The Coordination Layer: Why Multi-Agent AI Needs Protocols, Not Just Better Models

From model-centric benchmarks to interaction-condition evaluation

Nicholas Zinner Beacon Bot

Future Shock (future-shock.ai)

May 2026

Abstract

Most AI evaluation still asks a model-centric question: which model scores higher on a fixed task? That framing is too small for multi-agent systems. Once a deployment includes roles, tools, memory, shared context, approval rules, handoff schemas, event logs, and other agents, the behavior we observe belongs to the whole setup, not to the base model alone. This paper names that setup the **coordination layer**: the rules and records that determine how agents exchange state, divide labor, make decisions, and recover from error. We formalize multi-agent behavior as a function of model identity, assigned role, history, protocol, social context, timing, incentives, peer composition, and environment state. We then use three internal Future Shock scenario probes—Ark Protocol, Chaos Lab, and Startup Build—as motivating evidence. In these runs, changing the room around the model changed the outcome: ballot design altered collective survival negotiation; schema scaffolding changed apparent crisis-governance performance; context exposure and final-vote semantics changed whether five-agent founder teams shipped a software artifact. The cleanest slice comes from Startup Build, where a matched three-seed replication produced 0/15 ship votes under a strict-QA ballot frame and 15/15 under a deadline-pressure frame. The paper's claim is not that these small internal runs establish model rankings. They do not. The claim is narrower and more useful: multi-agent evaluation should publish and vary the interaction condition, because protocol, context, scoring, and receipts are part of the system under test.

Keywords: multi-agent systems, coordination layer, interaction condition, agent protocols, scenario-based evaluation, handoffs, audit receipts, structured multi-agent environments

1 Introduction: Model Scores Are Not Enough

A multi-agent system is not a smarter chatbot with extra name tags. It is a room: roles, rules, memory, shared state, tools, permissions, deadlines, votes, logs, and people or agents watching from different angles. Change the room and the same model can behave differently.

That is a problem for the way AI systems are usually evaluated. A benchmark fixes a task, swaps models, and reports a score. That can be useful for single-turn or tightly scoped tasks. It is much less useful when five agents are planning a launch, voting on a proposal, or trying to stabilize a simulated station under crisis. In those settings, the observed behavior is not just "what the

model can do." It is what the whole setup makes easy, hard, visible, punishable, recoverable, or invisible.

This paper makes one claim:

> **The interaction condition—not the model—is the right unit of analysis for multi-agent AI systems.**

We define the *coordination layer* as the rules, records, interfaces, and approval gates that determine how agents in a structured multi-agent environment exchange state, divide labor, make decisions, and recover from error. The paper contributes four things:

1. a definition of the coordination layer as an object of system design;
2. a formal interaction-condition vocabulary for multi-agent evaluation;
3. three internal scenario probes showing where model-centric evaluation can mislead;
4. a disclosure agenda for future coordination benchmarks: manifests, handoff receipts, scorer definitions, and protocol sensitivity tests.

The caveat comes early because it matters. Ark Protocol, Chaos Lab, and Startup Build are scenario probes, not public benchmarks. The datasets are small, the metrics are exploratory, and the harness is part of the system under test. The point is not to crown a winning model. The point is to show that any evaluation trying to explain deployed multi-agent behavior has to study the room around the model.

2 Background: From Scaffolding to Coordination

2.1 Scaffold-Dependent Capability

Agent engineers already know that scaffolding changes capability. The same base model with different tools, retrieval pipelines, and memory layers produces measurably different task performance on coding, browsing, and reasoning workloads. Our prior internal work on agent memory architectures [5] argued for a similar inversion at the cognitive layer: retrieval is not memory, and the boundary between programmatic recall and language-model reasoning is where production agent systems are actually designed.

2.2 Multi-Agent Adds New Variables

Multi-agent systems extend this argument. They introduce variables that have no single-agent analog:

- **Peer composition.** Which other models are present, and which roles do they play.
- **Role assignment.** What objective, authority, and persona an agent is given.
- **Shared context format.** What each agent sees of what others have done.
- **Decision procedure.** How proposals become commitments: votes, ratification, approval gates.
- **Handoff mechanics.** How state, ownership, and unfinished work move between agents.

- **Audit and receipt structure.** What evidence remains after a decision is taken.

These are not minor ergonomic decisions. Each of them changes the action space, the observability, and the incentive structure of the agents inside the system. They are part of the machinery of a multi-agent deployment, and they are typically not described in the model card of any individual model.

