

Coordination Boundaries and Incentive Governance in Distributed Data Centers

Anonymous

16 October 2026 | Version 1.0.0

Abstract

We formalize distributed capability through executable protocols with preservation and progress contracts. We prove transport and composition results, metric descent, and incentive separation. A presheaf infinity-topos distinguishes contextual correctness from global witnesses. Finite artifacts exhibit coordination failures and behavioral shifts under view-based rewards.

Keywords: distributed computation; safety; liveness; infinity-topos; contracts; incentives; observability.

1. Thesis and scope

A data center operates through distributed storage, communication, scheduling, and control. Service guarantees depend on coordination. Programmable ranking, allocation, and payments also influence participants. A feasible service need not induce desirable behavior; changing a reward need not change computational solvability.

Our thesis is relative: **given hardware, local operations, observations, resources, failures, and timing assumptions, guaranteed capabilities are the tasks realized by executable protocols satisfying their contracts.** This is a semantic definition, not a decision procedure from the labels preservation and progress.

We synthesize distributed semantics [1-6], topos semantics [7], and incentive analysis [8-10]. We contribute the combined framework and worked application. The transport, metric descent, and incentive separation lemmas are elementary. Impossibility results are imported; universal human control is not claimed.

2. Executions, preservation, and progress

Definition 2.1 (environment). Fix nonempty inputs X and an environment E : processes, local observations, executable primitives, network/storage semantics, failures, scheduling assumptions, and resources. Let Prog_E be its executable local protocols. A process acts only on its local history; an omniscient global prescription is not thereby executable.

For $P \in \text{Prog}_E$ and $x \in X$, let $\text{Exec}_E(P, x)$ be the nonempty set of admissible maximal executions. Record scheduling and failures in executions. Pad finite terminal or deadlocked executions by stuttering, so every execution has a state sequence $(s_n)_{n \in \mathbb{N}}$. An observation map o yields an

external history h . External assumptions must not silently exclude an internal deadlock. Require an admissible extension for every legal finite execution prefix; explicitly identified terminal cases may stutter.

Fairness assumes scheduling behavior; it does not prove completion. Weak fairness concerns continuously enabled actions; strong fairness concerns actions enabled infinitely often. Resources belong to E : exceeding a budget invalidates an implementation despite abstract correctness.

Definition 2.2 (contract). A task $T = (X, S_T, L_T)$ specifies a safety property S_T and a progress obligation L_T on external histories. Safety has a finite bad prefix whenever it fails [1]. A usual preservation certificate is an invariant I with

$$\text{Init} \subseteq I, \quad R_P(I) \subseteq I.$$

Induction then gives $\Box I$. Not every safety property is already a predicate on the original state: a monitor or history variable can be necessary. A representative progress obligation is

$$\forall j \Box(\text{Req}_j \Rightarrow \Diamond \text{Resp}_j),$$

with request identities and the obligated, for example nonfaulty, clients specified. This need not give a uniform latency bound. We use “progress obligation” broadly; it need not be a dense liveness property in the precise topology of [1]. Safety closure and response progress are separate proof obligations.

Definition 2.3 (realization and capability). Put

$$P \models_E T \iff \forall x \in X \forall \rho \in \text{Exec}_E(P, x), \quad o(\rho) \in S_T \cap L_T,$$

$$\text{Cap}(E) = \{T \mid \exists P \in \text{Prog}_E : P \models_E T\}.$$

Existence requires executable witnesses; nonempty execution sets prevent vacuity. This is universal adversarial correctness. Almost-sure correctness instead quantifies probability-one events for each adversary, with its access to random choices specified.

3. Transport and composition

Theorem 3.1 (environment restriction). Suppose two environments share inputs, protocols, observation maps, and costs, with $\text{Prog}_{E_1} \subseteq \text{Prog}_{E_2}$ and

$$\emptyset \neq \text{Exec}_{E_2}(P, x) \subseteq \text{Exec}_{E_1}(P, x)$$

for every $P \in \text{Prog}_{E_1}$ and x . Then $\text{Cap}(E_1) \subseteq \text{Cap}(E_2)$.

Proof. Reuse the realizing program. Every execution quantified over in E_2 was already covered by its correctness proof in E_1 . The program remains executable and within budget. $\square$

Changing hardware, local information, or budgets is not merely restricting an adversary. Adding an oracle changes the program model. Strengthened assumptions must be declared.

