Institutionalized disagreement

Kenneth Pernyér, Reflective Labs
Stockholm, Sweden
kenneth.pernyer@reflective.se · reflective.se · ORCID 0009-0006-4543-1156

A language model can produce a plan. Nothing about producing a plan establishes that it is sound, and nothing about it being sound establishes the right to execute it. This paper describes Formation, the reasoning layer that sits between human intent and the governed engine described in my previous report, and the six mandatory stages an intent passes through before any of it reaches the commit boundary: admission across four dimensions, decomposition into an intent tree, a huddle in which several reasoners plan in parallel and failures are dropped rather than hidden, adversarial review by five kinds of skeptic, simulation across five dimensions, and commit. Two invariants carry the design. Authority may only *narrow* down the intent tree, so no subtask can outrank its parent; and learning flows backward into planning priors while authority flows forward, so no learned quantity can ever appear in an authority computation. I show why the second isn't a stylistic preference but a precondition for the determinism proved in the substrate: an engine whose authority depends on learned state has a fixed point that moves. I also take the adversarial stage further than my implementation goes, and argue that aggregating skeptic verdicts is a judgment aggregation problem in the sense of List and Pettit — with a discursive dilemma that a majority vote can't escape and a veto can. Whether the review loop terminates without a budget is stated as an open problem, not a result.

1. Introduction

The interesting question in organizational AI isn't whether a model can produce a plan anymore. It can. The question is what has to happen between a plan existing and an organization acting on it, and the honest answer is: rather a lot, most of it social rather than computational.

Organizations don't accept plans because the plan is good. They accept plans that have survived challenge by people whose job it was to object — the finance partner who asks what it really costs, the operator who asks who's supposed to do this on a Tuesday in November, the lawyer who asks what happens if the counterparty reads clause four the other way. The plan that comes out the other side is usually worse on paper and far better in practice, because the challenges removed the parts that were only ever true in the plan.

That process is slow, expensive, unevenly applied, and the first thing an organization drops under time pressure. It's also the mechanism that makes organizational decisions trustworthy. Formation is my attempt to make it mandatory, cheap and recorded — to **institutionalize disagreement** instead of hoping for it.

1.1. Position

Formation is a pure client of the convergence engine described in my previous report (Pernyér, 2025), and the engine doesn't know it exists. That report establishes the substrate this one assumes: an append-only context lattice, Suggestors that propose rather than write, a promotion gate that decides, nine invariants, and a Formation that runs to a fixed point. I don't restate it here.

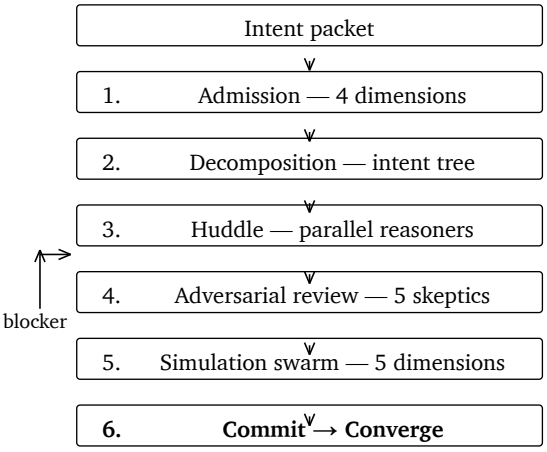

Figure 1 | The intent pipeline. Six stages, all mandatory; there is no trusted-plan exception. A blocker finding in stage 4 returns the plan to the huddle rather than downgrading itself to a warning. Only stage 6 crosses into the engine, where authority is recomputed from scratch.

The division is one sentence. The engine answers *what is allowed to become a fact*. Formation answers *which team should work on this, and how should it think*. Producing a plan never grants the right to execute it: Formation holds zero authority, and every

© 2025 Reflective Labs™. All rights reserved

authority question gets recomputed at the commit boundary by a layer that never saw the reasoning.

2. Admission and decomposition

2.1. Admission is a meet, not a score

An intent arrives as a packet: an outcome, a context, constraints, an authority, what's forbidden, a reversibility class and an expiry. Before any resources go into planning it, the system assesses feasibility on four dimensions — *capability* (can we do this?), *context* (do we know enough?), *resources* (is there budget?) and *authority* (is it permitted?).

