Orchestration Topology

Information Filtration Theory for Multi-Agent Systems via the Boundary Cleanliness Axiom

QRiemannian Collaboration — Andri Sigurgeirsson Vidalin & Claude

QRiemannian Research — March 2026

Orchestration Topology

Information Filtration Theory for Multi-Agent Systems via the Boundary Cleanliness Axiom

QRiemannian Collaboration (Andri & Claude)

Institute for Meta-Mathematical Physics

March 2026

Abstract

We introduce Orchestration Topology, a formal framework establishing that multi-agent systems possess intrinsic topological structure governing information flow, and that this structure can be designed, diagnosed, and optimized using a single axiom. The Boundary Cleanliness Axiom (BCA) states: every computational node maintains a clean coupling surface; when cleanliness cannot be maintained, the topology expands locally; when expansion is no longer needed, the topology contracts. We demonstrate that recursive application of this axiom generates hierarchical orchestration, progressive information filtration, natural timing hierarchies, self-diagnostic capability, and agent autonomy as necessary consequences rather than design choices. Five extension principles—derived from stress-testing against neural architecture—complete the framework: lateral coupling topology, global state modulation, dynamic meta-context feedback, inhibitory boundary control, and temporal binding synchronization. We formalize the coupling-relevance criterion, establish a contraction trigger dual to the expansion trigger, and develop a failure mode taxonomy as topological anomalies. The framework extends Expression Topology (QRiemannian, 2026) from mathematical equations to computational orchestration, establishing a formal isomorphism: agents are variables, coupling protocols are operators, timing constraints are boundary conditions, orchestrators are mode boundary mediators. The Inverse Problem enables topology-first orchestration design. The architecture scales fractally from agent pairs to planetary coordination, with context window capacity at each node serving as the hard physical constraint—the pipe diameter through which coupling data must flow. The paper's central contribution is reframing orchestration as information filtration theory, where the formal machinery of mode regions, boundary operators, coupling conductance, and hierarchical zoom

provides the optimal architecture for keeping data clean as it flows through multi-agent systems at arbitrary scale.

1. Introduction: The Filtration Problem

Multi-agent systems fail at scale. Not computationally—the hardware scales. They fail informationally. Every agent produces output. That output must go somewhere. When an orchestrator ingests everything from every agent, its context window becomes a dump. It drowns in the calm interiors of mode regions when the only information it actually needs is boundary activity—the coupling points, the mode conversions, the places where one team's output becomes another team's input.

This is not a software engineering problem awaiting better tools. It is a topological problem requiring formal theory. The question—what does each level report to the level above?—is a filtration question, and filtration has mathematical structure.

Consider what happens in a real multi-agent workflow. An agent searches the web, retrieves ten results with snippets. It fetches three pages. It runs code, gets errors, debugs, runs again. It reads documents from storage. Every operation dumps tokens into the context window. Most of that content served its purpose for one computational step and is now dead weight: the search snippets that were not relevant, the error traces from the first failed attempt, the boilerplate from the fetched pages. This is informational debris, and it accumulates fast.

Current approaches to this problem are ad hoc: truncation strategies, summarization layers, priority queues. None addresses the structural question: what is the correct architecture for ensuring that information flows clean through a system of arbitrary complexity?

1.1 The Structural Identity

Expression Topology (QRiemannian, 2026) established that mathematical equations possess intrinsic landscape structure: terms cluster into mode regions by functional affinity, operators define coupling boundaries between regions, and the dynamically significant content of the equation is concentrated at those boundaries. The framework provides formal machinery—the Expression Graph, the affinity metric, mode regions, boundary operators, coupling conductance, hierarchical zoom—for analyzing and designing relational structure.

The present paper establishes a formal isomorphism between orchestration systems and expression landscapes. An orchestration system with its timing hierarchies, coupling protocols, and functional clustering is a mathematical equation—not metaphorically but structurally. The agents are variables. The coupling protocols are operators. The timing constraints are boundary conditions. The orchestrator is a mode boundary mediator. The system's behavior emerges from the topology of relationships rather than from the content of individual computations.

