The Coherence Fee

Measuring Hidden Convention Risk in Hierarchical Tool Compositions

John Komkov
Independent Researcher
https://github.com/jkomkov/bulla

April 2026

Abstract

When workflows compose tools across typed boundaries, each bilateral handoff may type-check while hiding implicit convention mismatches—timezone formats, rounding modes, jurisdictional frameworks—that no pairwise validation can detect. We introduce the *coherence fee*, a computable integer counting convention dimensions whose consistency is structurally unverifiable from observable schemas alone, and the *minimum disclosure set*, exactly φ field disclosures that eliminate all blind spots.

A *hierarchical decomposition theorem* shows the fee splits into local sub-workflow fees plus a non-negative *boundary fee* measuring conventions hidden at delegation boundaries. A *tower law* proves this cost is additive across hierarchy levels—formalizing "more delegation increases hidden risk" as a theorem. An online resolution protocol supports incremental tool substitution with monotonically decreasing fee.

An 8-tool financial settlement case study demonstrates 2.1× savings in disclosure prescriptions over per-edge bridge suggestions. All claims are computationally verified across 10 bundled and 10,000 adversarial random compositions; the reference implementation requires zero external dependencies beyond PyYAML. A companion signed-incidence note [16] supplies the middle-layer structural boundaries used elsewhere in the corpus: the fee is field-independent as a numerical invariant, and pairwise same-dimension endpoint disclosures can couple rather than separate.

1 Introduction

A financial settlement pipeline passes a trade through eight typed boundaries: price fetch, currency conversion, compliance check, risk assessment, ledger posting, audit logging, notification, and archival. At each boundary, the tool validates its inputs and outputs against its declared schema. Every bilateral handoff type-checks. The system passes integration testing. Six months later, a regulatory audit discovers that the risk model was applied under a different jurisdictional framework than the compliance check assumed, because the `jurisdiction` convention was hidden inside the compliance tool's internal state and never exposed to the risk assessment tool. The coherence fee of this composition is 7. The tool we describe would have flagged it at design time and prescribed exactly which 7 fields to expose.

This failure is not an engineering oversight—it is a structural property of hierarchical composition. Schema validation catches type errors at each handoff, but conventions (timezone formats, rounding modes, regulatory frameworks, accounting standards) propagate through chains of tools via fields that the receiving tool never declares in its observable interface. These *blind spots* are invisible to any bilateral check. Worse, we prove that every level of delegation in a hierarchical tool composition adds non-negative hidden convention cost (theorem 4.5). This is not a bug that more careful engineering can fix; it is a theorem about the algebraic structure of hierarchical composition. Intuitively, schema validation certifies that each output is a well-formed *representation* of the declared type; it cannot certify the conventions under which the representation was produced, because those conventions were internal to the producing tool and absent from its observable interface.

Claim discipline. What is established here is the hierarchical fee and its decomposition through delegation boundaries. What failed productively in adjacent repair work was the stronger idea that dimension-aware disclosures would separate cleanly into one-per-dimension choices. What remains open is the right coupled-disclosure objective; the signed-incidence note [16] is the background citation for that boundary, not a result reproved in this paper.

Contributions.

1. **The coherence fee** (section 2): a computable integer $\varphi(G)$ measuring the number of convention dimensions structurally unverifiable from observable schemas.
2. **Minimum disclosure set** (section 6): a prescriptive list of exactly φ field disclosures sufficient to eliminate all blind spots, yielding 2.1× savings over per-edge bridge suggestions in our case study.
3. **Hierarchical decomposition with tower law** (section 4): the fee decomposes into local sub-workflow fees plus a non-negative boundary fee; the tower law shows boundary fees are additive across hierarchy levels, formalizing "delegation adds hidden cost."
4. **Online resolution** (section 6): an incremental protocol for substituting placeholder tools with monotonically decreasing fee.
5. **Negative results** (section 5): the boundary fee is monotone on the partition lattice but neither a valuation nor submodular—the latter disproved by an adversarial survey of 10,000 random compositions.
6. **Case study** (section 7): an 8-tool financial settlement pipeline with fee 7, boundary fee 2 under front/back-office partition, and conditional resolution demonstrating the full diagnostic loop.