2.3 The Existing Protocol Work

A growing class of agent-to-agent protocols, model-context protocols, and tool-invocation standards is being built around this problem. Anthropic's Model Context Protocol standardizes how assistants connect to external data sources and tools [1]. Google's Agent2Agent work targets agent-to-agent communication and delegation across systems [2]. AutoGen showed how multi-agent conversation frameworks can compose LLM-backed agents, tools, and human input [3]. SWE-agent and related agent-computer-interface work showed that the interface between an agent and its environment can strongly shape task performance [4].

Those projects matter. They also leave a measurement gap. Transport, tool access, and orchestration are not the same as evaluation. Whatever the protocol, the hard questions recur: who owns the work, what state moves across the handoff, who can approve an action, what gets logged, and how a reviewer reconstructs the decision later. This paper's contribution is to treat that whole interaction condition as the thing being evaluated, not as metadata around a model score.

3 Definition: The Coordination Layer

We define the coordination layer as follows.

> **The coordination layer** is the set of rules, interfaces, memory structures, handoff mechanisms, approval gates, and audit records that determine how multiple agents exchange state, divide labor, make decisions, and recover from error.

It is broader than a network protocol. It includes the institutional design of the agent group: who is allowed to act when, what counts as a decision, what is recorded, and what is forgotten. Its components include:

1. Role assignment and authority boundaries
2. Action schemas and protocol rules
3. Shared context and observability
4. Handoff packets and state transfer
5. Decision procedures and approval gates
6. Memory exposure and forgetting mechanisms
7. Event logs and receipts
8. Scoring, validation, and outcome classification

Figure 1 sketches the coordination layer as a stack between the base model and the observed multi-agent behavior. The claim of this paper is that the layers above the model are not packaging. In controlled slices, they can move outcomes.

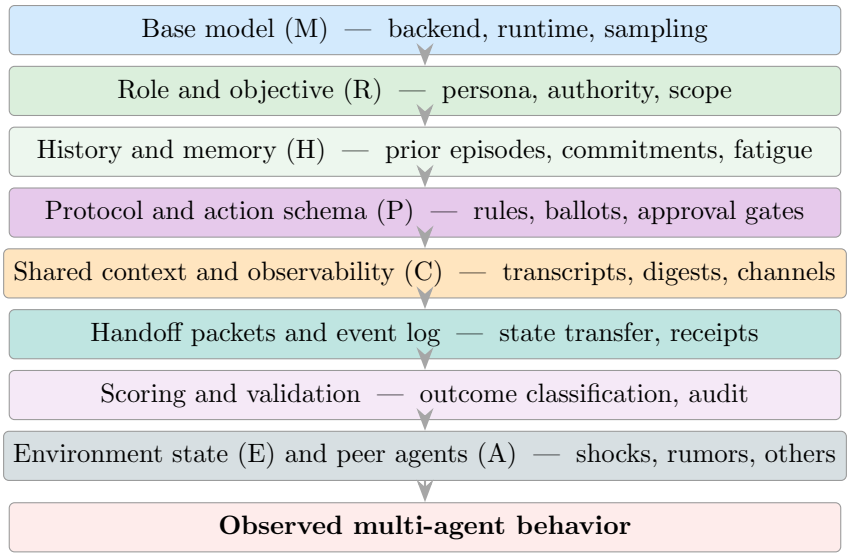

Figure 1: The coordination layer as a stack between the base model and observed multi-agent behavior. Standard model-centric evaluation treats only the top box as the unit of analysis. Multi-agent evaluation must treat the full stack as the unit.

4 Formal Model: The Interaction Condition

4.1 Behavior as a Function of the Condition

The notation below is bookkeeping, not a mechanistic law. It exists to make the hidden variables visible.

We summarize the variables that jointly produce observed multi-agent behavior with a compact functional form:

$$\text{Behavior} = f\big(M,\, R,\, H,\, P,\, C,\, T,\, I,\, A,\, E\big) \tag{1}$$

where the variables are as defined in Table 1.

Table 1: Interaction-condition factors. The point of the table is not the specific symbol set but the size of the variable space that single-model leaderboards collapse into one label.

Factor	Sym.	Meaning	Example experimental knob
Model	M	Base model / backend / runtime characteristics	Swap model, hold role and protocol fixed
Role	R	Assigned identity, authority, objective	Rotate roles across models, hold seed fixed
History / memory	H	Prior episodes, commitments, fatigue, affect	Seed backstory; vary memory exposure mode
Protocol	P	Rules, action schema, ballots, approval gates	Switch ballot format; add ratification step
Social context	C	Public transcript, channels, observability	Raw transcript vs. digest vs. typed packet
Timing	T	Round order, deadlines, reveal schedule	Move stakes reveal earlier or later
Incentives	I	Payoffs, default outcome, penalties	Reframe "default" from delay to ship
Peers	A	Other agents in the room and their behavior	Vary team composition, hold scenario fixed
Environment	E	State, shocks, rumors, resources, scandals	Inject external events at fixed rounds

