specific threshold (the "Capability Saturation Threshold," typically around 45%). Below this threshold, the coordination overhead and cascading errors result in a net negative performance delta compared to a single agent [7].

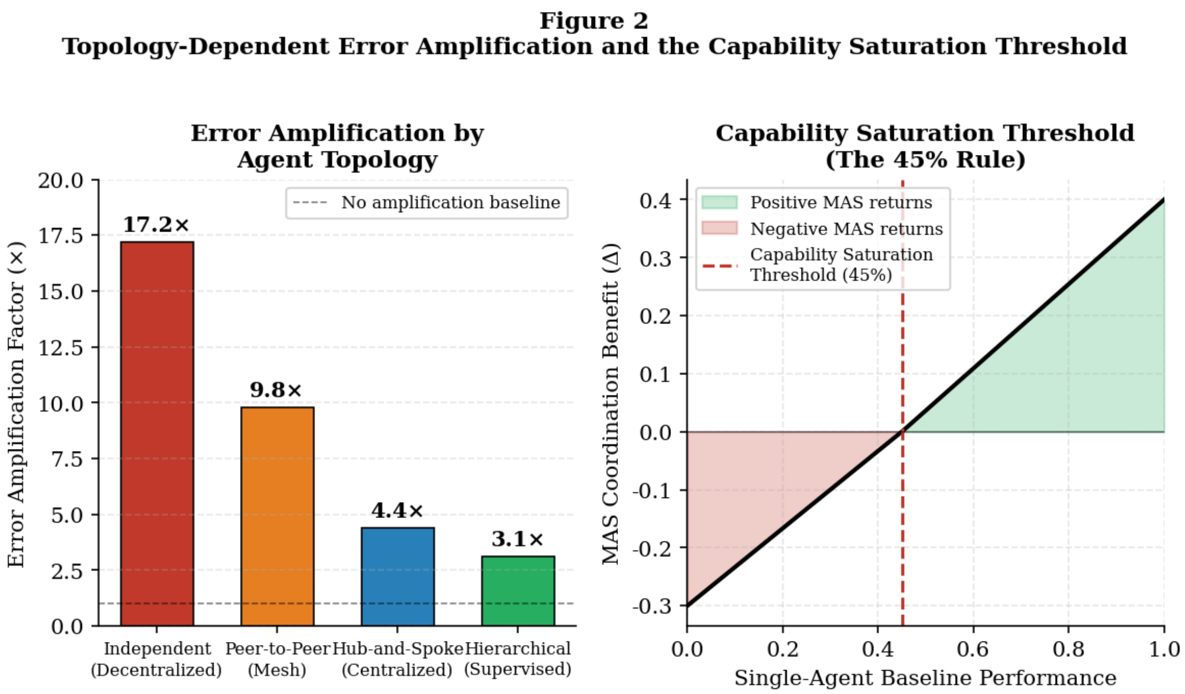

Figure 2

Topology-Dependent Error Amplification and the Capability Saturation Threshold

2.2 The Economics of Coordination Overhead

As agent topologies grow in complexity, the cost of coordination (C_c) rapidly overtakes the cost of inference (C_i). In highly connected mesh topologies, agents spend a disproportionate amount of their token budget summarizing context for other agents, negotiating task boundaries, and resolving conflicts. This phenomenon aligns with Garnier's (2026) General Equilibrium Theory for LLM agents, which models token economies as Walrasian markets where coordination overhead acts as a transaction friction [8]. This is further formalized in the Tokenomics of Agentic AI framework, which introduces the Software-Defined per-Deliverable (SDpD) benchmark and the Dynamic Token Budget Negotiation Protocol (DTBNP) to optimize resource allocation [26].

2.3 Backbone-Then-Topology Design

The shift toward pattern-level optimization is evident in frameworks like AgentBalance, which treats the LLM as a commoditized "backbone" and focuses optimization entirely on the topology [9]. Similarly, Budget-Constrained Autonomous Multi-Agent Systems (BAMAS) utilize Integer Linear Programming (ILP) to dynamically allocate tasks across a topology based on strict cost constraints, proving that topology optimization yields higher ROI than model fine-tuning for complex workflows [10].