Definition 3.2 (contract-preserving refinement). Q refines P through input map a and complete-history map r if every admissible execution of Q maps to an admissible execution history of P at input $a(x)$. Preserve task safety, progress, obligated clients, and request identities. Infinite internal stuttering cannot complete an outstanding abstract request. Also supply executable code and resource bounds [2].

Proposition 3.3 (refinement transport). If $P \models_E T$ and Q satisfies Definition 3.2, then Q realizes the pullback contract $r^{-1}(S_T \cap L_T)$.

Proof. Map each admissible complete implementation history to the promised admissible abstract history. Abstract correctness implies membership in $S_T \cap L_T$; taking the inverse image proves the assertion. $\square$

Theorem 3.4 (compatible composition). Let an executable assembly Q have component projections π_i . Suppose every admissible execution of Q projects to an admissible complete execution of component P_i , each P_i realizes T_i , and the global contract contains every history satisfying all pulled-back component contracts. Then Q realizes the global contract.

Proof. For any assembly execution, admissibility of each projection allows component correctness to be invoked. Its history belongs to every $\pi_i^{-1}(S_{T_i} \cap L_{T_i})$, hence to the global contract. $\square$

Projection admissibility is a scheduling and interface obligation. Components awaiting each other's first message can deadlock: a projection may violate the assumed eventual arrival of an initiating message. Circular assumptions cannot supply that event. Finite step simulation alone does not establish progress.

4. Coordination obstructions

The Fischer-Lynch-Paterson theorem rules out deterministic binary consensus with agreement, validity, and guaranteed termination in its asynchronous message-passing model with at least two processes and one possible crash [3]. Reliable eventual delivery does not give a known delivery bound. Replication or faster processors alone do not remove that obstruction. Partial synchrony changes assumptions and supports different solvability results [4]; randomization changes the termination guarantee and adversary model.

In the standard wait-free read/write setting, the asynchronous computability theorem characterizes finite task solvability by a chromatic decision map on a suitable protocol subdivision, carried by the task relation [5]. The colors, input/output carriers, and local-view structure are essential; bare equivalence of underlying homotopy types is insufficient.

For illustration, a mixed-input edge for two-process binary consensus has solo-input endpoints forced by validity to different decision components. Its wait-free protocol subdivision remains connected. A continuous decision map into the disconnected consensus output cannot carry both endpoints as required. This is the usual connectivity obstruction under these task hypotheses, not an assertion about all distributed models.

Proposition 4.1 (local invariants need not survive merge). Let replica states be subsets of $\{a, b\}$, merge be union, and the invariant be $|s| \leq 1$. Starting from $\emptyset$, independent reservations can each preserve the invariant while their merge violates it.

Proof. The reachable branch states $\{a\}$ and $\{b\}$ each satisfy the invariant; their union has cardinality two. $\square$

This violates invariant confluence in the merge model [6]. Union of event identifier sets instead preserves nonnegative counting and deduplication, though not honest event production. Coordination requirements depend on invariants and operations.

5. A categorical semantics with explicit witnesses

Choose a small category $\mathcal{C}$ of finite observation contexts, with an initial empty context and arrows toward more informative extensions. The presheaf infinity-topos

$$\mathcal{E} = \text{Fun}(\mathcal{C}^{\text{op}}, \mathcal{S})$$

supports restrictions and space-valued coherence [7]. The finite models use set-valued presheaves among its 0-truncated objects. The context category presents the semantic topos; it is not that topos. No additional sheaf gluing is presumed: imposing a topology requires specifying covers and their observational meaning.

Represent executable protocols by a 0-truncated object Prog and admissible executions by $e : \text{Run} \rightarrow \text{Prog}$. A baseline uses constant presheaves on Prog_E and $\coprod_{P,x} \text{Exec}_E(P,x)$, with e forgetting inputs and executions. Nonconstant contextual models require a specified operational witness interpretation. Let $G \hookrightarrow \text{Run}$ classify contract satisfaction. Define

$$\text{GoodProg} = \forall_e(G) \hookrightarrow \text{Prog}.$$

Here $\forall_e$ is right adjoint to pullback on subobjects: every admissible execution satisfies G . Nonempty fibers remain an environment obligation.

Proposition 5.1 (witness semantics). A specified global code $p : 1 \rightarrow \text{Prog}$ factors through GoodProg exactly when its execution fiber is contained in G .

