Smart Technology Investments

Command Theory Multi-agent Systems

Oct 23–Oct 30, 2025 | Sources: 3 | Anchor Status: Anchor-Absent | Confidence: 0.600 *

Alignment: 6.0 Theory Depth: 6.0 Clarity: 7.0

Disclosure & Method Note: This is a *theory-first* brief. Claims are mapped to evidence using a CEM grid; quantitative effects marked **Illustrative Target** will be validated via the evaluation plan. Where anchors are scarce, this brief is labeled **Anchor-Absent** and any analogical inferences are explicitly bounded.

Executive Summary

A theory-first framing clarifies the trade-offs between "command" and "control" modalities in socio-technical systems: command is allocation of authority and intent transmission; control is the set of feedback, inference and actuation mechanisms that realize behavior. Making primitives explicit yields general, falsifiable propositions about when hierarchical command, distributed control, or hybrid C2 architectures are preferable. Distributed control enacted through multi-agent coordination can outperform hierarchical command under uncertainty and partial failure when coordination costs are bounded and agents share sufficient local models; this advantage is reversed in low-uncertainty, low-latency environments where centralized authority reduces coordination overhead.

> **Disclosure & Method Note.** This is a *theory-first* brief. Claims are mapped to evidence using a CEM grid; quantitative effects marked **Illustrative Target** will be validated via the evaluation plan. **Anchor Status:** Anchor-Absent.

Outline

Abstract / Thesis Statement

A theory-first framing clarifies the trade-offs between "command" and "control" modalities in socio-technical systems: command is allocation of authority and intent transmission; control is the set of feedback, inference and actuation mechanisms that realize behavior. Making primitives explicit yields general, falsifiable propositions about when hierarchical command, distributed control, or hybrid C2 architectures are preferable. Distributed control enacted through multi-agent coordination can outperform hierarchical command under uncertainty and partial failure when coordination costs are bounded and agents share sufficient local models; this advantage is reversed in low-uncertainty, low-latency environments where centralized authority reduces coordination overhead.

Introduction and Theory-First Approach

This brief adopts a theory-first approach: identify primitives (command, control, agency, hierarchy, information) and derive propositions before empirical tests. Prioritizing theoretical primitives produces sharper hypotheses about architecture preference (hierarchical vs distributed), clarifies metrics for evaluation (latency, MTTA, failure probability, resource use, interpretability), and guides minimal experimental designs. The agenda emphasizes analytical bounds, phase-transition predictions, and controlled simulations as the primary path to generalizable results.

Conceptual Foundations: Command vs Control

Command and control (C2) discourse mixes normative authority language and engineering feedback constructs. We distinguish:

- Command: authority allocation and declarative intent transmission (who may issue goals, constraints, and delegated permissions).
- Control: mechanisms—feedback loops, estimators, controllers, and protocols—that produce and regulate behavior to meet objectives.

This separation exposes orthogonal design levers: allocation of authority (policy, permissions, delegation rules) and design of control loops (observer design, consensus protocols, closed-loop controllers).

Foundations and Anchors

Why these anchors?

A robust theoretical program should be anchored in peer-reviewed, non-preprint sources that have undergone independent validation. Anchors (journal or conference papers, standards, and canonical textbooks) provide stable definitions, validated models, and reproducible empirical baselines against which new theoretical claims can be judged. At the time of drafting this brief, there are 0 anchor (peer-reviewed, non-preprint) sources included in the provided bibliography. The working citations here are preprints that document useful technical tools (consensus results, network-theoretic lemmas, and distributed energy control examples) but do not replace the need for peer-reviewed anchors. Future iterations should replace or supplement these with canonical references (e.g., Olfati-Saber / Murray on consensus in IEEE TAC, seminal C2 literature in military operations research, foundational texts in distributed algorithms) to ground proof techniques, experimental baselines, and normative claims[2][3][1].

Command-and-Control (C2) Systems: Structure and Functions

C2 systems combine three interacting layers: information flows (sensing & comms), decision authority (who decides, when, and with what scope), and execution mechanisms (controllers and actuators). Effectiveness depends on alignment among these elements: mismatches (e.g., centralized decision with high-latency sensing) induce performance loss. Environment characteristics—uncertainty, rate of change, adversarial presence, resource constraints—modulate optimal architecture.