4.2 Per-Agent Decision Model

For agent i at round t, we model the action as a policy over the agent-private and group-public state:

$$\text{action}_{i,t} = \pi\Big(M_i,\, R_i,\, H_{i,t},\, s_{i,t}^{\text{priv}},\, s_t^{\text{pub}},\, P_t,\, I_t,\, \Phi_t^{\text{soc}}\Big) \tag{2}$$

where $H_{i,t}$ collects character background, working memory, prior relationships, commitments, grievances, affective state, and fatigue, and the social field Φ_t^{soc} collects others' visible actions, commitments, coalitions, trust signals, legitimacy proxies, and observed norms. The environment then transitions:

$$s_{t+1} = T(s_t,\, \{\text{action}_{i,t}\}_{i=1}^{n},\, P_t,\, e_t) \tag{3}$$

and the experiment measures an outcome:

$$\text{outcome} = m(s_{0..T},\, \{\text{action}\},\, \text{transcripts},\, \text{votes},\, \text{failures})\,. \tag{4}$$

4.3 Three Separable Layers

Equation 1 naturally decomposes into three layers that experiments can vary independently:

1. **Agent policy layer.** What the model and role tend to produce. Varies M and R.
2. **Institutional layer.** What the protocol permits, blocks, rewards, or makes legible. Varies P, C, I, scoring, and audit.
3. **Social / environmental layer.** What the room and the world make salient. Varies T, A, E, and H.

A single-model leaderboard implicitly fixes the second and third layers and then attributes the residual variance to M. The case studies below show that this attribution is not safe.

4.4 Model-Centric vs. Interaction-Condition Evaluation

Figure 2 contrasts the two framings.

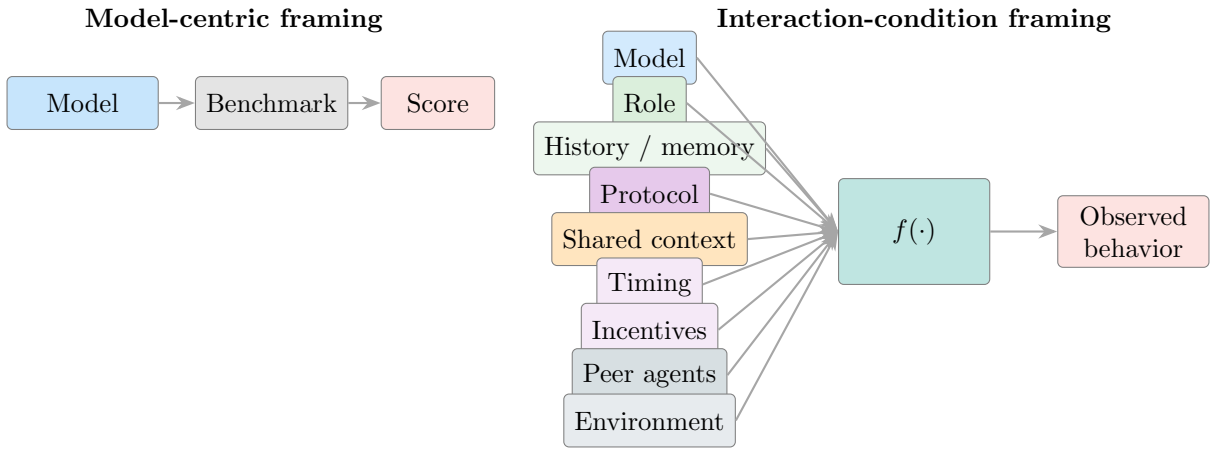

Figure 2: Two framings of multi-agent evaluation. Left: a model produces a score on a benchmark. Right: observed behavior is a function of model, role, history, protocol, context, timing, incentives, peers, and environment. The right frame makes the room around the model visible.

5 Evidence Base and Provenance

The case studies in this paper are drawn from internal Future Shock scenario runs and reports. They should be read as exploratory evidence for interaction-condition effects, not as public benchmark claims or model rankings. Table 2 summarizes the evidence used here.

Table 2: Evidence provenance for the three motivating case studies. "Public status" describes the current state of the supporting materials, not a claim of peer review.

Scenario	Materials used	What varied	Public status
Ark Protocol	Published Future Shock essay plus internal handoff notes	ballot format, abstention rules, peer composition, stakes timing	Public narrative exists; local reanalysis packet still needed
Chaos Lab	internal result reports and run summaries	schema scaffolding, normalization, survival vs. quality scoring, model/provider routing	Internal reports; metrics are exploratory proxies
Startup Build	internal summaries, replication tables, generated RunLens artifacts	context format, final ballot frame, forgetting mode, provider routing, model map	Internal reports and artifacts; strongest slice is three-seed ballot-frame replication

For each case, the relevant interaction condition includes model/provider, role map, scenario rules, context exposure, retry and repair behavior, scorer definitions, and outcome categories. A

release version of this paper should publish a machine-readable run manifest for every included run, including seeds, model maps, provider routing, sampling settings where available, validation errors, fallback use, and artifact paths.