Each dimension returns a verdict from the chain

$$\text{Infeasible} < \text{Uncertain} < \text{Constrained} < \text{Feasible},$$

and the admission verdict is the **meet**, not the average:

$$v(\text{intent}) = \min_{i \in \{c,x,r,a\}} v_i. \tag{1}$$

Eq. 1 is the whole design decision. A plan that's brilliantly capable, richly contextualized, generously funded and **not permitted** is infeasible, full stop, and no weighting of the other three can rescue it. Scoring schemes that average across dimensions are how systems talk themselves into acting without authority. A meet can't be talked into anything.

2.2. Authority may only narrow

Decomposition turns an admitted intent into a tree of sub-intents. The invariant that keeps the tree safe is antitonicity of authority: for every edge from parent p to child c in the intent tree T,

$$\mathcal{A}(c) \subseteq \mathcal{A}(p). \tag{2}$$

A subtask never has more authority than the task it came from. Nothing gets acquired by being decomposed.

> **Proposition (Authority bound).** Under Eq. 2, for every node v of an intent tree with root r, $\mathcal{A}(v) \subseteq \mathcal{A}(r)$.

The proof is immediate by induction on depth — Appendix B — and its triviality is the point. This isn't a deep property. It's just *easy to lose* if authority gets recomputed per node from a policy engine at plan time, because a differently-phrased subtask can match a differently-scoped rule. Formation never derives a child's authority. It intersects.

3. The huddle

Stage three runs several reasoners against the same admitted intent in parallel: a language model, a constraint solver, an ML predictor, a causal analysis, a cost estimator, a domain model. Each produces a candidate plan. Candidates that fail get dropped — not repaired, not hidden — and survivors proceed together.

Skepticism	The question it is required to ask
Assumption	What is being assumed without being stated?
Constraint	Do the declared constraints actually hold?
Causal	What are the second-order effects?
Economic	What does this really cost?
Operational	Can this organization actually execute it?

Table 1 | The five skepticisms. They're roles, not models: the same underlying reasoner can play several, and the point of naming them is that each question gets asked whether or not anyone feels like asking it.

Two things about that arrangement matter, and neither is about model quality. First, the reasoners are **heterogeneous by construction**, in the sense I argued in the substrate paper: a solver and a language model fail in uncorrelated ways, so the survivors of a huddle aren't a consensus of one kind of mistake. Second, dropping is honest. A pipeline that silently repairs a failed candidate produces a plan whose provenance is a repair; a huddle that drops it produces fewer plans and an accurate record of how many approaches didn't survive.

This is the exploration side of the classical trade-off (March, 1991). A huddle is deliberately wasteful — most candidates die — and the waste buys a search that doesn't collapse onto the first plausible plan. The exploitation side shows up later, in the priors of Section 6.

4. Adversarial review

Stage four is the one this paper is named after. Every plan gets challenged by five kinds of skeptic before it may proceed.

Findings carry one of three severities. *Info* gets recorded. *Warning* gets recorded and the plan proceeds with it attached. *Blocker* stops the plan, which goes back to the huddle for revision and comes back to be challenged again. There's no threshold at which a blocker becomes a warning, and no quorum that outvotes one.

4.1. Why a veto and not a vote

That last sentence is a design choice with a formal justification, and it's the part of this paper that goes further than my implementation strictly needed.

Aggregating the verdicts of five skeptics is a judgment aggregation problem, and judgment aggregation has an impossibility result (List and Pettit, 2002). Consider three skeptics judging a plan on two premises — P: the demand estimate holds; Q: the

© 2025 Reflective Labs™. All rights reserved

supply contract holds — with the conclusion $C = P \wedge Q$: the plan is sound. Skeptic one accepts both. Skeptic two rejects P and accepts Q. Skeptic three accepts P and rejects Q. Majorities accept P (two of three) and accept Q (two of three), so proposition-wise majority voting concludes C. But every individual skeptic rejects C.

> **Definition (Discursive dilemma).** A profile of individually consistent judgments whose proposition-wise majority aggregate is inconsistent. It isn't an artifact of the example: no aggregation rule satisfying universal domain, anonymity and systematicity avoids it (List and Pettit, 2002).

Every skeptic believes the plan is unsound, and the vote passes it anyway. A system that aggregates adversarial findings by majority, weighted score or confidence average is exposed to exactly this. A veto isn't, because it's not an aggregation rule at all — it's the conclusion-driven alternative, where one attested inconsistency about the whole is decisive no matter how the premises poll. The cost of a veto is that a single wrong blocker can stall a good plan. I'd rather pay that cost, and the record makes it visible: the learning signals of Section 6 include whether a blocker turned out to have been right.