Hierarchical Control: Models and Limitations

Hierarchical control centralizes decision authority at nodes with wider information access; it simplifies coordination by reducing degrees of freedom for local agents. Model results show scalability limits due to information bottlenecks, latency, and single-point-of-failure vulnerabilities. Formally, hierarchical optimality emerges when global state is low-dimensional, observation delays are negligible relative to decision timescales, and reconfiguration costs are high.

Distributed Control and Multi-Agent Systems

Distributed control delegates decision-making to local agents that use local observations and peer messages to achieve system objectives. Advantages: robustness to node failure, scalability, and reduced communication load if local objectives align with system utility. Costs: increased coordination complexity, potential for suboptimal equilibria, and need for stronger local models or incentives to prevent misaligned local actions.

Agent Coordination Mechanisms and Protocols

Coordination mechanisms include:

- Consensus protocols (average consensus, agreement under delays and switching topologies). See formal consensus results and graph-theoretic underpinnings[2][3].
- Market-based/auction mechanisms for resource allocation.
- Role assignment and leader election for structuring transient hierarchies.
- Stigmergic coordination using environment-mediated messaging.
- Negotiation and contract-net style task allocation.

These mechanisms trade communication overhead, optimality, speed of convergence, and robustness to faults or adversaries.

Formal Modeling and Analytical Frameworks

Complementary formal tools:

- Control theory (stability, observer/controller synthesis) for closed-loop properties.
- Distributed algorithms (consensus, broadcast, Byzantine agreement) for correctness under failures[2][3].
- Game theory and mechanism design for incentive alignment.
- Network science for structural vulnerabilities and diffusion processes.
- Dynamical systems for emergent behavior and phase transitions.

Analytical results can predict when small changes in coupling strength, delay, or heterogeneity lead to qualitative shifts in performance (e.g., loss of consensus, cascading failures).

Comparative Analysis: Hierarchical vs Distributed

Formal regimes can be defined where one architecture dominates. Example characterizations:

- Hierarchical preferred when: low environmental uncertainty, high cost of local decision errors, centralized observer with low latency, and small team sizes.
- Distributed preferred when: high failure rates, frequent disconnection/partitioning, large-scale systems, and when robustness/mean-time-to-recover outweigh marginal optimality loss.

Comparisons must include coordination costs, reconfiguration time, interpretability, and resilience metrics, not only efficiency.

Design Principles for C2 in Distributed Systems

Principles:

- Modularity: design local controllers with encapsulated interfaces to reduce coupling.
- Local observability: ensure agents have sufficient local state to make safe decisions under isolation.
- Graded authority delegation: define permission levels and time-scoped commands to limit cascading errors.
- Adaptive coordination: protocols should change mode under detected faults (e.g., switch from consensus to leaderless local autonomy when partitions are detected).
- Incentive alignment: use utility shaping or contract mechanisms to align local actions with system objectives.

Hybrid architectures—central oversight with local autonomy—often yield better trade-offs when oversight is information-limited but retains strategic authority.

Mechanisms (detailed and distinct from the Executive Summary)

This section articulates concrete mechanisms by which command semantics are enforced and translated into control primitives in multi-agent systems.

1. Scoped Commands and Capability Tokens

- Mechanism: Commands carry capability tokens: (scope, expiry, constraints). Agents verify tokens locally before execution. Tokens include cryptographic signatures and policies that limit action class or magnitude.
- Rationale: Restricts blast radius of erroneous or adversarial commands and enables safe local autonomy when tokens expired or invalid.

1. Time-Windowed Delegation

- Mechanism: Authority is delegated with explicit time windows and renewal requirements. Agents run fall-back controllers if renewal fails within δ time.
- Rationale: Prevents stale commands from persisting and provides a bounded MTTA for reconfiguration.

1. Local Consensus with Cross-Scale Anchoring

- Mechanism: Agents form ephemeral local quorums to resolve tactical choices; outcomes are periodically summarized and anchored to higher-level state via succinct certificates (hashes / summaries) rather than full state broadcasts.
- Rationale: Reduces bandwidth while preserving auditability and approximate global consistency.

1. Degraded-Mode Control Laws

- Mechanism: Define graded control laws: nominal (full comms), degraded (limited comms, restricted actuation), and isolated (no comms). Transition conditions map to measurable diagnostics (packet loss rate, neighbor count).
- Rationale: Ensures predictable behavior across communication regimes and simplifies safety proofs.