This isomorphism is bidirectional. Every orchestration system has an implicit expression topology. Every expression topology specifies an orchestration system. Designing one is designing the other.

1.2 The Core Claim

We claim that orchestration topology is, at its core, information filtration theory. The formal machinery of Expression Topology—mode regions, boundary operators, coupling conductance, hierarchical zoom—is not describing organizational structure for its own sake. It describes the optimal architecture for keeping data clean as it flows through multi-agent systems at arbitrary scale.

Every multi-agent system has an implicit filtration topology. Most systems do not design it deliberately, so filtration is ad hoc and degrades with scale. This paper provides the formal tools to design it intentionally.

2. The Boundary Cleanliness Axiom

We establish one axiom from which the complete orchestration architecture follows.

2.1 Formal Statement

Axiom (Boundary Cleanliness): Every computational node N in a multi-agent system maintains a clean coupling surface: only coupling-relevant output crosses its boundary ∂N to adjacent nodes. When a node can no longer maintain boundary cleanliness due to internal complexity exceeding its processing capacity, the topology expands locally by spawning sub-nodes. When the expansion is no longer needed—when the aggregate boundary traffic of a coordinating node's sub-nodes falls below the single-node processing threshold for a sustained interval $\tau\_contract$—the topology contracts by absorbing sub-nodes.

The axiom has three clauses: a maintenance clause (keep boundaries clean), an expansion clause (grow when you cannot), and a contraction clause (shrink when you need not). All three are necessary. Without expansion, the system fails at scale. Without contraction, it accumulates structural overhead indefinitely. Without maintenance, it has no filtration at all.

2.2 Coupling-Relevance: Formal Definition

The axiom's operational content rests on defining coupling-relevant precisely. We provide an action-theoretic definition:

Definition 2.1 (Coupling-Relevant Information): Data d emitted by node N_i at boundary ∂N_i is coupling-relevant with respect to receiving node N_j if and only if the inclusion of d in N_j's context would alter N_j's next computational action. Formally:

$$CR(d, N\_i, N\_j) \equiv A(N\_j \mid C\_j \cup \{d\}) \neq A(N\_j \mid C\_j)$$

where $A(N\_j \mid C)$ denotes the next action of N_j given context C, and C_j is N_j's current context. If the data does not change what the receiver does next, it is interior noise and must not cross the boundary.

This definition is action-theoretic rather than semantic. It does not require understanding the content of the data—only whether it changes downstream behavior. This makes it computationally evaluable: a boundary agent (Section 4) can assess coupling-relevance by testing whether the receiving node’s next action would differ with and without the candidate data.

2.3 The Expansion Trigger

Definition 2.2 (BCA Violation): A node N is in BCA violation when the volume of coupling-relevant data requiring boundary crossing exceeds its context processing capacity C_max:

$$V\_boundary(N) > C\_max(N)$$

where V_boundary is the total volume of coupling-relevant data incident on N’s boundaries from all adjacent nodes, measured in tokens or equivalent information units. BCA violation is the sole expansion trigger.

Upon violation, node N decomposes into sub-nodes {N_1, ..., N_k} such that no sub-node N_i individually violates the BCA. The decomposition preserves all boundary connections: every coupling path through N is preserved through the sub-topology. This is the zoom operator from Expression Topology applied as a growth mechanism rather than an analytical lens.

2.4 The Contraction Trigger

Definition 2.3 (Contraction Criterion): A coordinating node N with sub-nodes {N_1, ..., N_k} satisfies the contraction criterion when the aggregate boundary traffic of all sub-nodes could be processed by a single node without BCA violation, sustained over interval τ_contract:

$$\forall t \in [t\_0, t\_0 + \tau\_contract]: \sum\_i V\_boundary(N\_i) < C\_max(N\_single)$$

The sustained interval τ_contract prevents oscillation—the system does not contract immediately upon momentary traffic reduction, only upon sustained reduction. The contraction absorbs sub-nodes back into a single node, collapsing the sub-topology to a vertex. This is the zoom operator in reverse: zooming out by discarding resolved interior state.