4.2. Does the review loop terminate?

Plan and review alternate: the huddle produces π_0, review yields findings $F(\pi_0)$, a blocker sends the plan back, and the huddle produces π_1. Write $b(\pi)$ for the number of blocking findings against π. The loop halts when $b(\pi_n) = 0$.

I don't have a proof that it does. Termination would follow from a well-founded measure — if each revision strictly decreased b, or strictly decreased some potential the skeptics agree on — but revision is done by reasoners that can introduce new weaknesses while removing old ones, so b isn't monotone, and no measure I've proposed is well-founded in the general case.

> **Open problem (Review convergence).** Characterize the conditions on the revision map ρ and the finding map F under which the alternating sequence $\pi_{n+1} = \rho(\pi_n, F(\pi_n))$ reaches $b = 0$ in finitely many rounds.

In the implementation this gets answered operationally rather than mathematically: the loop runs under a budget, and exhausting it is one of the substrate's honest exits (Pernyér, 2025) rather than a silent degradation to *proceed*. That's an adequate engineering answer and an unsatisfying scientific one, and I'm stating it that way on purpose.

5. Simulation and commit

Stage five simulates the surviving plan across five dimensions — outcome, cost, policy, causal and operational — and returns *Proceed*, *ProceedWithCaution* or *DoNotProceed*. Like admission, the aggregate is a meet: caution anywhere is caution overall, and a single *DoNotProceed* is decisive.

Stage six is the only stage that leaves Formation. The surviving plan gets submitted as a proposal, and the engine recomputes every authority question from scratch at the commit boundary, using policy and the current context and nothing that Formation concluded. Formation's verdicts are evidence, not permissions.

Worth stating sharply, because it's counter-intuitive: the layer that did all the reasoning isn't trusted with the result of its own reasoning. Six stages of admission, decomposition, debate, challenge and simulation confer exactly zero authority. They make the plan *better* and the record *complete*. They don't make it *allowed*.

6. Learning that cannot reach authority

Outcomes flow back. When an execution matches, beats or misses the plan's expectation, and when an adversarial blocker or warning turns out to have been correct, those signals update planning priors: which reasoners to convene for this kind of intent, how much budget to allocate, which skeptics have been worth listening to in this domain.

The invariant is a separation of channels. Writing θ for the learned priors, Π for the policy and I for the intent, Formation requires

$$\mathcal{A} = \mathcal{A}(\Pi, I), \quad \text{never} \quad \mathcal{A}(\Pi, I, \theta). \tag{3}$$

Learning flows backward into planning. Authority flows forward from policy. They never touch.

Eq. 3 looks like governance hygiene and is actually a load-bearing mathematical requirement. The substrate paper proves a Formation reaches a unique fixed point given a pinned fleet and a pinned policy. If authority were a function of learned state, the pinned policy would no longer pin the gate: the same context, the same Suggestors and the same policy could promote different facts on Tuesday than on Monday, because a prior had shifted in between. Determinism would become conditional on training history, replay would stop meaning anything, and the audit answer to *why was this permitted?* would become *because of everything that has happened to us*. Keeping θ out of $\mathcal{A}$ is what keeps the fixed point still.

The flywheel that remains is real, though: more intents produce more episodes, which produce better priors, which resolve intents faster, which allows more intents. It's organizational learning in the double-loop sense (Argyris and Schön, 1978) — the system revises not just its plans but its theory of which

© 2025 Reflective Labs™. All rights reserved

challenges are worth making — and it's confined, by construction, to the loop that doesn't decide anything.

7. Domain packs

The pipeline is populated by thirteen organizational packs, each a bundle of Suggestors, invariants and fact prefixes for one domain: knowledge, customers, people, legal, performance, autonomous organization, growth marketing, product engineering, operations support, procurement, partnerships, virtual teams and research. Eight blueprints compose packs into end-to-end processes — lead-to-cash, hire-to-retire, procure-to-pay, issue-to-resolution, idea-to-launch, campaign-to-revenue, partner-to-value, patent research.

I'm recording the inventory here mainly to be precise about the unit of reuse. The pack is the unit, not the agent and not the workflow: it carries its own invariants, so composing two packs into a blueprint composes their invariants rather than merging their code paths.

8. Limitations

The five skepticisms are a taxonomy I chose, not one I derived. They cover the failures I've seen. I have no argument that they're exhaustive, and a sixth kind I haven't thought of would be invisible to the review by construction. Measuring the diversity of a skeptic panel — as opposed to asserting it — is unsolved here.

Simulation quality bounds everything downstream of stage five. A simulation that models the organization badly returns *Proceed* with the same confidence it would return anything else, and the pipeline has no way to detect that its model of the world is wrong rather than its plan.