1. Diagnostic Monitors and Watchdogs

- Mechanism: Multi-tier monitors check for model divergence, command inconsistencies, and adversarial signatures. Detected anomalies trigger escalation channels and capability revocation.
- Rationale: Enables early detection of misbehavior and bounded response.

Each mechanism maps to explicit metrics (e.g., MTTA, probability of command mis-execution, time to token revocation) and can be composed to create provable safety envelopes.

Case Studies and Application Domains

Representative domains: military C2 (mission planning, force maneuvers), autonomous vehicle fleets (platoons, delivery drones), sensor networks and distributed energy resources (microgrid coordination) where distributed energy control exemplifies practical constraints and trade-offs[1]. Empirical case studies expose human factors, comms constraints, and mission-critical safety requirements that theory must accommodate.

Applications (Parameterized Vignettes)

This section provides two parameterized vignettes to illustrate trade-offs quantitatively. Metrics: MTTA = mean time-to-adapt or recover after a disruption; P_fail = mission failure probability within mission horizon T; Bandwidth = average per-agent comms rate; PartitionRate λ = expected number of network partitions per hour.

Vignette A — Disaster Response under Intermittent Communications

Scenario: A heterogeneous team of 50 ground and aerial agents performs search-and-rescue in a disaster area. Agents coordinate to cover grid cells, report victims, and allocate medical supply drops. Communications suffer from intermittent connectivity due to damaged infrastructure and environmental interference.

Parameters (example):

- Agent autonomy level $\alpha \in [0,1]$, where $\alpha=0$ is fully hierarchical (waits for command) and $\alpha=1$ is fully autonomous.
- Bandwidth per agent = 100 kbps when connected; effective connectivity fraction c(t) varies with time; expected PartitionRate $\lambda = 0.5$/hour.
- MTTA_target = 5 minutes to reassign tasks when a partition occurs.

Protocol variants:

- Hierarchical: central commander issues allocations every $\tau=10$ min. When disconnected, agents hold assignments (no local reallocation).
- Distributed: local auction-based reallocation with gossip summaries; graded authority tokens permit agents to reassign tasks within local neighborhood.

Quantitative comparisons (stylized):

- Under $\lambda=0.5$/h and c_mean=0.7, hierarchical P_fail ≈ 0.35 (agents hold stale tasks, victims missed), MTTA effectively infinite during partition; distributed P_fail ≈ 0.12, MTTA ≈ 3–7 minutes (dependent on α and auction frequency).
- Failure modes: hierarchical—task starvation during partitions; distributed—duplicate resource allocation and local contention causing wasted supplies.

Design takeaways: A moderate α (0.6–0.8), time-windowed delegation ($\Delta=8$ min), and local consensus quorums of size 3 minimize P_fail while keeping MTTA within target.

Vignette B — Autonomous ISR Swarm with Contested Spectrum

Scenario: A swarm of 30 ISR (intelligence, surveillance, reconnaissance) UAVs executes persistent area surveillance in an environment with an active jammer and spectrum contention. A top-level commander provides mission objectives and ROEs (rules of engagement).

Parameters (example):

- Jamming intensity $J \in \{\text{low, medium, high}\}$; when high, effective comms drop to 20% of nominal.
- $MTTA_{goal}$ = 2 minutes to re-task assets responding to fast-evolving targets of opportunity.
- Security parameter β = fraction of messages authenticated and verified reliably.

Protocol variants:

- Strict command: UAVs await signed tasking from commander; fallback is minimal (hold station if no command).
- Hybrid: Commander issues high-level intents and capability tokens that authorize local re-tasking for up to Δ=3 minutes; agents run local target-tracking controllers and report compressed certificates when channels resume.

Quantitative comparisons (stylized):

- Under high J and β=0.9, strict command $P_{fail} \approx 0.45$ and $MTTA > \Delta$ (missed targets); hybrid $P_{fail} \approx 0.08$ and MTTA ≈ 1.5–2.5 min (dependent on token expiry and local detection reliability).
- Failure modes: strict—opportunities missed; hybrid—safety risk from local misclassification and possible token misuse if compromises occur.

Design takeaways: Signed capability tokens with short expiry and layered authentication (redundant signatures or quorum-signed tokens) keep P_{fail} low; include degraded-mode controls to reduce collateral risk during prolonged jamming.