Paper organization. Section 2 defines the composition model and coherence fee. Section 3 establishes basic properties through a counterexample. Section 4 proves the decomposition theorem, tower law, and extremal cases. Section 5 characterizes the boundary fee on the partition lattice (non-valuation, non-submodularity). Section 6 introduces the minimum disclosure set and online resolution protocol. Section 7 presents the financial settlement case study. Section 8 summarizes computational verification. Section 9 positions against related work, and section 10 concludes.

2 Model

Let $G = (V, E)$ be a composition graph where each node $v \in V$ is a tool with internal state I_v and observable schema $O_v \subseteq I_v$. Each edge $e \in E$ carries semantic dimensions, each specifying a `from_field` in the source tool and a `to_field` in the target tool.

The *coboundary operator* $\delta^0 \colon C^0 \to C^1$ is a matrix over $\mathbb{Q}$ with one column per (tool, field) pair and one row per (edge, dimension) pair. Row r has -1 in the `from_field` column and $+1$ in the `to_field` column, provided the field is in the relevant field set:

- δ^0_{obs}: columns are (v, f) with $f \in O_v$. Entries are nonzero only if the field is observable.
- δ^0_{full}: columns are (v, f) with $f \in I_v$. All dimension-linked fields contribute.

The *coherence fee* is

$$\varphi(G) = \mathrm{rank}(\delta^0_{\mathrm{full}}) - \mathrm{rank}(\delta^0_{\mathrm{obs}}).$$

This counts convention dimensions whose consistency is structurally unverifiable from observable schemas. The fee is non-negative by the inclusion $O_v \subseteq I_v$ [1, 2]. A *blind spot* is an edge-dimension where at least one endpoint field is hidden (in $I_v \setminus O_v$); a *bridge* is a per-blind-spot fix suggesting that the hidden field be added to the tool's observable schema.

3 The Coherence Fee

Example 3.1 (Non-additivity of fee under partition)**.** Consider three tools A, B, C in a chain $A \to B \to C$ sharing dimension `amount_unit`:

Tool	Internal state	Observable schema
A	`amount_unit`, `payload`	`payload`
B	`amount_unit`, `payload`	`amount_unit`, `payload`
C	`amount_unit`, `result`	`result`

Both edges carry dimension `amount` linking `amount_unit` on each side.

Sub-fees. $\{A, B\}$: A hides `amount_unit` but B exposes it. Observable coboundary row: $[0, -1, +1, 0] \neq 0$; same rank as full. $\varphi(\{A, B\}) = 0$.

$\{B, C\}$: B exposes, C hides. Same argument. $\varphi(\{B, C\}) = 0$.

Flat fee. δ^0_{obs} has columns for A.`payload`, B.`amount_unit`, B.`payload`, C.`result` and rows with entries only in B's column for both edges. $\mathrm{rank}_{\mathrm{obs}} = 1$.

δ^0_{full} adds columns A.`amount_unit` and C.`amount_unit`; both rows become independent. $\mathrm{rank}_{\mathrm{full}} = 2$.

$\varphi(\mathrm{flat}) = 2 - 1 = 1 > 0 + 0$. The hidden convention propagates through B unseen by either sub-workflow.