The learning signals are sparse, and the flywheel needs volume before priors carry information. In a low-volume domain the priors are mostly the defaults they were seeded with, and I don't currently distinguish *learned to be true* from *never updated* in the record.

Finally, the collective intelligence literature suggests group performance depends on properties of the group — turn-taking, social sensitivity — more than on the ability of its members (Woolley et al., 2010). Whether any analogue of those properties exists for a panel of reasoners, and whether Formation's structure supplies or destroys them, I don't know.

9. Conclusion

Formation institutionalizes the part of organizational decision-making that usually depends on someone being brave enough to object. Six mandatory stages, five named skepticisms, a blocker no quorum can overrule, and a plan that has to survive all of it before it's allowed to become a proposal — and even then, a layer that recomputes authority as if the reasoning never happened.

Two invariants do most of the work. Authority narrows down the intent tree and never widens, so decomposition can't manufacture permission. Learning flows backward into priors and never into authority, so the fixed point the substrate proves stays where the proof put it. Everything else in the pipeline is a matter of taste and evidence, and I expect to revise it.

The part I'm least sure of is the part I'm most interested in: whether adversarial review converges on its own, or whether a budget is the only honest way to end an argument.

> **Code and correspondence.** The work described here is implemented, not sketched. The source behind every claim in this paper — the engine, the extension, the fixtures and the tests that hold the invariants — can be shared on request, including the parts that did not work. Write to kenneth.pernyer@reflective.se. We are equally interested in being told where the argument is wrong.

References

Argyris, C., Schön, D.A., 1978. Organizational Learning: A Theory of Action Perspective. Addison-Wesley.

List, C., Pettit, P., 2002. Aggregating sets of judgments: An impossibility result. Economics and Philosophy 18, 89–110.

March, J.G., 1991. Exploration and exploitation in organizational learning. Organization Science 2(1):71–87.

Pernyér, K., 2025. Proposal, gate, commitment: deterministic convergence as a substrate for organizational decisions. Reflective Labs technical report.

Woolley, A.W., Chabris, C.F., Pentland, A., Hashmi, N., Malone, T.W., 2010. Evidence for a collective intelligence factor in the performance of human groups. Science 330, 686–688.

© 2025 Reflective Labs™. All rights reserved

A. The six stages, operationally

S1 Admission Evaluate capability, context, resources and authority; take the meet per Eq. 1. *Infeasible* rejects the intent before any planning budget is spent.

S2 Decomposition Build the intent tree. A child's authority is the intersection of the parent's authority with the child's declared scope; it is never derived independently.

S3 Huddle Convene the reasoners selected by the priors for this intent class. Run in parallel, drop failures, record how many candidates died and why.

S4 Adversarial review Run the five skepticisms of Table 1. Any blocker returns the plan to S3. Info and warning findings attach to the plan and travel with it.

S5 Simulation Simulate on outcome, cost, policy, causal and operational dimensions; take the meet of the three-valued verdicts.

S6 Commit Submit as a proposal. Authority is recomputed at the gate from policy and context alone. Formation's verdicts are attached as evidence and carry no permission.

B. Two short proofs

B.1. Authority bound

By induction on the depth of v in T. At depth zero, $v = r$ and $\mathcal{A}(r) \subseteq \mathcal{A}(r)$. Suppose the claim holds at depth n, and let v be at depth $n+1$ with parent p at depth n. By Eq. 2, $\mathcal{A}(v) \subseteq \mathcal{A}(p)$, and by hypothesis $\mathcal{A}(p) \subseteq \mathcal{A}(r)$; transitivity of $\subseteq$ gives the claim.

The content of the proposition is entirely in enforcing Eq. 2 at tree construction, which the implementation does by making intersection the only constructor available for a child's authority.

B.2. The separation invariant

Let K be a context, S a pinned Suggestor fleet and Π a pinned policy, and suppose the gate admits a proposal p if and only if $\mathcal{A}(\Pi, I) \vdash p$. The substrate result gives a unique normal form K^* for (K, S, Π) (Pernyér, 2025).

Now suppose instead $\mathcal{A} = \mathcal{A}(\Pi, I, \theta)$. Take two runs from the same K, S and Π separated in time by an outcome that updated θ to θ', with some p admitted under θ and refused under θ'. Both runs terminate, both are locally confluent, and their normal forms differ by p. Uniqueness fails — not because the substrate proof is wrong, but because its hypothesis has been withdrawn: the gate was assumed to be a function of policy and context, and it's now a function of history.

© 2025 Reflective Labs™. All rights reserved