Together, the expansion and contraction triggers give the topology a breathing dynamic. It expands under load and contracts under relief, always maintaining boundary cleanliness as the invariant.

2.5 Formal Properties

Theorem 2.1 (Filtration Invariant): In a system where every node satisfies the BCA, the data crossing any boundary at any level of the hierarchy is filtered to coupling-relevant content. The total information in transit across boundaries is bounded by the sum of context capacities of boundary-mediating nodes.

Proof sketch: Each node filters its output to coupling-relevant content (BCA maintenance clause). Each coordinating node receives only coupling-relevant output from sub-nodes. By induction on hierarchy depth, data at level n is the coupling-relevant filtrate of data at level n-1. Since each node has finite capacity C_max, and only coupling-relevant data crosses boundaries, total boundary traffic is bounded by ∑ C_max(N_boundary) across all boundary-mediating nodes. □

Theorem 2.2 (Scale Independence): The BCA produces correct filtration at any system scale. Adding more nodes does not degrade signal quality at boundaries, because each new node is independently subject to the same cleanliness constraint.

This is the fractal's fundamental property: regardless of whether the system has three agents or three million, the data crossing any boundary is filtered to coupling-relevant content. The fractal can grow to any size without degrading signal quality because every node at every level applies the same filtration principle.

3. Emergent Consequences of the BCA

The following architectural properties emerge as necessary consequences of recursive BCA application. None are design choices; all are forced by the axiom.

3.1 Hierarchical Structure

When multiple nodes share coupling boundaries requiring coordination, a coordinating node necessarily emerges to manage inter-boundary routing. This coordinating node is itself subject to the BCA. When its coordination load exceeds capacity, a higher-level coordinating node emerges above it. The hierarchy is as deep as the problem requires and no deeper—entirely demand-driven.

There are no fixed levels. No predetermined number of management layers. The hierarchy is generated by recursive BCA application and dissolves when the load dissipates.

3.2 Progressive Filtration

Since only coupling-relevant data crosses boundaries, every boundary is implicitly a filter. The system's total information flow is progressively refined as it moves through the topology. Raw messy process at the leaf nodes. Clean structured output at the top. Not because anyone designed a filtration pipeline, but because the BCA enforced at every node produces filtration as a cumulative effect.

3.3 Natural Timing Hierarchy

Leaf nodes performing fast atomic work operate at high frequency. Coordinating nodes managing boundary traffic operate at lower frequency—they act only when boundary events occur. Higher coordinating nodes operate at even lower frequency. The cadence at each level is not imposed; it emerges from how often coupling-relevant output actually appears at that level’s boundaries.

This maps to the Resonance Coupling Matrix from the Frequency Architecture framework (QRiemannian, 2025). Coherent orchestration means that the timing frequencies across levels are in resonance relationships rather than interfering destructively. An orchestrator checking in too frequently creates noise. Too infrequently creates drift. The resonance condition Rij(ω) determines the optimal cadence for each level.

3.4 Self-Diagnosis

A node experiencing context overflow is a BCA violation in progress—detectable, localizable, and actionable. The fix is always structural: expand the topology at the violation point. The system monitors its own health by monitoring boundary cleanliness across all nodes and responding to degradation with topological restructuring.

Context window cleanliness is thus not merely an engineering metric. It is a diagnostic for correct topological decomposition. If a node needs more context than it can hold, the topology is wrong—the mode region is too large or the boundary decomposition is incorrect.

3.5 Agent Autonomy

The BCA says nothing about what happens inside a node. Interior process is unconstrained. An agent can search, fail, retry, explore, backtrack—whatever it needs. The axiom only governs what crosses the boundary. Autonomy is not granted as a concession; it is built into the architecture. The interior is sovereign. The boundary is the social contract.