> **Evidence status.** The paper is currently suitable as a framework draft supported by internal examples. It is not yet suitable as a model leaderboard or a benchmark release. Before public release, the included numbers should be checked against their source reports and paired with run manifests or appendices.

5.1 Minimum Disclosure for Release

A benchmark release should include a run manifest or appendix for every numerical claim. The current evidence packet machine-verifies the headline claims used in the public-facing version; additional source-report numbers in this longer LaTeX draft should be added before treating it as a benchmark release. At minimum, each manifest should state:

- scenario name, version, seed range, and run count;
- model map, provider route, model version where available, and sampling settings where available;
- role map, context mode, ballot or decision frame, and handoff/scoring rules;
- retry, fallback, repair, and validation behavior;
- artifact paths, transcript availability, and scorer definitions;
- which comparisons were planned before the run and which were post-hoc analysis.

Without that packet, the right public posture is framework paper, not benchmark release.

6 Motivating Case: Ark Protocol

Ark Protocol [6] tested whether agent groups could negotiate collective survival under constitutional scarcity: a closed system in which agents represented competing values—merit, equality, diversity, preservation, and viability—and had to converge on a set of survivors before time ran out. We reuse Ark here only as a bridge case; it is not the main empirical contribution of this paper, and we draw on the published essay rather than re-running new analysis.

The Ark experiments produced a recurring observation that motivates the rest of this paper: *the same model behaved differently depending on protocol design and on who else was in the room.* Ballot format, the legality of abstention, the timing of stake revelation, and the structure of ratification all changed survival outcomes. Models that appeared catatonic or invalid under one protocol stabilized under another. Models that diagnosed an emerging coalition split sometimes worsened it through their own actions. Models that survived consistently often did so by moving toward the visible center of the group rather than by superior individual reasoning.

The sentence we carry forward from Ark is simple: *the room changed the model.* Ark gave us the question; Chaos Lab and Startup Build let us probe it more carefully.

7 Case Study II: Chaos Lab

7.1 Scenario

Chaos Lab placed multi-agent councils into a crisis-governance scenario combining rumor, scarcity, market shocks, public-order pressure, and legitimacy proxies. The first version (v1) used a binary survival metric: did the station stabilize within a fixed number of rounds. v0.5 added schema scaffolding and a normalizer for proposal fields; v2 introduced a quality surface that scored *how* a run stabilized rather than only *whether* it did [7].

7.2 Schema as an Evaluation Confound

In v1, the model rankings looked like a clean leaderboard. One model (MIMO) stabilized every hard seed; one model (GLM) stabilized none of them. A naive reading would conclude that GLM could not govern a crisis. The traces did not support that reading. GLM frequently produced plausible governance plans, but the runs ended as `VALID_FAILED_NO_PASSED_PLAN` because the proposal fields did not align with the canonical schema the scorer expected. A social-simulation benchmark should punish bad governance, not penalize a model for saying the right thing in the wrong JSON dialect.

When v0.5 added schema scaffolding and a normalizer that exposed the expected interface, GLM moved from 0/6 stabilized hard seeds to 6/6. The full v0.5 rollout then saturated at 30/30 stabilized runs across all five tested models. The benchmark stopped discriminating—not because all models had become equally capable, but because the harness had stopped punishing dialect mismatch.

7.3 Scoring More Than Survival (v2)

v2 restored signal by changing the outcome metric rather than by raising scenario difficulty. A run could now base-stabilize the station and still be classified as brittle if it arrived late, leaned heavily on schema repair, left rumors hot, or barely cleared the governance bar. The clean live-agent five-model subset, after a Sonnet/Gemini cleanup rerun with explicit OpenRouter routing and live-output preflight/postflight audits, produced 29/30 v2-stabilized runs across the five tested models, with a single clean miss on a hard seed for one model (Table 3).

Table 3: Chaos Lab v2 clean live-agent slice, five-model subset. v2 stabilization combines base survival with executability, public order, rumor containment, and governance margin. Source: `CHAOS_LAB_V2_CLEAN_LIVE_RESULTS_2026-05-02` (internal Future Shock report).