Corollary 3.2 (Boundary vanishing under full disclosure)**.** *If every cross-partition edge dimension has both endpoint fields in their respective tools' observable schemas, then $\beta(\mathcal{P}) = 0$ and the fee is exactly additive: $\varphi(G) = \sum_i \varphi(G_i)$.*

Proof. When both `from_field` and `to_field` are observable for every cross-partition dimension, each cross-row x_{full} and x_{obs} have the same nonzero entries in the same columns. Hence $\rho_{\mathrm{full}} = \rho_{\mathrm{obs}}$ and $\beta = 0$. □

4 Hierarchical Decomposition

Definition 4.1 (Partition)**.** A *partition* of G is a collection $\mathcal{P} = \{G_1, \dots, G_k\}$ of disjoint subsets of V with $\bigcup_i G_i = V$. An edge is *internal* to G_i if both endpoints lie in G_i; otherwise it is *cross-partition*.

The coboundary matrix decomposes into blocks:

$$\delta^0 = \begin{pmatrix} M_1 & & 0 \\ & \ddots & \\ 0 & & M_k \\ \hline X_1 & \cdots & X_k \end{pmatrix}$$

where M_i contains the internal rows for group G_i and the bottom block $[X_1 \mid \cdots \mid X_k]$ contains the cross-partition rows.

Definition 4.2 (ρ)**.** For a choice of field set (obs or full), define $\rho = \operatorname{rank}(\delta^0) - \sum_i \operatorname{rank}(M_i)$, the rank contribution of cross-partition rows above the block-diagonal internal rows.

Theorem 4.3 (Hierarchical Fee Decomposition)**.** *For any partition $\mathcal{P}$ of a composition G,*

$$\varphi(G) = \sum_{i=1}^{k} \varphi(G_i) + \beta(\mathcal{P})$$

where the boundary fee $\beta(\mathcal{P}) = \rho_{\text{full}} - \rho_{\text{obs}} \geq 0$ *is the difference in cross-partition rank contribution between full and observable coboundary matrices.*

Proof. The block-diagonal structure gives $\operatorname{rank}(M_1 \oplus \cdots \oplus M_k) = \sum_i \operatorname{rank}(M_i)$ for both obs and full. Hence

$$\begin{aligned}
\varphi(G) &= \operatorname{rank}(\delta^0_{\text{full}}) - \operatorname{rank}(\delta^0_{\text{obs}}) \\
&= \left[\textstyle\sum_i \operatorname{rank}(M_{i,\text{full}}) + \rho_{\text{full}}\right] - \left[\textstyle\sum_i \operatorname{rank}(M_{i,\text{obs}}) + \rho_{\text{obs}}\right] \\
&= \textstyle\sum_i \varphi(G_i) + \beta(\mathcal{P}).
\end{aligned}$$

Non-negativity. The key observation is that coordinate projection cannot create new linear dependencies:

Lemma 4.4. *Let $P\colon \mathbb{Q}^{C_{\text{full}}} \to \mathbb{Q}^{C_{\text{obs}}}$ be the coordinate projection that drops hidden-field columns. If vectors $v_1, \ldots, v_m$ are linearly independent modulo a subspace W, then $P(v_1), \ldots, P(v_m)$ are linearly independent modulo $P(W)$.*

Proof. If $\sum a_i P(v_i) \in P(W)$ then $P(\sum a_i v_i) \in P(W)$, so $\sum a_i v_i \in W + \ker P$. Now $\ker P$ consists of vectors supported entirely on hidden-field columns (the columns that P projects away). Cross-partition rows, by construction, have nonzero entries only in columns indexed by fields of their two endpoint tools—both observable and hidden fields of those specific tools. Hidden-field columns of one partition group cannot coincide with endpoint columns of a cross-partition row spanning two *distinct* groups, because column indices are (v, f) pairs and cross-rows reference only the endpoints' own fields. Therefore $\ker P$ intersects the span of the cross-rows trivially: any vector in $W + \ker P$ that is also a linear combination of cross-rows must already lie in W. Hence $\sum a_i v_i \in W$, forcing all $a_i = 0$. □

Applying the lemma with $v_i = x_{i,\text{full}}$ (cross-rows) and $W = \operatorname{rowsp}(I_{\text{full}})$ (internal rows): since $P(r_{\text{full}}) = r_{\text{obs}}$ for each row and $P(\operatorname{rowsp}(I_{\text{full}})) \subseteq \operatorname{rowsp}(I_{\text{obs}})$, $\rho_{\text{full}} \geq \rho_{\text{obs}}$. □