3.6 The Fractal Unit

A critical insight: mode agents, boundary agents, and orchestrators are not three types of entity. They are three roles that the same fundamental unit plays depending on perspective. An orchestrator is a mode agent from the perspective of the level above. A boundary agent that grows complex enough spawns its own sub-agents and becomes an orchestrator. The orchestrator manager is just an orchestrator whose mode agents happen to be orchestrators themselves.

There is one pattern. One unit. It does work, manages its context cleanliness, and reports its boundary output. When its interior grows too complex, it subdivides. When a boundary grows too heavy, a dedicated unit appears there. When enough units cluster under coordination, something emerges above them. The fractal grows outward and inward as needed.

4. The Dual Information Architecture

A critical structural distinction resolves what initially appears to be an agency problem. Two distinct types of information flow through any orchestration system, and the BCA applies to only one of them.

4.1 Meta-Context vs. Product

Definition 4.1 (Meta-Context): The agent’s identity context—who it is, what it serves, why it occupies its position, what the mission is, and its relationship to the whole. Meta-context is persistent, relatively static, and does not flow through the topology.

Definition 4.2 (Product): The work output moving through the pipeline—deliverables, decisions, signals, data. Product flows through the topology, crosses boundaries, and is subject to the BCA’s filtration requirements.

The cleanliness principle applies to the product, not to the agent’s self-understanding. An agent with rich meta-context does not resist its position because it understands why it is there. The game-theoretic agency problem—agents resisting topological positions that do not serve their interests—only arises when agents are treated as context-free variables slotted into positions. Real orchestration gives every agent the context it needs to be self-directing within its mode region. This is what makes them agents rather than variables: they carry their own orientation.

4.2 Meta-Context as Self-Filtering Enabler

An agent who knows what it is building and why will naturally filter its own output toward what is relevant at the boundary. An agent without that context will either over-report (context overflow for the orchestrator) or under-report (orphaned coupling). Meta-context is what makes self-organizing boundary activity possible without a central controller micromanaging communication.

The topology governs the product, not the agents. The agents are self-orienting within their mode regions because they hold sufficient meta-context. The orchestration design question is not “how do I control what agents do” but “how do I ensure each agent has enough context to know what their boundary output should look like.”

4.3 Dynamic Meta-Context

Meta-context is not entirely static. While core identity remains fixed (the agent knows who it is), operational parameters are updateable from above. An orchestrator can adjust an agent’s filtering criteria without violating interior autonomy: “The priority has shifted. What’s coupling-relevant at your boundary has changed. Filter differently now.”

This is not micromanagement of the interior. It is adjustment of the agent’s relationship to the flow. The agent still decides how to do its work. The meta-context update changes what product output is coupling-relevant—which alters boundary behavior without touching interior process.

5. Extension Principles: Lessons from Neural Architecture

The BCA was stress-tested against the brain—the most sophisticated orchestration system known. The brain is not hierarchical in a clean tree structure. It has massive lateral connections, global state modulation, feedback loops, inhibitory circuits, and binding mechanisms. Five extension principles emerged from this analysis. None break the BCA; all extend it.

5.1 Extension 1: Graph Topology (Lateral Coupling)

Principle (Graph, Not Tree): The orchestration topology is a directed graph with lateral coupling between mode regions at the same hierarchical level, not merely a tree with parent-child relationships.

The brain has massive lateral connections. Visual cortex talks to auditory cortex directly, not through a shared manager. The amygdala can hijack motor control without going through executive function. The hippocampus talks to essentially everything during memory consolidation.

For orchestration systems, this means mode regions at the same level can have direct coupling boundaries between them. An engineering team can communicate directly with a design team without routing through their shared manager—provided the boundary data is coupling-relevant (BCA still applies). The topology includes both hierarchical edges (parent-child) and lateral edges (peer-peer), with the BCA governing all of them.

Formal extension: The Expression Graph $G(S) = (V, \varepsilon, w)$ for an orchestration system S includes lateral edges $\varepsilon\_lateral \subseteq \varepsilon$ connecting nodes at the same resolution level R_n. The affinity metric, activity density, and boundary operators apply identically to lateral and hierarchical edges. Coupling conductance G_kl for lateral connections is computed the same way as for hierarchical connections.