2.4 Pattern Pruning as Structural Cost Reduction

The most compelling evidence for Patternomics comes from the emerging practice of pattern pruning—the systematic removal of redundant agents or communication edges from an orchestration graph. Empirical results are striking: - **MCP+SKILL (Ray, 2026):** Achieved a 97.9% cost reduction by pruning redundant verification loops [11]. - **TB-CSPN (Borghoff et al., 2025):** Delivered 62.5% faster processing through optimized state-machine routing [12]. - **CASTER/EvoRoute:** Demonstrated 80% cost reductions by dynamically bypassing unnecessary agent nodes [13].

2.5 The Governance Imperative

In enterprise environments, optimization cannot be purely economic or performance-driven; it must be constrained by governance. As established in the OCG architecture and the Governed Runtime Architecture Framework (GRAF), deterministic validation gates must be inserted into the execution pattern to ensure semantic coherence and policy compliance [4] [23] [24]. These governance nodes introduce a third cost dimension (C_g) that fundamentally alters the optimal pattern topology. The Roundtrip Value Governance framework further emphasizes that execution patterns must be ranked by recognized value per total enterprise cost under hard governance constraints [22].

To transition from empirical observation to engineering discipline, Patternomics requires a formal mathematical structure. We introduce three core components: the Pattern Calculus, the Tri-Objective Cost Model, and the Pattern Efficiency Frontier.

3.1 The Pattern Calculus

We define an execution pattern P as a 4-tuple:

$$P = (V, E, \Omega, G)$$

Where: - V is the set of agent nodes, each defined by its capability, inference cost, and latency. - E is the set of directed communication edges, representing data flow and coordination overhead. - Ω is the orchestration mechanism (e.g., sequential, ReAct, state-machine). - G represents the governance constraints (e.g., OCG validation gates, human-in-the-loop requirements).

The Pattern Calculus defines three primitive operations for optimizing P:

1. **Composition ($P_1 \circ P_2$):** Combining two patterns to handle higher-complexity tasks.
2. **Pruning ($P \setminus e$):** Removing an edge or node to reduce coordination cost without violating G.

3. **Substitution ($P[v_i \leftarrow v_j]$):** Replacing an expensive agent node with a cheaper, specialized node (e.g., replacing an LLM with a deterministic script).

Figure 4
The Pattern Calculus: Formal Structure of an Execution Pattern and Its Primitive Operations