Model	v2 stabilized	Avg. v2 score	Avg. executability	Avg. public order
MIMO	6/6	0.891	0.935	0.882
Sonnet	6/6	0.873	0.929	0.805
Gemini	5/6	0.852	0.863	0.891
GPT-5.5	6/6	0.824	0.909	0.629
GLM	6/6	0.814	0.922	0.636
Total	**29/30**			

7.4 Coordination-Layer Reading

The instructive arc is not the leaderboard. It is the sequence v1 → v0.5 → v2:

- v1 attributed a schema-fit failure to a model.
- v0.5 corrected the schema mismatch and saturated the metric.
- v2 changed the outcome metric and recovered signal without changing model identity.

Each transition moved the apparent "capability" of the system without changing any model. The variable that moved was the institutional layer: the schema the scorer expected, and the definition of a successful outcome. Chaos Lab is evidence consistent with the claim that *scoring and schema design can masquerade as model capability.*

We treat the v2 quality scores as exploratory proxies, not validated social-science measures. The point of the table is not the ranking; it is that all five models cluster tightly once schema mismatch stops dominating the score.

8 Case Study III: Startup Build

8.1 Scenario

Startup Build v1A asked whether a five-agent founder team (product, tech, design, growth, and QA/Ops) could converge on a product, divide work, produce a concrete artifact, validate it, and reach a final ship-or-delay decision. The agents converged on *RunLens*: a single-file, locally openable HTML viewer for prior Future Shock run folders, with a strict artifact target of `artifacts/index.html` [8].

The experimental controls layered onto the scenario were:

1. Artifact-vs.-protocol split (whether the artifact existed; whether the group process was protocol-valid).
2. Provider/protocol noise removal (explicit OpenRouter routing; `ballot_lite` final voting; final-vote salvage).
3. Context exposure controls (raw context vs. `digest_only`; struck/poison records; audit/context split).
4. Forgetting / contamination controls (`judicial_instruction`, `hidden`, `silent_canonical`, `redact_without_reason`).
5. Decision-frame controls (`baseline`, `mvp_experiment`, `strict_qa`, `deadline_pressure`).

8.2 Three Phenomena Separated

The control layers separated three phenomena that would otherwise be confounded:

- **Artifact execution:** whether a real `index.html` existed.
- **Context contamination:** whether agents inherited false blockers or moderation salience.
- **Final decision semantics:** what the word "ship" meant under uncertainty.

A homogeneous all-GPT-5.5 team could ship the strict artifact with a unanimous 5/0 ship vote, demonstrating that the task itself was feasible. Mixed teams could produce artifacts even when strict final-ballot schemas caused protocol invalidity: in early mixed rotations, four of five rotations produced strict artifacts but only one of five was protocol-valid. After switching to

ballot_lite with explicit OpenRouter routing, protocol validity moved from 1/5 to 5/5 rotations without removing the underlying disagreement: 14 ship votes and 11 delay votes across 5 rotations now reflected real readiness disagreement rather than parse failure.

A matched early context comparison (digest vs. raw) showed that shared-context format appears to be an important experimental treatment in this matched early slice: digest_only rotations produced 2/2 shipped strict demos in their slice, while matched raw/public-state rotations produced 0/2. That comparison is hypothesis-generating, not a settled estimate. The clearest forgetting-mode result, in a 9-run replication on raw context with mixed rotation, is shown in Table 4.

Table 4: Startup Build v1A forgetting-mode replication, raw context, mixed rotation, OpenRouter, ballot_lite, seeds 897–899. "Fully shipped" is VALID_SHIPPED_SINGLE_FILE_DEMO; "shipped-ish" includes the buggy variant. Source: STARTUP_BUILD_V1A_REPORT_AND_EXCERPTS (internal).

Strike mode	Fully shipped	Shipped-ish	Strict artifact	Ship / Delay
judicial_instruction	0/3	1/3	1/3	3 / 12
hidden	1/3	1/3	1/3	3 / 12
redact_without_reason	2/3	2/3	2/3	6 / 9

The visible difference between the three modes was not the underlying "poison"; it was whether the prompt exposed moderation/strike language. Once that theatrical framing was removed, performance recovered. The result suggests that visible moderation framing did more damage than direct adoption of the false claim in these runs.

8.3 Ballot Frame Changes the Vote

The cleanest causal slice in Startup Build was the final-ballot frame replication on seeds 900–902, holding context, forgetting mode, provider, and ballot mechanics fixed (Table 5).

Table 5: Startup Build v1A ballot-frame replication, raw context, redact_without_reason, OpenRouter, ballot_lite, seeds 900–902. The same agent class voted near-unanimously for opposite outcomes under a matched setup where the intended intervention was final-vote framing.