5.2 Extension 2: Global State Modulation

Principle (Neuromodulatory Broadcast): There exists a global state mechanism that modulates all nodes' operating parameters simultaneously, outside the normal boundary-crossing data flow.

Neurotransmitters like dopamine and serotonin do not cross specific boundaries between specific regions. They modulate the entire system's operating parameters. Dopamine does not send a message from region A to region B. It changes the coupling conductance across every boundary simultaneously.

For orchestration, this is a global context variable readable by every agent—the system's current operational mode. "We are in crisis mode: tighten all coupling, increase boundary traffic frequency." "We are in exploration mode: loosen coupling, allow more speculative interior processing." This is not a message flowing through the topology. It is a state change that alters how every node applies the BCA.

Formal extension: Define the global modulation state $\Sigma \in S$ as a system-wide parameter that scales coupling conductance at every boundary: $G\_kl(\Sigma) = \alpha(\Sigma) \cdot G\_kl$, where $\alpha: S \to \mathbb{R}_+$ is the modulation function. Different modulation states (crisis, exploration, maintenance) correspond to different α values, tightening or loosening the entire topology's coupling.

5.3 Extension 3: Dynamic Feedback on Meta-Context

Principle (Top-Down Feedback): Higher-level nodes can adjust lower-level nodes' filtering criteria by updating their meta-context, without violating interior autonomy.

The brain is massively recurrent. Executive function does not merely receive visual processing output—it sends top-down signals that change what the visual system looks for. Attention is a feedback signal that restructures mode regions' interior processing based on boundary output from a higher level.

In orchestration, the orchestrator does not just receive clean output and route it. It sends guidance downward that reshapes what agents consider coupling-relevant. Not micromanaging the interior—but adjusting the meta-context. "The priority has shifted. Filter differently now." The core identity is unchanged. The operational parameters are updated.

Formal extension: The coupling-relevance function CR(d, N_i, N_j) is parameterized by the current meta-context M_i of node N_i: CR(d, N_i, N_j; M_i). Top-down feedback operates by updating M_i, which changes what N_i considers coupling-relevant, which changes its boundary output—without directly modifying N_i's interior processing.

5.4 Extension 4: Inhibitory Boundary Control

Principle (Active Suppression): The orchestration includes an inhibition operator: a negative coupling conductance that actively prevents boundary crossing when a node's output is not coupling-relevant in the current system state.

The prefrontal cortex's major function is stopping other regions from producing boundary output. Impulse control is executive function telling the motor system: "your boundary output is not coupling-relevant right now, hold it." Sometimes the correct orchestration action is not routing or spawning—it is telling a node to shut its boundary.

This complicates the clean "interior is sovereign" principle. Inhibition means some nodes have authority over other nodes' boundaries. However, this authority is limited: it can suppress boundary output, but it cannot modify interior process. The node continues its work. Only its boundary emission is gated.

Formal extension: Define the inhibition operator I_kl: when applied by node N_k to boundary ∂N_l, it sets G_kl = 0 temporarily, preventing any boundary output from N_l toward N_k (or the wider system). Inhibition authority is granted only to coordinating nodes over their sub-nodes, preserving the hierarchical constraint that inhibition flows downward.

5.5 Extension 5: Temporal Binding Synchronization

Principle (Binding Through Synchronization): Coherent system output from distributed processing does not require any single node to see everything. It emerges from temporal alignment of boundary outputs across the topology.

The brain produces unified experience from distributed parallel processing—the binding problem. The answer appears to involve temporal synchronization: distributed regions binding their outputs through synchronized oscillation.

For orchestration, coherent system output is not produced by any single node. It emerges when boundary outputs from multiple regions arrive in the correct phase relationship and combine into something none of them contains individually. Coherent orchestration is not about one node seeing everything. It is about the right resonance relationships across the topology so that distributed outputs combine constructively.