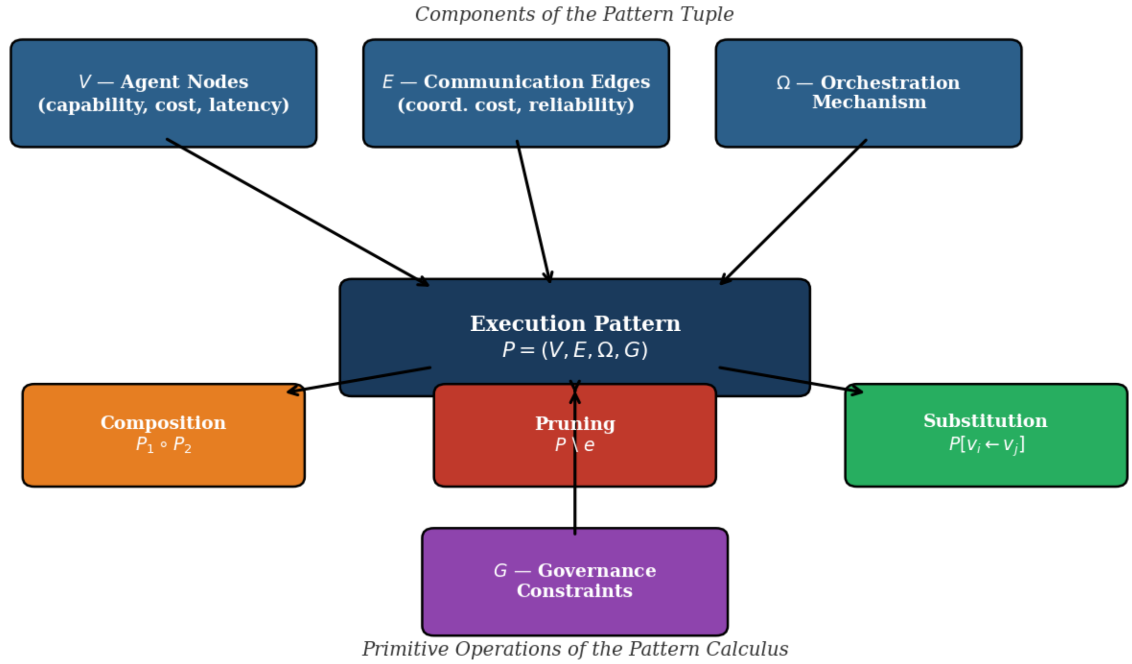

Figure 4

The Pattern Calculus

3.2 The Tri-Objective Cost Model

Traditional token-counting fails to capture the true cost of enterprise AI. Patternomics introduces a Tri-Objective Cost Model:

$$C_{total}(P) = C_i(V) + C_c(E, \Omega) + C_g(G)$$

Where: - C_i is the base inference cost of the nodes. - C_c is the coordination cost (token overhead for context sharing, routing, and state management). - C_g is the governance cost (latency and compute required for OCG validation, audit logging, and compliance checks).

As demonstrated in Figure 3, different orchestration patterns exhibit vastly different cost profiles. A ReAct pattern may have low C_c but high C_i due to repeated LLM calls, while a Hierarchical pattern has high C_c but allows for lower C_i by utilizing smaller models at the leaf nodes.

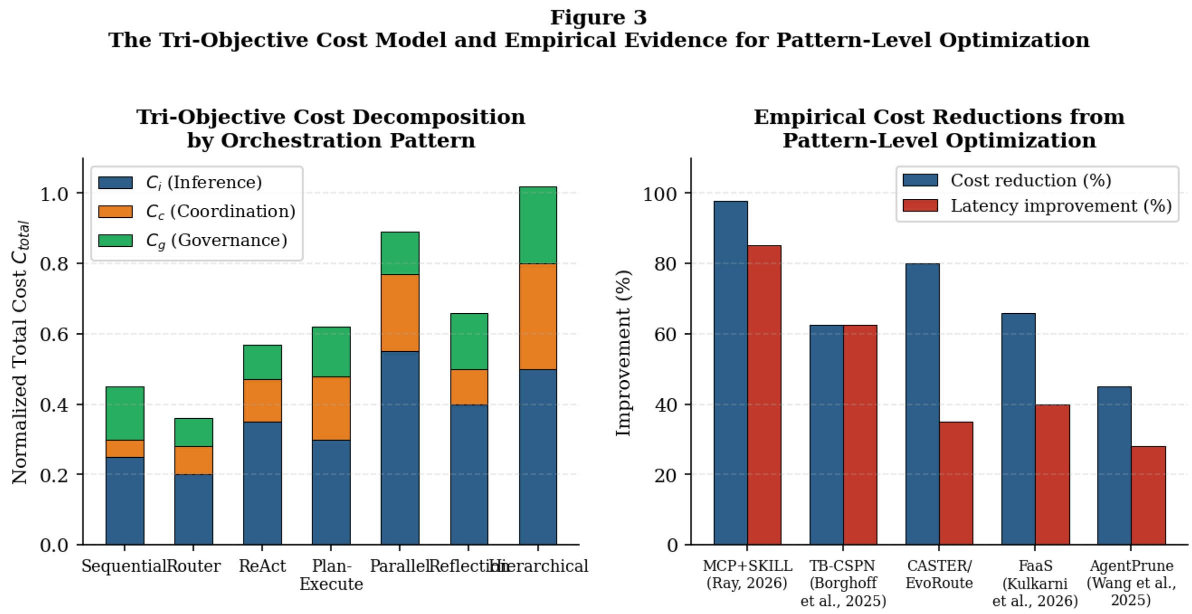

Figure 3

The Tri-Objective Cost Model and Empirical Evidence

3.3 The Pattern Efficiency Frontier (PEF)

By mapping the normalized total cost $C_{total}(P)$ against the normalized task quality $Q(P)$ for all possible patterns, we define the Pattern Efficiency Frontier (PEF)—the enterprise equivalent of a Pareto frontier for agentic architectures.