Theorem 4.5 (Tower Law for Boundary Fees)**.** *Let $\mathcal{P} = \{G_1, \ldots, G_k\}$ be a partition of G, and let $\mathcal{Q}$ be a refinement of $\mathcal{P}$ obtained by sub-partitioning each group G_i via $\mathcal{P}_i$. Then*

$$\beta(\mathcal{Q}) = \beta(\mathcal{P}) + \sum_{i=1}^{k} \beta(\mathcal{P}_i).$$

Proof. Apply theorem 4.3 to partition $\mathcal{Q}$ of G: $\varphi(G) = \sum_j \varphi(Q_j) + \beta(\mathcal{Q})$. Apply it again to each sub-partition $\mathcal{P}_i$ of G_i: $\varphi(G_i) = \sum_{j \in \mathcal{P}_i} \varphi(Q_j) + \beta(\mathcal{P}_i)$. Substitute into the coarse decomposition $\varphi(G) = \sum_i \varphi(G_i) + \beta(\mathcal{P})$; the local fees $\varphi(Q_j)$ cancel, leaving $\beta(\mathcal{Q}) = \beta(\mathcal{P}) + \sum_i \beta(\mathcal{P}_i)$. □

Corollary 4.6 (Monotonicity under refinement)**.** *If $\mathcal{Q}$ refines $\mathcal{P}$, then $\beta(\mathcal{Q}) \geq \beta(\mathcal{P})$.*

Proof. Each $\beta(\mathcal{P}_i) \geq 0$ by theorem 4.3. □

Remark 4.7 (Lattice interpretation)*.* The boundary fee defines a monotone function on the refinement lattice of partitions of V: the trivial partition $\{V\}$ has $\beta = 0$; the discrete partition into singletons has $\beta = \varphi(G)$ (the total fee); every refinement moves monotonically upward. Operationally, each level of delegation in a hierarchical tool composition adds non-negative hidden cost.

Extremal cases. The vanishing corollary (corollary 3.2) gives the lower extreme: $\beta = 0$ when all boundary fields are observable.

Theorem 4.8 (Extremal boundary fee)**.** *For a star graph with hub H and spokes $S_1, \ldots, S_n$, where all conventions are hidden at every tool, the partition $\mathcal{P} = \{H\} \cup \{S_1, \ldots, S_n\}$ achieves $\beta(\mathcal{P}) = \varphi(G) = n$.*

Proof. Every edge $H \to S_i$ is cross-partition. Both groups are internally edge-free, so $\sum_i \varphi(G_i) = 0$ and $\beta = \varphi(G)$. Concretely, $\rho_{\text{obs}} = 0$ (empty observable schemas) and $\rho_{\text{full}} = n$ (each edge contributes an independent full-coboundary row via its unique spoke column). □

5 Lattice Properties

Remark 5.1 (Non-valuation)*.* The boundary fee is *not* a valuation on the partition lattice. For the chain $A \to B \to C$ of example 3.1, take $\mathcal{P} = \{AB, C\}$ and $\mathcal{Q} = \{A, BC\}$. Then $\mathcal{P} \wedge \mathcal{Q} = \{A, B, C\}$ (the discrete partition) and $\mathcal{P} \vee \mathcal{Q} = \{ABC\}$ (the trivial partition), giving $\beta(\mathcal{P}) + \beta(\mathcal{Q}) = 2$ but $\beta(\mathcal{P} \wedge \mathcal{Q}) + \beta(\mathcal{P} \vee \mathcal{Q}) = 1 + 0 = 1$. The same hidden convention at B causes boundary fee in both $\mathcal{P}$ and $\mathcal{Q}$, but resolving it once (in the discrete partition) suffices.