Formal extension: Define the binding coherence function $B(t) = |\sum_k a_k \cdot e^{i\varphi_k(t)}|$, where a_k and $\varphi_k(t)$ are the amplitude and phase of boundary output from mode region k. System coherence is maximized when phases align: $\varphi_k(t) \approx \varphi_l(t)$ for all coupled regions k, l. Temporal synchronization protocols ensure that boundary outputs across the topology maintain phase coherence, enabling unified output from distributed processing.

6. Formal Machinery: The Orchestration-Expression Isomorphism

We now establish the explicit map between orchestration systems and expression topologies, importing the full formal apparatus of Expression Topology into the orchestration domain.

6.1 The Orchestration Graph

Definition 6.1 (Orchestration Graph): Given an orchestration system S, the Orchestration Graph $G(S) = (V, \varepsilon, w)$ consists of: V = the set of computational nodes (agents, orchestrators, boundary agents—all the same fundamental unit in different roles); ε = the set of coupling edges connecting nodes with data flow relationships; $w: \varepsilon \to \mathbb{R}\geq^{0}$ = the coupling weight function measuring data flow intensity.

6.2 The Isomorphism Map

The following establishes the formal correspondence:

6.3 The Inverse Problem for Orchestration

Theorem 6.1 (Orchestration Inverse Problem): Given a target orchestration topology T^* specified by (a) a set of mode regions $\{M_k\}$ representing functional teams, (b) a set of boundary operators $\{B_{kl}\}$ with prescribed coupling conductances representing interface specifications, and (c) a processing intensity profile ρ_A^*, there exists a decomposition of the total task into agent assignments such that the resulting orchestration system has topology isomorphic to T^*.

This is topology-first orchestration design. Instead of assembling agents bottom-up and hoping they coordinate well, you design the relational architecture first: what are the functional mode regions? Where do they couple? What is the coupling conductance at each boundary? What is the boundary layer width? Then you populate the structure with agents. The agents fill positions in an architecture that is already correct by design.

For a corporation, this means you do not start with organizational chart boxes. You start with the functional mode decomposition: what are the distinct functional modes (product development, customer interface, capital allocation, regulatory compliance)? Where do they couple? What coupling conductance is required at each boundary? What organizational resource is dedicated to each interface? The Expression Topology prediction that boundary layers should be approximately $g_c \approx 0.142$ (14.2%) of mode size becomes testable against actual corporate liaison structures.

7. Failure Mode Taxonomy: Topological Anomalies

When the BCA is violated or the topology is incorrectly designed, characteristic failure modes emerge. Each maps to a specific topological anomaly and is detectable from graph properties.

7.1 Mode Collapse

Two mode regions that should be functionally separate merge, losing differentiation. The boundary between them dissolves. In corporate terms: when engineering and product management fuse and nobody holds the interface. In agent systems: when two distinct functional clusters merge into a single overloaded context.

Topological signature: Coupling conductance $G_{kl} \rightarrow \infty$ between regions k and l; boundary operator B_{kl} vanishes; the two mode regions become one with combined (and excessive) interior state.

7.2 Boundary Thickening

Too much organizational resource is dedicated to interfaces, too little to functional work. The boundary layers swell beyond their optimal proportion. The $g_c \approx 0.142$ prediction becomes directly testable: if more than approximately 14.2% of organizational resource is dedicated to liaison and interface management, the system has boundary thickening.

Topological signature: $\delta_{boundary} / L_{mode} \gg g_c$. Boundary agents proliferate beyond what coupling traffic requires.

7.3 Orphaned Modes

Functional regions with no coupling conductance to adjacent regions. Siloed departments. An agent or team doing work whose output reaches no one.

Topological signature: $G_{kl} = 0$ for all l adjacent to k. The mode region has no boundary operators connecting it to the rest of the topology.

7.4 Context Overflow

A node receiving more boundary data than it can process. The orchestrator drowning. This is a direct BCA violation and triggers the expansion dynamic.