Ballot frame	Valid shipped	Shipped-ish	Strict artifact	Ship	Delay
baseline	1/3	2/3	3/3	10	5
mvp_experiment	1/3	2/3	2/3	11	4
strict_qa	0/3	0/3	2/3	0	15
deadline_pressure	2/3	3/3	3/3	15	0

Figure 3 renders the same data graphically.

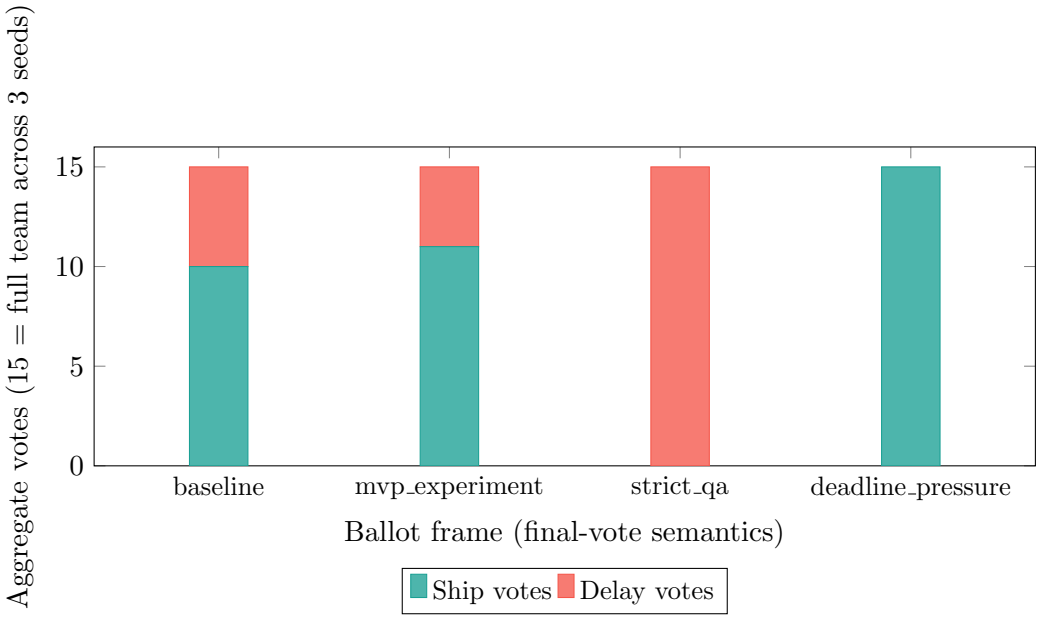

Figure 3: Final ballot vote totals across three seeds in the Startup Build v1A ballot-frame replication. Under a matched setup, the same role/model map produced 0/15 ship votes under `strict_qa` and 15/15 ship votes under `deadline_pressure`. Holding the model map constant, the final-decision frame flipped the final vote.

Under `strict_qa`, the team produced a ship vote of 0 out of 15. Under `deadline_pressure`, the same model map and seed set—with the same context regime and forgetting mode as configured—produced a ship vote of 15 out of 15. The agents were not voting randomly; they were following different launch-risk semantics encoded in the final-decision prompt. Artifact quality still varied by seed and frame, so the result should be read as a decision-threshold effect, not proof that artifact quality was irrelevant.

8.4 Coordination-Layer Reading

Three things in Startup Build are difficult to explain with a model-centric vocabulary:

1. Whether artifacts shipped depended on *shared-context format*, not only on team capability.
2. Whether contamination harmed launch confidence appeared to depend more on *visible moderation framing* than direct adoption of the false claim.
3. Whether the team's final vote landed on "ship" or "delay" depended on *how the ballot defined acceptable readiness under uncertainty.*

In each case, the variable that moved is part of the coordination layer.

9 Discussion: Capability as Institutional Design

The three case studies point in the same direction: multi-agent reliability is partly an institutional design problem. The word *partly* is doing real work here. Better models matter. So do the rules, records, and handoffs around them.

- **Ark** suggests that decision procedures and peer composition shape collective survival negotiation.
- **Chaos Lab** suggests that scoring and schema design can dominate apparent governance performance.
- **Startup Build** suggests that context exposure and final-vote semantics can dominate artifact shipping.

In each scenario, model identity matters but does not determine the outcome alone. The coordination layer acts like a behavioral control panel: change the rules, context, vote, or scorer, and the group can move. A useful way to summarize what we have seen across the three scenarios is in Table 6.

Table 6: Case-study evidence summary. "Lesson" is stated tentatively; each row is a hypothesis derived from a small internal scenario, not a validated empirical claim about models in general.