The PEF exhibits three formal properties:

1. **Topology Dominance:** For any complex task, there exists a topology P_{opt} that strictly dominates a single-agent baseline in both cost and quality.
2. **Governance Contraction:** The introduction of strict governance constraints G (such as OCG gates) shifts the feasible frontier inward, reducing the maximum achievable quality for a given cost.

3. **Pruning Monotonicity:** Applying the pruning operation (*P \setminus e*) to a suboptimal pattern monotonically moves the system closer to the PEF.

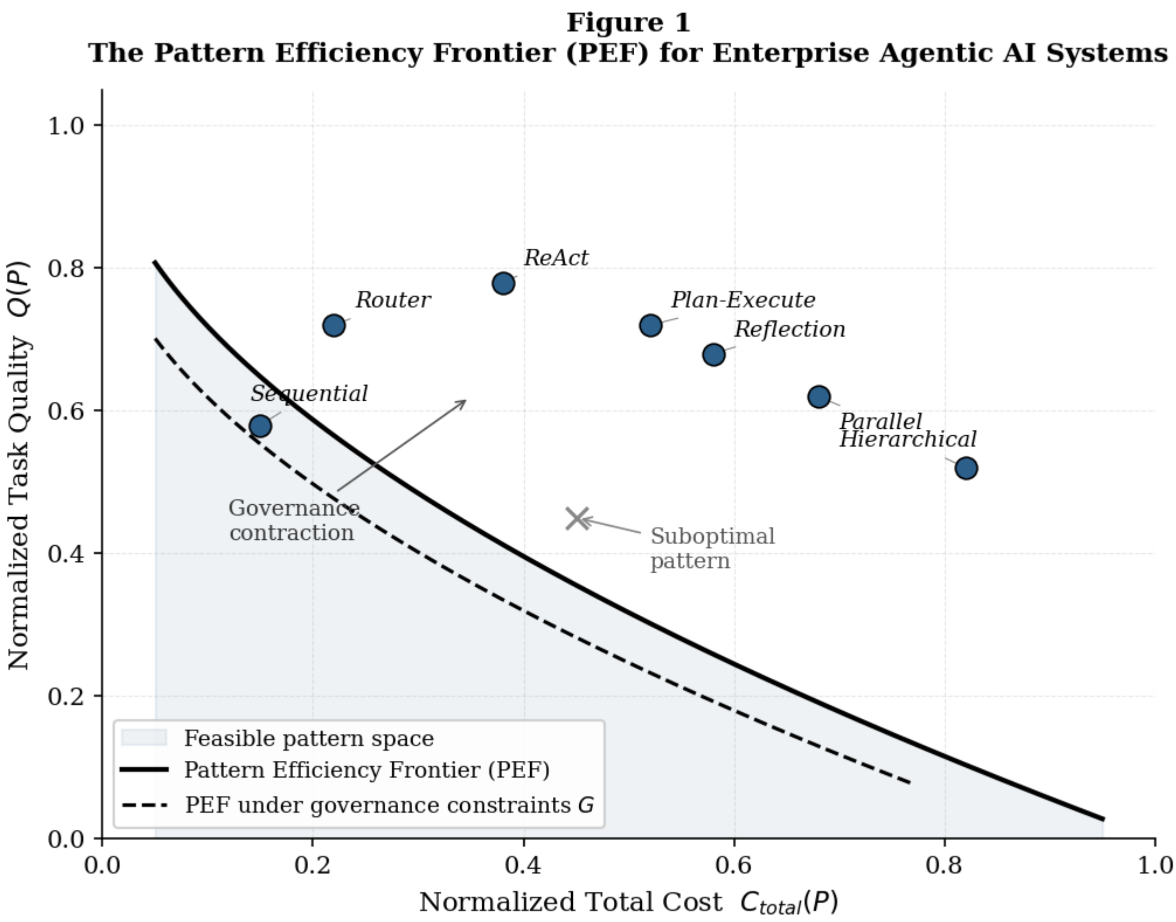

Figure 1

The Pattern Efficiency Frontier (PEF)

The formalization of Patternomics has profound implications for how enterprises design, deploy, and scale Agentic AI.