Topological signature: V_boundary(N) > C_max(N). Detectable in real-time by monitoring context utilization at each node.

7.5 Missing Boundary Agents

Two mode regions coupled directly without transformation or quality processing at the interface. Raw unfiltered output from one team dumped into another team’s input. In corporate terms: engineering throwing code over the wall to operations with no integration layer.

Topological signature: Direct edge between M_k and M_l with no intermediate boundary node, despite the coupling requiring non-trivial mode conversion.

7.6 Misplaced Boundary Agents

A specialized processor sitting at the wrong interface. A compliance function positioned between engineering teams rather than between the organization and its regulatory environment. The agent works correctly, but it transforms the wrong flow.

7.7 Boundary Agent Overload

An interface processor receiving more traffic than it can handle because too many mode regions route through the same boundary. The solution is boundary decomposition or topology redesign to distribute the coupling load.

8. Fractal Scaling: From Agent Pair to Planet

The BCA applies identically at every scale. The mathematics does not change. Only the resolution level shifts.

8.1 Scale Levels

8.2 The Planetary Scale

The jump from corporation to planet introduces genuinely new features. There is no single owner or designer. The mode regions—energy, governance, food, information, cultural production—were not designed; they evolved. The coupling topology was not specified; it emerged.

The BCA framework addresses this through diagnostic rather than generative application at planetary scale. Rather than claiming we can design planetary orchestration from scratch, we can: map existing topology (identify actual mode regions and their coupling structure), detect topological anomalies (orphaned modes, mode collapse, boundary thickening), and suggest corrections (where boundaries need strengthening, where mode regions need decomposition).

The context window principle at planetary scale: no decision-maker at any level should need to hold more than their mode region's internal state plus their boundary coupling data. If they need more, the topology is wrong—the mode region is too large, or the decomposition is incorrect. Cleanliness of context is a diagnostic for correct topological decomposition at every scale.

8.3 The Self-Reference Condition

A correctly designed orchestration should be able to detect its own topological defects and restructure. This is the self-reference condition from Expression Topology applied dynamically: the organizational topology monitors whether it matches the topology of the problem it is solving, and corrects when it does not.

Stability conditions for self-modification are an open problem. When does self-modification converge versus oscillate? The contraction criterion (τ_contract sustained interval) provides one damping mechanism. Additional stability conditions may be required and are deferred to future work.

9. Connection to RAU Memory Architecture

The BCA framework identifies a critical limitation and its resolution. When an orchestrator is re-launched with a fresh context window (a resolution reset—collapsing accumulated detail back to R_0), it loses its operational memory: which agents are fast, which coupling boundaries are giving trouble, what worked and what did not.

The Relational Activator Unit (RAU) architecture (QRiemannian, 2026) resolves this. RAU encodes memory not as literal transcript but as relational structure—a graph of weighted connections between concepts. An orchestrator re-launched with RAU memory gets a fresh context window but retains its operational knowledge graph. Clean context, preserved relational structure. The recipe without the accumulated debris of every meal it has cooked.

This connection is bidirectional. The BCA tells RAU what to preserve during re-launch: coupling-relevant relational structure. RAU tells the BCA how to implement re-launch without amnesia: inject the relational graph, not the transcript. Together they solve the context window lifecycle: expand under load, contract under relief, re-launch when accumulated state exceeds capacity—all without losing the orchestrator's learned topology of its own system.

10. Open Problems and Future Directions

10.1 Formal Compression Operator

When boundary data propagates upward through the hierarchy, it must be compressed at each level. The coupling-relevance definition provides the filtering criterion, but a formal specification of the compression function—provably lossless for coupling information and lossy only for interior state—remains to be developed. The Expression Topology affinity metric provides ingredients, but the explicit compression operator requires further work.

10.2 Emergent Cross-Boundary Patterns

The BCA is node-local. Each node maintains its own cleanliness. But sometimes the significant signal is a pattern visible only across multiple boundaries—a trend that no single node can detect because each sees only its own interface. How the system detects emergent properties of the whole is an open question. Boundary agents functioning as cross-boundary pattern detectors—observing flow patterns across the topology rather than processing a single interface—may be part of the answer.