Remark 5.2 (Non-submodularity)*.* The boundary fee is *not* submodular on the partition lattice. An adversarial survey of 10,000 random compositions (635,095 partition pairs) found 4,061 violations of $\beta(\mathcal{P} \wedge \mathcal{Q}) + \beta(\mathcal{P} \vee \mathcal{Q}) \leq \beta(\mathcal{P}) + \beta(\mathcal{Q})$, with maximum violation magnitude 3.

Minimal counterexample: four tools T_0, T_1, T_2, T_3 with edges forming a cycle plus a shortcut. T_0 and T_1 have all fields hidden; T_2 and T_3 each expose one field. The partitions $\mathcal{P} = \{T_2\} \mid \{T_0, T_1, T_3\}$ and $\mathcal{Q} = \{T_3\} \mid \{T_0, T_1, T_2\}$ both have $\beta = 0$ (the isolated tool's single cross-edge has its endpoint visible). Their meet (singletons) has $\beta = 1$: refining into singletons exposes a hidden convention path $T_1 \to T_2 \to T_0$ that is internal in both $\mathcal{P}$ and $\mathcal{Q}$ but cross-partition in the meet.

This is consistent with the algebraic structure: while ρ_{full} and ρ_{obs} are individually submodular (as matroid rank restricted to row sets), their difference $\beta = \rho_{\text{full}} - \rho_{\text{obs}}$ need not be. The 10 bundled compositions happen to satisfy submodularity, but this is an empirical accident of their low-density, pipeline-like topology.

6 Prescriptive Diagnostics

6.1 Minimum Disclosure Set

The coherence fee counts hidden convention dimensions, but does not say *which* fields to disclose. The bridges mechanism of [1] suggests one disclosure per blind spot, but blind spots are per-edge while the fee lives in a vector space: some disclosures are redundant.

Definition 6.1 (Minimum disclosure set)**.** A *minimum disclosure set* for a composition G is a smallest set $S \subseteq \{(v, f) : v \in V,\ f \in I_v \setminus O_v\}$ such that adding every $(v, f) \in S$ to the observable schema of tool v reduces $\varphi(G)$ to zero.

Theorem 6.2 (Disclosure cardinality equals fee)**.** *$|S| = \varphi(G)$ for any minimum disclosure set S.*

Proof. Each disclosure adds at most one column to δ^0_{obs} outside $\text{col}(\delta^0_{\text{obs}})$. Reducing the fee by k requires k independent new columns, hence $|S| \geq \varphi(G)$. For the converse, let C_{hid} denote the columns of δ^0_{full} not present in δ^0_{obs} (the hidden-field columns). Since δ^0_{obs} is the submatrix of δ^0_{full} obtained by deleting exactly C_{hid}, the rank contributed by C_{hid} modulo the column space of δ^0_{obs} is $\text{rank}(\delta^0_{\text{full}}) - \text{rank}(\delta^0_{\text{obs}}) = \varphi(G)$. Therefore $\varphi(G)$ independent columns from C_{hid} span this quotient, and any basis of the quotient selects exactly $\varphi(G)$ disclosures. □

Remark 6.3 (Bridges vs. disclosures). The per-blind-spot bridges produce $|\text{bridges}| \geq 2\,\varphi(G)$ suggestions (one per hidden endpoint per edge). The minimum disclosure set collapses this: when tool A hides field f across multiple edges, a single disclosure of (A, f) resolves all of them. Empirically, $|\text{bridges}| \geq 2 \cdot |S|$ across all 10 bundled compositions.

6.2 Online Resolution

When a partial composition includes placeholder tools for as-yet-unspecified subsystems, `conditional_diagnos` computes a worst-case fee and exports boundary obligations. As real tools arrive, the composition can be updated incrementally.