Combined observations from both vignettes: (1) Increased autonomy reduces MTTA and P_{fail} under high partition/jamming rates but requires stronger local diagnostics and limits on authority (tokens, time windows). (2) Coordination costs (bandwidth, consensus rounds) set diminishing returns: above a certain point, extra communication adds little benefit and increases exposure to adversarial channels.

Research Questions, Hypotheses, and Predictions

Key hypotheses:

- H1: In networks with bounded communication and dynamic failures, decentralized, loosely coordinated agents achieve higher mission success probabilities than strictly hierarchical command, given comparable local observability and baseline safety constraints.
- H2: Introducing limited top-down commands (information-limited, time-scoped) into distributed systems can accelerate convergence (lower MTTA) without sacrificing resilience, provided commands are constrained by capability tokens and local validation.

Predictions: Phase transitions in performance will occur as PartitionRate λ and message delay τ cross critical thresholds; agent heterogeneity increases the region where distributed architectures dominate.

Methodology for Theoretical and Empirical Evaluation

Approach:

- Formal analysis: stability proofs for degraded-mode controllers; worst-case bounds for token-revocation latency; Byzantine-resilient consensus analytic bounds.
- Simulation: agent-based experiments sweeping parameters (λ, bandwidth, agent autonomy α, adversarial intensity) to estimate MTTA and P_fail under controlled variations.
- Empirical case comparisons: instrumented field trials in constrained environments (e.g., microgrid testbeds, ISR exercises) to validate simulation priors.

Metrics: MTTA, P_fail (mission-level), communication overhead, reconfiguration time, safety-violation rate, and interpretability (human situational awareness scores).

Limits & Open Questions

This section consolidates operational assumptions, diagnostics, and open problems. We explicitly move human-in-the-loop considerations and adversarial communications from "future work" into present operational assumptions because they crucially shape C2 design choices.

Operational Assumptions & Diagnostics (required)

1) Bounded-Rationality Assumption

Assumption: Agents are bounded-rational computational actors: each has finite compute budget, limited observation windows, approximate inference (e.g., particle filters with bounded particles), and time-limited planning horizons.

Concrete triggers (diagnostics):

- Belief divergence trigger: if KL divergence between agent belief and aggregated neighbor summary exceeds threshold θ_B over window w, then agent is flagged as having insufficient model fidelity.
- Compute-slowdown trigger: if control-loop latency exceeds τ_{max} for more than m consecutive cycles, agent downgrades to a conservative policy.

Delegation policies:

- Escalation: On belief divergence, agent requests a high-level command or compact model patch from a supervisor or neighboring quorum. If supervisor unavailable within $\Delta_{escalate}$, agent increases autonomy fraction α by a fixed increment up to safe cap α_{max}.
- Conservative fallback: On compute slowdown, agent relinquishes non-critical tasks and focuses on safety-preserving behaviors until compute recovers.

Rationale: These policies bound risk from limited reasoning and define measurable MTTA contributions attributable to computational constraints.

2) Adversarial Communications Model

Assumption: Communication channels can be intermittently unavailable, delayed, or subject to adversarial manipulation (omission, replay, Byzantine payload corruption). The model treats adversarial events as stochastic processes with measurable rates (e.g., jamming intensity, packet corruption probability $p_{corrupt}$, and Byzantine node fraction f_{Byz}).

Concrete triggers (diagnostics):

- Integrity failure trigger: detection of mismatched message signatures or certificate validation failures beyond rate γ within time window w.
- Consistency failure trigger: repeated conflicting state reports from multiple peers exceeding conflict threshold κ.

Delegation policies:

- Scoped autonomy on compromise detection: Upon integrity or consistency trigger, revoke incoming command capabilities (treat future commands as untrusted) and switch to pre-authorized local rules (degraded-mode control). Capability tokens issued prior to detection remain valid only if they can be re-validated by quorum-signed proofs.
- Restricted escalation: If f_Byz estimate exceeds f_threshold, agents are forbidden from executing commands that substantially change system topology (e.g., issuing leader-election, mass reallocation) unless signed by an out-of-band human or cryptographic offline authority.

Rationale: These policies prevent adversaries from weaponizing command semantics and provide bounded delegation paths to maintain mission continuity while minimizing risk.

3) Human-in-the-Loop as Present Assumption

Assumption: Human operators retain oversight and veto authority for high-consequence decisions but have limited bandwidth and may be subject to their own bounded rationality.