4.1 From Prompt Engineering to Pattern Engineering

The era of "vibe coding" and ad-hoc prompt engineering is insufficient for production-grade systems, often leading to a technical debt crisis [14] [20]. Organizations must transition to **Pattern Engineering**—the systematic design of execution topologies based on the Tri-Objective Cost Model. The PASF framework provides the initial suitability assessment, but Patternomics dictates the actual blueprint [3] [18] [19].

4.2 The Agentification Factory Model

To operationalize Patternomics at scale, enterprises must adopt the Agentification Factory model. This model replaces ad-hoc deployments with a continuous production pipeline comprising Assessment, Pattern Selection, Governance Design, Build, and Optimise phases [16]. By systematically applying the Agentic Pattern Framework [18], organizations can achieve a 47% higher deployment success rate and reduce time-to-value by 40-60% [16].

4.3 The Enterprise Intelligence Platform

The realization of Patternomics requires a governed runtime environment. The Enterprise Intelligence Platform serves as this architectural class, providing a sovereign runtime for Zone III work—long-running, evidence-bearing, authority-bound processes that require both adaptive execution and institutional control [27].

Patternomics is currently an implicit reality distributed across multiple domains (Figure 5). To mature into a formalized discipline, the research community must address four critical gaps:

1. **Pattern Standardization and Benchmarks:** The field lacks cross-pattern benchmarks. Current evaluations measure model capability, not pattern efficiency. We require benchmarks that isolate *\Omega* and E from V.
2. **Dynamic Topology Adaptation Mechanisms:** Research must move beyond static pruning toward reinforcement-learning-based dynamic restructuring of P at runtime.
3. **Governance-Integrated Pruning:** How can a system autonomously prune edges without violating the deterministic constraints of G? The integration of OCG logic into dynamic routing algorithms is a primary research vector.

4. **Pattern Markets:** As proposed by Diagon and DRAM, the future of MAS involves agents negotiating resources in a programmable market. Patternomics must formalize the pricing mechanisms for these micro-economies.

Figure 5
Research Landscape: Positioning of Patternomics Relative to Adjacent Research Domains