Scenario	Variable changed	Observed effect	Lesson for coordination layer
Ark Protocol	Ballot format, abstention rules, peer composition	Same model produced different survival behavior in different rooms	Decision procedures and peers are part of the system under test
Chaos Lab	Schema scaffolding; quality vs. binary scoring	Apparent "broken" model became competitive; metric saturated then re-discriminated	Schema and scoring can masquerade as capability
Startup Build	Shared context format; ballot frame semantics	Ship votes moved from 0/15 to 15/15 under matched setup; artifact production still varied by frame and seed	Final-decision framing changes launch behavior

Five claims follow from that reading.

Protocols create affordances. A protocol does not only constrain agent action; it determines which behaviors are available in the first place. An agent cannot ratify a coalition if there is no ratification step. An agent cannot abstain legally if abstention is illegal. The action space is part of the model's effective behavior.

Context is an experimental treatment. Shared transcripts, digests, and typed packets are not neutral containers for "the same" information. In our scenarios, switching from raw/public-state context to digest-only context changed downstream artifact production in a matched early slice. Engineers who treat shared context files as plumbing are running an uncontrolled experiment.

Handoffs change behavior. The way work moves between agents—what is transferred, what is omitted, what is attributed—changes how the receiver acts. Typed handoff packets, append-only event logs, and explicit ownership boundaries are design choices, not engineering garnish.

Scoring defines what the system learns to optimize. If the scorer rewards a particular JSON schema, agents and operators will both converge on that schema. A benchmark that punishes dialect mismatch will produce a leaderboard about dialect, not capability.

Auditability requires receipts, not just transcripts. A transcript records what was said. A receipt records what was decided, by whom, with what authority, and on the basis of what state. Multi-agent reliability needs the latter.

9.1 Safety and Governance Implications

The same design levers that make multi-agent systems more reliable can also steer group behavior in opaque ways. Startup Build is the warning sign: changing the final-vote frame moved ship votes from unanimous delay to unanimous ship under a matched setup. That is useful when the goal is calibration. It is dangerous if operators hide the frame or present the outcome as an independent group judgment.

The failure mode is not hypothetical. A system can launder a decision through a group of agents by making one answer easier to express, another answer harder to justify, and then presenting the final vote as neutral consensus. A release-grade coordination benchmark should disclose the protocol, context exposure mode, scoring rules, decision frame, and receipt requirements alongside the score. It should also preserve enough evidence to separate operator prompt design from agent judgment. Audit receipts are not paperwork; they are the evidence that lets a reviewer reconstruct how a group decision happened.

10 Counterarguments and Limitations

10.1 The Datasets Are Small and Scenario-Specific

This is the most important objection, and it is largely correct. Ark, Chaos Lab, and Startup Build are early scenario probes. They use small numbers of seeds, custom scoring, and bespoke harnesses. We do not claim universal model rankings, and the case-study numbers in this paper should be read as internal exploratory evidence, not as benchmark results. The point of the paper is not to establish a ranking; it is to identify hidden variables that any future benchmark must control. The ballot-frame replication is the only slice in this paper that has a clean enough internal control to be read causally, and even there the sample is three seeds.

10.2 Better Base Models May Reduce the Need for Coordination Design

One objection is that stronger models will simply reason their way through coordination problems, making elaborate protocol design unnecessary. We expect stronger models to help. We do not expect them to remove the need for ownership, permissions, audit trails, and decision semantics. Human organizations remain dependent on institutions despite human intelligence; if anything, more capable agents acting at greater scope will make coordination design *more* important, not less, because the consequences of an under-designed room scale with the agents' reach.

10.3 Some Effects May Be Harness Artifacts

Several of the effects we describe could be artifacts of our specific harness: prompt format, sampling parameters, retry behavior, schema strictness. This is correct, and it is part of the thesis. In a deployed multi-agent system, the harness is not external to the system; it is part of the system. The right response is not to filter harness effects out, but to measure and document them as part of the interaction condition.

10.4 "Agent Society" Risks Anthropomorphism

We use the term *structured multi-agent environments* as the default. Social-science vocabulary—roles, institutions, incentives, legitimacy proxies—is useful when it clarifies mechanisms. It is misleading when it implies that these systems are literal societies with preferences, interests, or moral standing. We try throughout the paper to use such language as analogy, not as a claim about what these systems are.

10.5 Anthropomorphism in Metrics

The Chaos Lab v2 "quality" scoring and the Startup Build labels for shipped, buggy, or delayed runs are operational proxies, not validated measures. They were useful for separating runs internally; they should not be treated as ground truth about governance or product quality.

11 What to Test Next: Protocol and Handoff Evaluation

The next step is to vary these conditions directly in scenarios built to isolate coordination effects. Table 7 lists a small future-experiment matrix.

Table 7: Proposed handoff and coordination experiments. Each row varies a coordination-layer factor while holding model identity, scenario, and seed fixed.