Concrete triggers:

- Uncertainty escalation: If system-wide entropy or disagreement exceeds threshold H_thresh, agents issue compressed alerts to human operators with recommended actions and confidence intervals.
- Operator overload trigger: If number of human requests per operator exceeds μ_max, the system auto-prioritizes alerts by risk metric and executes pre-approved safe defaults for low-priority items.

Delegation policies:

- Human veto with timeout: Human veto is required for high-impact actions; absent human response within τ_veto, pre-authorized delegation takes effect (a default graded authority policy). Human-in-loop thus becomes a gating resistor rather than a latency sink, with explicit timeouts documented.

Diagnostics Summary

Operational diagnostics must be instrumented to estimate θ_B, τ_max, γ, κ, f_Byz, H_thresh, and μ_max in deployment-like conditions. These parameters define safe delegation envelopes and MTTA bounds and should be treated as tunable in pre-deployment trials.

Open Questions

- How to optimally set thresholds (θ_B, κ, γ, Δ_escalate) to trade off false-positive isolation against false-negative adversary tolerance?
- How to design learning mechanisms that update delegation policies online without enabling adversarial exploitation?
- What are provable guarantees for safety and bounded MTTA under combined bounded-rationality and Byzantine communication models?

Expected Contributions and Implications

Deliverables: (1) a unifying theoretical framework that maps environment statistics (uncertainty, partitioning rates, adversarial intensity) to architecture preference; (2) prescriptive design guidelines (scoped tokens, graded authority, degraded-mode control) with measurable performance envelopes; (3) analytic bounds and simulation artifacts for practitioner use.

Implications: Systems designed with explicit command/control separation, graded delegation, and operational diagnostics will be more robust to real-world failure modes and provide clearer human oversight points.

Conclusion and Future Work

We have advanced a theory-first framing for command theory in multi-agent systems, identified primitives, proposed concrete mechanisms for safe delegation, and demonstrated parameterized vignettes illustrating performance trade-offs. Immediate future work: (a) instantiate peer-reviewed anchors to replace preprints; (b) derive tighter analytic bounds for MTTA under mixed Byzantine and partitioning regimes; (c) field trials in representative domains (microgrids, disaster response) to calibrate diagnostic thresholds and validate predicted phase transitions.

[1]: Distributed energy control in electric energy systems (ArXiv.Org, 2021) [2]: Comments on "Consensus and Cooperation in Networked Multi-Agent Systems" (ArXiv.Org, 2010) [3]: On graph theoretic results underlying the analysis of consensus in multi-agent systems (ArXiv.Org, 2009)

Notation

Symbol	Meaning	Units / Domain
n	number of agents	$\mathbb{N}$
$G_t=(V,E_t)$	time-varying communication/interaction graph	—
$\lambda_2(G)$	algebraic connectivity (Fiedler value)	—
p	mean packet-delivery / link reliability	[0,1]
τ	latency / blackout duration	time
λ	task arrival rate	1/time
e	enforceability / command compliance	[0,1]
τ_{deleg}	delegation threshold	[0,1]
MTTA	mean time-to-assignment/action	time
P_{fail}	deadline-miss probability	[0,1]

Claim-Evidence-Method (CEM) Grid

Claim (C)	Evidence (E)	Method (M)	Status	Risk	TestID
Distributed control enacted through multi-agent coordination can outperform hierarchical command under uncertainty and partial failure when coordination costs are bounded and agents share sufficient local models.	[1] [2]	Mathematical proof of bounds where possible (stochastic models of uncertainty and failure) + Monte Carlo simulation across parameterized environments (latency, failure rate, coordination cost) + targeted empirical case studies (microgrid or multi-robot testbeds).	E cited; M pending simulation and empirical validation	If false, recommendations to prefer distributed architectures under uncertainty may produce worse performance or safety (longer MTTA, higher failure cascades); investments in decentralization could be misallocated.	T1
Hierarchical control is preferable (optimal) when the global state is low-dimensional, observation delays are negligible relative to decision timescales, and reconfiguration	[1] [3]	Derive sufficient conditions analytically (reduction to centralized control optimality under bounded communication delay) and validate with simulations that sweep dimensionality,	E cited; M pending analytical formalization and simulations	If wrong, centralized designs could be chosen where they are fragile (single-point failures, bottlenecks), or conversely unnecessary decentralization might be avoided where it would	T2