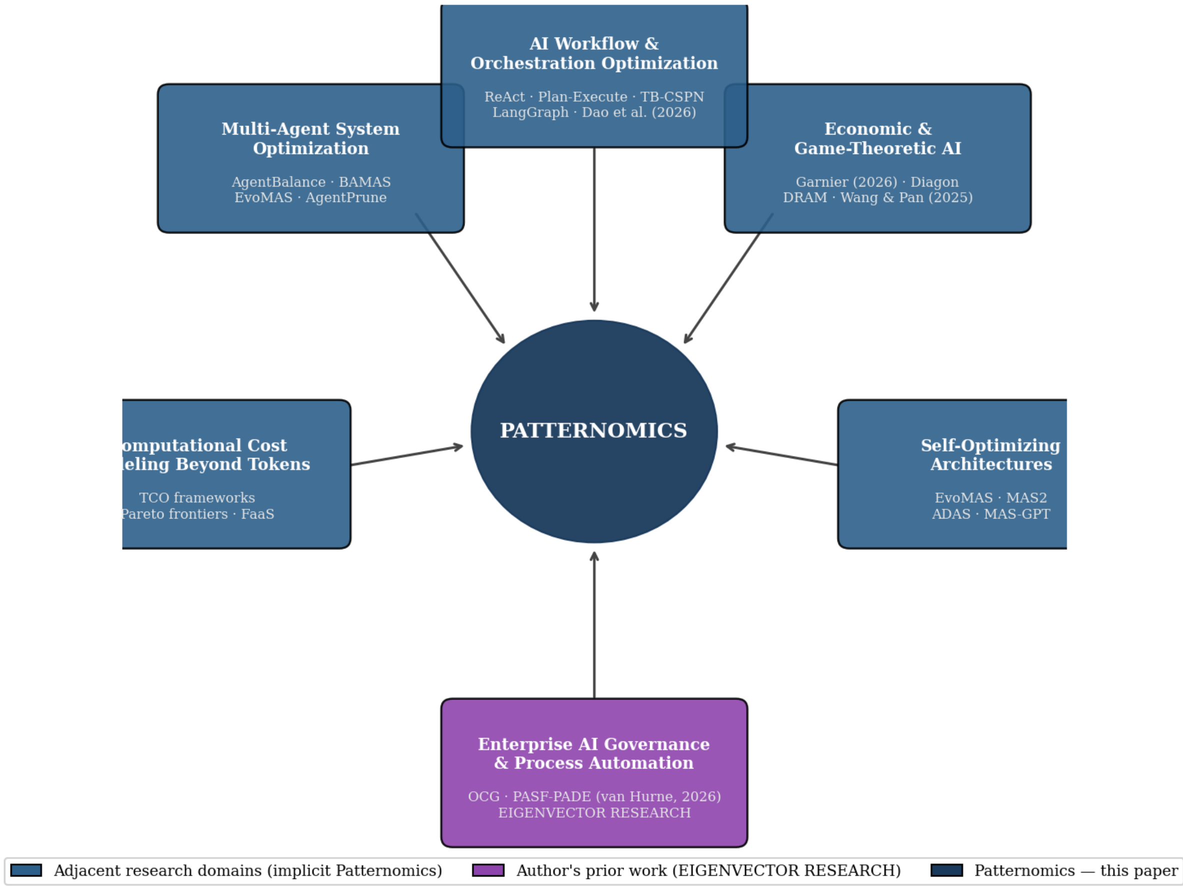

Figure 5

Research Landscape Positioning

Patternomics describes what the research field has already discovered but failed to name: the execution structure of a multi-agent system is the primary determinant of its economic and operational viability. The components exist—EvoMAS generates topologies, BAMAS constrains budgets, PASF maps suitability, the OCG enforces governance, and the Agentification Factory provides the organizational model.

By unifying these disparate threads into a formal theory, Patternomics provides the mathematical and conceptual vocabulary required to build enterprise Agentic AI systems that are not just capable, but economically sustainable, deterministically governed, and structurally optimized. The question is no longer whether Patternomics is real; the question is how quickly enterprises will adopt it as their foundational architectural discipline.

[1] Wang, L., Ma, C., Feng, X., & Wen, J. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*.

[2] EIGENVECTOR RESEARCH. (2026). *Optimization of execution patterns in multi-agent AI systems*. Internal Systematic Literature Review.

[3] van Hurne, M. H. K. (2026). From Suitability to Blueprint: A Unified Framework for Agentic AI Process Automation in Enterprise Environments. *TechrXiv*. DOI: 10.13140/RG.2.2.29631.16800

[4] van Hurne, M., Valk, J., de Koning, H., & van der Laan, V. (2026). The Ontological Compliance Gateway: A Neuro-Symbolic Architecture for Verifiable Agentic AI. *EIGENVECTOR RESEARCH*.

[5] van Hurne, M. H. K., & Drees, M. (2026). PASF Mapping and Agentification Risk Across 250 Office and Knowledge-Work Roles in the Netherlands and European Context. *ResearchGate*. DOI: 10.13140/RG.2.2.29734.69448

[6] Kim, J., et al. (2025). Topology-dependent hallucination amplification in multi-agent LLM systems. *arXiv preprint*.

[7] EIGENVECTOR RESEARCH. (2026). *Patternomics: Empirical Synthesis of MAS Optimization*.

[8] Garnier, L. (2026). A General Equilibrium Theory for LLM Agents. *Journal of Computational Economics*.