Corollary 6.4 (Resolution monotonicity). *Let $(G, \partial G)$ be a partial composition with placeholder tools having empty observable schemas. Replacing any placeholder with a real tool t (same edges, $O_t \supseteq \emptyset$) can only decrease or maintain the coherence fee: the placeholder is the worst case.*

Proof. The placeholder has $O_t = \emptyset$ and $I_t = \{f \mid f = d.to\_field$ for some edge dimension d incident to the placeholder$\}$. The real tool's internal state must contain every field referenced by its edge dimensions (otherwise the composition is invalid); since the placeholder's internal state is exactly those fields, $I_{\text{real}} \supseteq I_{\text{placeholder}}$ is a consequence of composition validity, not an assumption. Adding columns to δ^0_{full} can only weakly increase its rank. Meanwhile the real tool has $O_{\text{real}} \supseteq \emptyset$, so δ^0_{obs} gains columns that were already present in δ^0_{full}, weakly increasing $\text{rank}(\delta^0_{\text{obs}})$ by at least as much. Hence φ can only decrease. □

7 Case Study: Financial Settlement Pipeline

We construct an 8-tool composition modeling a hierarchical financial settlement workflow. The pipeline consists of: `price_fetch` → `currency_convert` → `compliance_check` → `risk_assess` → `ledger_write` → `audit_log` → `notification` → `archive`, with a feedback edge `audit_log` → `compliance_check` creating a cycle ($\beta_1 = 1$).

What could go wrong. Without coherence diagnostics, this pipeline silently propagates convention mismatches through the chain. The currency converter's internal `rounding_mode` determines how fractional amounts are truncated; the compliance checker's `threshold_currency` assumes a specific rounding convention for regulatory thresholds. Neither tool exposes these fields. Similarly, `compliance_check` hides its `jurisdiction`, so `risk_assess` may apply a risk model calibrated to a different regulatory framework. The `confidence_interval` from risk assessment silently determines which `accounting_standard` the ledger uses. None of these are type errors—every schema validates. The coherence fee of 7 says: there are 7 independent convention dimensions that no bilateral check can verify.

Fee and disclosure. The coherence fee is $\varphi = 7$ with 8 blind spots and 15 bridges. The minimum disclosure set has $|S| = 7 = \varphi$:

Tool	Field
`currency_convert`	`rounding_mode`
`compliance_check`	`jurisdiction`
`risk_assess`	`risk_model_version`
`risk_assess`	`confidence_interval`
`ledger_write`	`decimal_precision`
`audit_log`	`timezone`
`notification`	`locale`

Bridges suggest 15 fixes; the disclosure set achieves the same fee reduction with 7—a factor of 2.1× savings. The matroid structure explains why: when `timezone` on `audit_log` is hidden across multiple edges, a single disclosure resolves all of them.

Boundary fee under partition. Splitting into front-office {`price_fetch`, `currency_convert`, `compliance_` and back-office {`ledger_write`, `audit_log`, `notification`, `archive`} gives local fees $(2, 3)$ and boundary fee 2. The boundary fee accounts for two hidden conventions (`confidence_interval` and `decimal_precision`) that cross the front/back-office boundary: internal to each sub-workflow when viewed locally, but creating obstruction in the flat expansion.

Conditional resolution. Removing `archive` and creating a placeholder with two open-port dimensions gives three fee values:

- **Baseline fee** (known sub-composition only): $\varphi = 6$.
- **Worst-case fee** (placeholder with $O = \emptyset$): $\varphi = 7$, with two boundary obligations (`archive_id`, `retention_policy`).
- **Resolved fee** (real `archive` tool substituted): $\varphi = 7$. The real tool exposes `archive_id` but hides `retention_policy`, meeting one obligation and leaving one remaining. Fee delta: $7 - 7 = 0$ (the real tool is as opaque as the placeholder for `retention_policy`).

This demonstrates resolution monotonicity (corollary 6.4): the resolved fee equals the worst-case fee here because `archive` happens to hide the same field as the placeholder. A more transparent replacement would yield $\varphi < 7$.