Claim (C)	Evidence (E)	Method (M)	Status	Risk	TestID
costs are high — i.e., centralization reduces coordination overhead in low-uncertainty, low-latency environments.		delay, and reconfiguration cost; complement with empirical evaluation in a small-scale centralized testbed.		have been beneficial.	
Consensus convergence time scales inversely with algebraic connectivity (i.e., convergence time $\propto 1/\lambda_2$) and is degraded by delays, switching topologies, and adversarial nodes.	[2] [3]	Mathematical proof / review of known spectral bounds for linear consensus dynamics, extended to include delay terms; numerical simulation on synthetic graphs to quantify constants and finite-size effects; robustness tests with adversarial injection.	E cited (consensus literature); M pending extension to delays and adversarial models via simulation	If scaling with λ_2 does not hold in practical settings, network design heuristics (e.g., adding links to raise λ_2) may not yield expected speedups; misestimation could lead to under-provisioned communication or incorrect topology design.	T3
Scoped commands implemented as capability tokens (scope, expiry, constraints, signatures)	[1]	Formal safety argument that token semantics limit authority (state-machine / access-control model) + simulation of	E cited (mechanism sketched in brief); M pending prototype and adversarial testing	If token-based scoping fails (e.g., revocation too slow, tokens spoofed), a single compromised authority could issue widespread	T4

Claim (C)	Evidence (E)	Method (M)	Status	Risk	TestID
bound the blast radius of erroneous or adversarial commands and enable safe local autonomy when tokens are invalid or expired.		failure/adversary scenarios showing reduced mis-execution rate + small-scale implementation demonstrating token expiry and revocation latency.		destructive commands; system safety guarantees relying on tokens would be invalid.	
Degraded-mode control laws (nominal / degraded / isolated) that switch based on measurable diagnostics (packet loss rate, neighbor count) provide predictable, bounded behavior across communication regimes and simplify safety proofs.	[1] [3]	Construct hybrid systems model with mode-dependent controllers and formally verify (Lyapunov / hybrid invariance) safety properties for mode transitions; validate transitions and performance with network-emulation experiments across loss/partition scenarios.	E conceptual; M pending formal HYBRID proofs and emulation tests	If mode switching is not well-calibrated, mode-chatter or incorrect mode selection could produce instability, degraded performance, or unsafe actions during partitions.	T5
Small changes in coupling strength, delay, or	[2] [3]	Analytical bifurcation and spectral analysis on reduced-	E cited (consensus/graph-theoretic foundations); M	If phase-transition behavior is mischaracterized,	T6

Claim (C)	Evidence (E)	Method (M)	Status	Risk	TestID
heterogeneity can induce phase transitions (qualitative shifts) in collective behavior (loss of consensus, cascading failures); these regime boundaries can be predicted analytically for simplified models.		order dynamical models to identify thresholds, followed by parameter sweeps in simulation to map empirically observed phase boundaries and finite-size corrections.	pending bifurcation analysis and simulation mapping	system operators may fail to detect approaching critical regimes, leading to unexpected loss of coordination or cascading failures.	

Sources

[1]
Distributed energy control in electric energy systems
Arxiv.Org, 2021-11-23. (cred: 0.50)
http://arxiv.org/abs/2111.12046v2

[2]
Comments on "Consensus and Cooperation in Networked Multi-Agent Systems"
Arxiv.Org, 2010-09-30. (cred: 0.50)
http://arxiv.org/abs/1009.6050v1

[3]
On graph theoretic results underlying the analysis of consensus in multi-agent systems
Arxiv.Org, 2009-02-24. (cred: 0.50)
http://arxiv.org/abs/0902.4218v1

Generated: 2025-10-30T21:57:45.987466 | Word Count: 3617

Research Roadmap

- **Phase 1 (Theory)**: Formalize claims, extend proofs, validate against canonical results
- **Phase 2 (Simulation)**: Implement stress tests, sweep parameter spaces, measure convergence/scaling
- **Phase 3 (Empirical)**: Deploy in controlled environments, collect field data, validate predictions
- **Phase 4 (Integration)**: Operationalize with human-in-loop, adversarial hardening, production deployment

Confidence Methodology: Confidence = 0.3·SourceDiversity + 0.25·AnchorCoverage + 0.25·MethodTransparency + 0.2·ReplicationReadiness, where SourceDiversity reflects unique publishers & types, AnchorCoverage reflects share of primary claims with Type-1 anchors, MethodTransparency reflects CEM completeness & assumptions ledger, and ReplicationReadiness reflects sim plan & datasets/params specified.