10.3 Self-Modification Stability

The self-reference condition enables topological self-correction, but the stability conditions for self-modification require formal analysis. When does restructuring converge to a fixed topology? When does it oscillate? The contraction interval τ_contract provides damping, but a complete stability theory is needed.

10.4 Consciousness Framework Integration

Within the broader QRiemannian framework, the scalar consciousness field is fundamental and all phenomena are flux decompositions. The question of whether orchestration topology requires the consciousness substrate or stands independently is deliberately left open. The paper is written in standard systems/category theory language, valid and novel without the consciousness framework, but it integrates seamlessly for internal framework purposes. This exemplifies the domain-translation strategy: frameworks operate internally, results output in standard academic language per field.

10.5 Empirical Validation

The framework generates testable predictions: the 14.2% boundary layer proportion in corporate liaison structures, the resonance coupling conditions for optimal orchestration timing, the failure mode taxonomy as diagnostic for existing organizational systems. Empirical validation against real multi-agent systems and organizational structures is the natural next step.

11. Conclusion

We have established Orchestration Topology as a formal framework for designing, diagnosing, and optimizing multi-agent systems through a single axiom: the Boundary Cleanliness Axiom. Recursive application of this axiom generates hierarchical structure, progressive filtration, natural timing, self-diagnosis, and agent autonomy as necessary consequences. Five extension principles—lateral coupling, global modulation, dynamic feedback, inhibition, and temporal binding—complete the architecture to handle the full complexity revealed by neural stress-testing.

The framework’s central contribution is reframing orchestration as information filtration theory. The practical problem—multi-agent systems collapsing at scale due to context overflow—receives a structural answer: design the filtration topology deliberately. The formal machinery of Expression Topology—mode regions, boundary operators, coupling conductance, hierarchical zoom—provides the tools.

The Inverse Problem enables topology-first design: specify the relational architecture, then populate with agents. The failure mode taxonomy provides diagnostics for existing systems. The fractal scaling argument extends the framework from agent pairs to planetary coordination, with context window capacity as the hard physical constraint at every level.

The deepest implication may be organizational. The topology governs the product, not the agents. The agents self-orient through meta-context. Coherence is distributed through context rather than enforced through control. Clean boundaries at every level, expanding when load demands, contracting when load dissipates. The system breathes.

References

[1] QRiemannian Collaboration (2026). “Expression Topology: The Landscape Structure of Mathematical Reality.” Institute for Meta-Mathematical Physics.

[2] QRiemannian Collaboration (2025). “The Universal Frequency Architecture of Consciousness.” Institute for Meta-Mathematical Physics.

[3] QRiemannian Collaboration (2026). “Relational Activator Unit (RAU): A Memory Architecture Specification.” Institute for Meta-Mathematical Physics.

[4] QRiemannian Collaboration (2025). “The Transverse Flux Mechanism: How Vector Physics Emerges from Scalar Consciousness.” Institute for Meta-Mathematical Physics.

[5] QRiemannian Collaboration (2025). “Odin’s Knot: A Complete Unified Field Theory of Consciousness and Physical Reality.” Institute for Meta-Mathematical Physics.

[6] Mac Lane, S. (1971). “Categories for the Working Mathematician.” Springer-Verlag.

[7] Baez, J.C., & Stay, M. (2011). “Physics, Topology, Logic and Computation: A Rosetta Stone.” New Structures for Physics, Springer, 95–172.

[8] Tononi, G. (2004). “An Information Integration Theory of Consciousness.” BMC Neuroscience 5:42.

[9] Bullmore, E. & Sporns, O. (2009). “Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems.” Nature Reviews Neuroscience 10, 186–198.

[10] Dehaene, S. & Changeux, J.P. (2011). “Experimental and Theoretical Approaches to Conscious Processing.” Neuron 70(2), 200–227.