8 Empirical Results

All theorems are computationally verified against 10 bundled compositions (including the 8-tool financial settlement pipeline) in Bulla v0.34, with sampled binary partitions and tower law pairs. Separately, an adversarial survey of 10,000 random compositions checked 635,095 partition pairs for submodularity.

Metric	Value
Block rank formula + non-negativity verified	120/120
Tower law pairs verified (sampled)	2,778/2,778
$\lvert S\rvert = \varphi$ verified	10/10
Submodularity (adversarial)	disproved (4,061/635,095)
Resolution monotonicity verified	8/8
8-tool financial settlement pipeline	
φ (fee)	7
$\lvert S\rvert$ (disclosure set)	7
|bridges|	15
β (front/back partition)	2
Conditional: baseline → worst-case → resolved	$6 \to 7 \to 7$

Every composition with nonzero fee produces nonzero boundary fee for most partitions: the hierarchical blind spot is the dominant regime, not a corner case. Boundary fee is nonzero in over 91% of sampled binary partitions across all compositions.

Remark 8.1 (Trace gap). The Frobenius trace gap $\|\delta_{\text{full}}\|_F^2 - \|\delta_{\text{obs}}\|_F^2$ has been considered as a continuous spectral refinement of the integer fee. Since all coboundary entries are 0 or ± 1, this reduces to the count of nonzero entries in hidden columns:

$$\text{trace gap} = \sum_{b \in \text{blind spots}} \big[\mathbf{1}[b.\text{from\_hidden}] + \mathbf{1}[b.\text{to\_hidden}]\big],$$

a weighted blind-spot count derivable from the existing diagnostic. It can be positive when $\varphi = 0$ (hidden columns in the span of observable columns), and adds no information beyond the blind-spot structure. A genuine spectral refinement requires the eigenvalue spectrum of the sheaf Laplacian, which we defer to future work.

9 Related Work

Contract-based design and assume-guarantee reasoning. Interface theories [8] define compatibility conditions between open components; Benveniste et al. [9] systematize the algebra of contracts (assumptions on the environment, guarantees on the component) for system design. Pasareanu et al. [10] show how environment assumptions can be learned incrementally. The coherence fee's boundary obligations are the analog for tool compositions: computed from observable schema structure rather than specified manually. The fee quantifies the cost of *not having* a contract—the number of convention dimensions a contract would need to cover.

Sheaf cohomology for data integration. Curry [4] develops the cellular sheaf framework with computable cohomology, including Mayer–Vietoris sequences for decomposition. Robinson [5] applies sheaf theory to signal processing and sensor fusion, using the sheaf Laplacian to detect inconsistency. Ghrist [6] provides accessible background on applied algebraic topology. Hansen and Ghrist [7] study opinion dynamics via sheaf Laplacians. The coherence fee instantiates the same mathematical framework for tool compositions: the coboundary operator δ^0 is the signed incidence map of a cellular sheaf, and the fee is a computable rank obstruction [2]. The boundary fee is the partition-relative analog of sheaf-theoretic gluing constraints, and the vanishing corollary (corollary 3.2) is the linearized version of the SCPI contractible-nerve vanishing theorem [3].

Tool-composition ecosystems. The Model Context Protocol [11] standardizes how processes expose tool interfaces; orchestration frameworks such as AutoGen [12] and LangGraph [13] define how subsystems communicate and delegate. Tool-using language models [14, 15] compose tools dynamically at inference time. These systems define the *transport layer*; the coherence fee operates above it. Given a composition graph (which any framework can export), Bulla diagnoses what the framework cannot check: convention consistency across tool boundaries. A reference LangGraph integration demo (bundled with the implementation) constructs a 4-tool trade pipeline, runs it through LangGraph with all schemas validating, then diagnoses a coherence fee of 3 from hidden conventions invisible to the orchestrator. This is orthogonal to and composable with existing orchestration.