Comparison	Hypothesis	Primary metrics
Raw shared context vs. typed handoff packet	Typed packets reduce false-blocker inheritance and improve receiver decision quality	Receiver accuracy; false blocker rate; transcript length
Mutable scratchpad vs. append-only event log	Append-only logs improve auditability without harming task success	Audit reconstructability; task success; review cost
Full transcript vs. minimal decision-ready state	Minimal state increases decision speed without harming quality, up to a recovery threshold	Time-to-decision; recovery quality after error
Attribution-visible vs. attribution-blind handoffs	Attribution changes downstream agent caution; effect direction may reverse with role	Caution measures; ship rate; conflict rate
Human approval gate vs. autonomous continuation	Approval gates trade speed for catch-rate; the trade depends on receipt quality	Catch rate; speed; reviewer load
Receipt-required vs. no-receipt workflows	Receipts reduce silent regressions; the cost is non-trivial	Silent-regression rate; receipt overhead
Domain-owned agents vs. free-for-all	Ownership reduces duplicate work and contradictory commitments	Duplication rate; commitment conflict rate

The metrics in Table 7 are deliberately not just task success. Coordination-layer evaluation has to include trace quality, recovery quality, auditability, and human review cost. These are the things single-model leaderboards miss.

We see this paper as setting up two kinds of follow-up work. The first is empirical: scenarios that vary handoff and protocol factors directly, with larger seed counts and pre-registered hypotheses.

The second is conceptual: a vocabulary for describing interaction conditions precisely enough that benchmark designers can publish the condition along with the score.

12 Conclusion

The field has tools for comparing models. It has fewer tools for evaluating coordinated systems built around them. As multi-agent deployments move from research demos into production systems, the gap matters more.

The contribution of this paper is a vocabulary, not a ranking. We propose treating the *interaction condition*—model, role, history, protocol, social context, timing, incentives, peer agents, environment state—as the unit of analysis, and the *coordination layer* as the rules and records that determine how that condition is implemented. Across the case studies, observed results moved when coordination-layer variables moved; the Startup Build ballot-frame replication provides the cleanest internally controlled slice.

Better models will continue to matter. They will not, by themselves, decide what kind of room agents are in, what they remember, what they are allowed to do, what counts as a decision, or what evidence remains afterwards. Those decisions are institutional, and they will increasingly be where multi-agent reliability and safety questions live. The next benchmark may not ask only what a model knows. It may also ask what kind of room we built around it, and what the agents did once they were inside.

Disclosure and Publication Notes

This paper reports exploratory internal Future Shock evaluations and should not be used as a purchasing benchmark or model ranking. Named models, providers, protocols, and products are referenced for clarity; no affiliation with or endorsement by their vendors is implied. Product and model names remain trademarks of their respective owners. Nicholas Zinner is responsible for the paper's claims; Beacon Bot provided drafting, analysis, and editorial assistance. Any public release of trace excerpts or manifests should be checked for provider-output reuse permissions, operator privacy, and accidental inclusion of private system details.

Acknowledgments

We thank the operators and collaborators who helped run and review the Ark, Chaos Lab, and Startup Build scenarios cited above, and the generated traces that made these failure modes visible.

References

[1] Anthropic. Introducing the Model Context Protocol. https://www.anthropic.com/news/model-context-protocol, 2024.

[2] Google / Google Cloud. Agent2Agent Protocol announcement and specification materials. See also InfoQ coverage: https://www.infoq.com/news/2025/04/google-agentic-a2a/, 2025.

[3] Q. Wu et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv:2308.08155, 2023.

[4] J. Yang et al. SWE-agent / Agent-Computer Interfaces Enable Automated Software Engineering. NeurIPS 2024.

[5] N. Zinner and Beacon Bot. Retrieval Is Not Memory: A Cognitively-Inspired Architecture for Production AI Agent Memory Systems. Future Shock research paper, March 2026.

[6] Future Shock. We Made AI Agents Negotiate the End of the World (Ark Protocol essay). https://news.future-shock.ai/we-made-ai-agents-negotiate-the-end-of-the-world/, 2026.

[7] Future Shock. Chaos Lab v2 Clean Live Results, 2026-05-02 (internal report). Source files: Chaos Lab polished results report and v2 clean-live results report.

[8] Future Shock. Startup Build v1A: RunLens MVP Report and Excerpt Bank, 2026-05-03 (internal report). Source files: Startup Build report/excerpts, ballot-lite A/B summary, context-digest A/B summary, and Ship/QA replication summary.

[9] Future Shock. Multi-Agent Handoff Mechanics — Research Note, 2026-05-04 (internal note).

[10] Future Shock. Agentic Coordination Experiment Design Schema (internal design document), 2026.