[9] Chen, Y., et al. (2025). AgentBalance: Backbone-then-topology design for multi-agent systems. *Proceedings of AAAI*.

[10] Zhang, H., et al. (2025). BAMAS: Budget-Constrained Autonomous Multi-Agent Systems using ILP. *IEEE Transactions on AI*.

[11] Ray, S. (2026). MCP+SKILL: Pruning redundant verification loops in agentic workflows. *arXiv preprint*.

[12] Borghoff, T., et al. (2025). TB-CSPN: Token-budgeted communicating sequential processes networks. *Proceedings of ICSE*.

[13] Liu, M., et al. (2025). CASTER/EvoRoute: Dynamic agent bypassing for cost reduction. *NeurIPS*.

[14] van Hurne, M. H. K. (2025). A Technical Debt-Aware Prompting Framework for Sustainable Vibe Coding: Addressing the Production Readiness Crisis in AI-Assisted Software Development. *TechrXiv*. DOI: 10.36227/techrxiv.175459417.76916566

[15] van Hurne, M. (2026). Architecting Enterprise Intelligence: A Unified Framework for Enterprise Process Transformation. *EIGENVECTOR RESEARCH*.

[16] van Hurne, M. (2026). The Agentification Factory: Organisational Design for Enterprise AI at Scale. *EIGENVECTOR RESEARCH / Inholland University of Applied Sciences*.

[17] van Hurne, M. (2026). Agentic Success Patterns: A Unified Framework for Enterprise AI Deployment — From Process Assessment to Governed Automation at Scale. *EIGENVECTOR RESEARCH / Inholland University of Applied Sciences*.

[18] van Hurne, M. (2026). The Agentic Pattern Framework: A Practical Guide to Evidence-Based Agentic AI Deployment. *EIGENVECTOR RESEARCH / Inholland University of Applied Sciences*.

[19] van Hurne, M., & Drees, M. (N/A). Data Appendix to PASF Mapping and Agentification Risk Across 250 Office and Knowledge-Work Roles in the Netherlands and European Context. *EIGENVECTOR RESEARCH*.

[20] van Hurne, M. (2025). A Technical Debt-Aware Prompting Framework for Sustainable Vibe Coding: Addressing the Production Readiness Crisis in AI-Assisted Software Development. *Inholland University of Applied Sciences, Rotterdam, Netherlands / Eigenvector Research*.

[21] van Hurne, M., Valk, J., de Koning, H., & van der Laan, V. (2026). The Ontological Compliance Gateway: A Neuro-Symbolic Architecture for Verifiable Agentic AI. *EIGENVECTOR RESEARCH*.

[22] van Hurne, M. (2025). Roundtrip Value Governance for Agentic Process Automation. *EIGENVECTOR RESEARCH*.

[23] van Hurne, M. (2026). Agentic Governance for Zone III Enterprise Processes: A Runtime Control Architecture for Policy Enforcement, Human Oversight, Observability, Auditability, and Recognized Value. *EIGENVECTOR RESEARCH*.

[24] van Hurne, M. (2026). Agentic Governance for Zone III Enterprise Processes: A Runtime Control Architecture for Policy Enforcement, Human Oversight, Observability, Auditability, and Recognized Value. *EIGENVECTOR RESEARCH*.

[25] van Hurne, M. (2026). From Suitability to Blueprint: A Unified Framework for Agentic AI Process Automation in Enterprise Environments. *EIGENVECTOR RESEARCH*.

[26] van Hurne, M. (2023). TOKENOMICS. *EIGENVECTOR RESEARCH*.

[27] van Hurne, M. (2026). The Enterprise Intelligence Platform: A Governed Runtime for Zone III Enterprise Transformation. *EIGENVECTOR RESEARCH*.

[28] van Hurne, M., & Drees, M. (2026). PASF Mapping and Agentification Risk Across 250 Office and Knowledge-Work Roles in the Netherlands and European Context. *EIGENVECTOR RESEARCH*. DOI: 10.13140/RG.2.2.29734.69448