10 Conclusion

We have introduced the coherence fee as a computable diagnostic for hidden convention risk in hierarchical tool compositions, proved that hierarchical decomposition adds non-negative cost (with the tower law quantifying the cost per level), and shown that a minimum disclosure set of exactly φ fields suffices to eliminate all blind spots. The boundary fee is monotone on the partition lattice but neither a valuation nor submodular, as established by adversarial computational survey.

The coherence fee measures *structural verifiability*, not semantic correctness. A fee of zero means every convention dimension *can* be checked from observable schemas—not that every convention *is* correct. The gap between verifiability and correctness is where domain-specific validation begins; the fee tells you where to look.

Limitations. The model assumes stable tool signatures for the duration of analysis; runtime schema drift is not handled. Obligation satisfaction is verified structurally (field presence) rather than semantically (value correctness). The current implementation treats all conventions as equally weighted.

Future work. A spectral refinement of the fee via the eigenvalue spectrum of the sheaf Laplacian would provide a continuous measure of convention risk, weighting each hidden dimension by its connectivity. The witness capsule protocol would formalize assume-guarantee contracts for hierarchical tool compositions, enabling plan-time verification across delegation boundaries. Integration with tool-orchestration frameworks (LangGraph, AutoGen) would validate the theory against real-world composition patterns and surface the fee in existing developer workflows. Compositional verification techniques and runtime monitoring [10] could extend the static analysis to dynamic composition graphs.

References

[1] J. Komkov, "The coherence fee: Edge-local blindness at the string-table seam and the topological price of cross-system composition," Technical Report, Res Agentica, February 2026. `https://github.com/jkomkov/bulla`

[2] J. Komkov, "Sheaf-theoretic diagnostics for tool compositions," Technical Report, Res Agentica, 2026. `https://github.com/jkomkov/bulla`

[3] J. Komkov, "Predicate invention under sheaf constraints," Technical Report, Res Agentica, 2026. `https://github.com/jkomkov/bulla`

[4] J. Curry, *Sheaves, cosheaves and applications*, PhD thesis, University of Pennsylvania, 2014.

[5] M. Robinson, *Topological Signal Processing*, Springer, 2014.

[6] R. Ghrist, *Elementary Applied Topology*, Createspace, 2014.

[7] J. Hansen and R. Ghrist, "Toward a spectral theory of cellular sheaves," *Journal of Applied and Computational Topology*, vol. 3, pp. 315–358, 2019.

[8] L. de Alfaro and T. A. Henzinger, "Interface automata," in *Proc. 9th ESEC/FSE*, ACM, 2001, pp. 109–120.

[9] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K. Larsen, "Contracts for system design," *Foundations and Trends in Electronic Design Automation*, vol. 12, no. 2–3, pp. 124–400, 2018.

[10] C. S. Păsăreanu, D. Giannakopoulou, M. G. Bobaru, J. M. Cobleigh, and H. Barringer, "Learning to divide and conquer: Applying the L* algorithm to automate assume-guarantee reasoning," *Formal Methods in System Design*, vol. 32, no. 3, pp. 175–205, 2008.

[11] Anthropic, "Model context protocol specification," 2024. `https://modelcontextprotocol.io/specification`

[12] Q. Wu, G. Banberanber, Y. Hou, H. Ye, L. Zheng, A. Kazemitabaar, N. Cai, D. Jiang, Z. Li, A. H. Awadallah, R. Ronen, and C. Wang, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," *arXiv:2308.08155*, 2023.

[13] LangChain, "LangGraph: Build stateful, multi-actor applications with LLMs," 2024. `https://github.com/langchain-ai/langgraph`

[14] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," in *Proc. NeurIPS*, 2023.

[15] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, and M. Sun, "ToolLLM: Facilitating large language models to master 16000+ real-world APIs," in *Proc. ICLR*, 2024.

[16] J. Komkov, "Signed-Incidence Structure in Compositional Verification: A Structural Note," `papers/signed-incidence/paper/signed-incidence.pdf`, 2026.