Proof. Since Prog is 0-truncated, p represents a subobject. The adjunction $e^* \dashv \forall_e$ identifies $p \subseteq \forall_e G$ with $e^* p \subseteq G$, the required fiber condition. $\square$

Internal truth of $\exists p \text{GoodProg}(p)$ says that $\text{GoodProg} \rightarrow 1$ is an effective epimorphism. It need not furnish a global code $1 \rightarrow \text{GoodProg}$. For example, in sheaves on a circle, the sheaf of local sections of the nontrivial double covering is locally inhabited but has no global section. The operational capability definition requires a global executable witness, with its locality and cost certificates.

Finite-limit constructions express invariant intersections and pullbacks. Progress, by contrast, refers to complete executions and temporal quantification. Transporting the latter requires preservation of the corresponding execution objects and quantifiers; left exactness by itself does not assert preservation of arbitrary dependent products or countable limits. We make no claim that passing to an infinity-topos resolves a coordination obstruction.

6. Incentives and behavioral selection

Fix an infrastructure protocol P . Let θ parameterize ranking, access, payments, or metrics, b participant strategies, and z admissible disturbances and schedules. A specified model produces histories $\text{hist}(P, \theta, b, z)$. Response sets may encode bounded rationality or heterogeneity; their empirical

validity requires evidence. Performative prediction [9] studies related feedback; compositional game theory supplies richer interfaces [10].

For finite actions A , let $v(a)$ be expected credited views, $w(a)$ a separately specified desired contribution, and $c(a)$ a cost. One possible utility is

$$U_{\alpha,\beta}(a) = \alpha v(a) + \beta w(a) - c(a), \quad \alpha, \beta \geq 0.$$

Coefficients convert scores to common utility units. Views may yield revenue, status, or allocation: an assumed instrumental payoff. Historical YouTube research modeled expected watch time [8]; this does not establish any platform's current objective.

Theorem 6.1 (incentive separation). Suppose infrastructure preserves its safety and progress contract under either participant action. Let actions a, b have

$$(v(a), w(a), c(a)) = (1, 1, 0), \quad (v(b), w(b), c(b)) = (3, 0, 0).$$

With utility-maximizing selection, b is uniquely selected when $2\alpha > \beta$, a when $2\alpha < \beta$, and both are optimal at equality. Thus an incentive change can reverse selected behavior with the infrastructure protocol and its guarantee unchanged.

Proof. Subtract utilities: $U(b) - U(a) = 2\alpha - \beta$. Its sign determines the maximizing action. Infrastructure correctness is uniform over both actions by hypothesis. $\square$

This proves conditional influence, not universal control. For a population share p using b , an assumed replicator model yields

$$\dot{p} = p(1-p)(2\alpha - \beta), \quad \text{logit } p(t) = \text{logit } p(0) + (2\alpha - \beta)t$$

for $0 < p(0) < 1$. Differentiating verifies the solution. Selection depends on this assumed response law; preferences, learning, outside alternatives, and regulation can change it. Workloads outside E require a new infrastructure proof.

7. What a metric can certify

Theorem 7.1 (metric descent). For histories H , a metric $q : H \rightarrow V$, and a Boolean property $g : H \rightarrow \{0, 1\}$, there is an exact metric-only classifier $\bar{g} : q(H) \rightarrow \{0, 1\}$ with $g = \bar{g}q$ if and only if

$$q(h) = q(h') \implies g(h) = g(h').$$

Proof. Factorization implies equality on each fiber. Conversely, define $\bar{g}(v)$ to be the common value of g on the nonempty fiber over v . This value is unique, so its definition requires no choice of a representative. $\square$

Categorical form. Let $q : H \twoheadrightarrow V$ be an effective epimorphism between 0-truncated objects of an infinity-topos, and $\chi : H \rightarrow \Omega$ classify a subobject. There is a unique $\bar{\chi} : V \rightarrow \Omega$ with $\chi = \bar{\chi}q$ precisely when

$$\chi\pi_1 = \chi\pi_2 \quad \text{on } H \times_V H.$$

Proof. Necessity follows by composition. Effective descent identifies maps from V with compatible maps from the Čech nerve of q . Since Ω is 0-truncated, the kernel-pair equality supplies the compatibility, with no further nontrivial higher coherence. Descent gives existence and uniqueness. For general q , apply this to its image. $\square$

Consequently, if two histories have equal views but different welfare, legality, or preservation outcomes, views alone cannot give an exact certificate of that property. A sound but incomplete classifier can reject a mixed fiber entirely. The theorem concerns information sufficiency, not whether a reward increases a metric. Even a perfectly implemented counter and reward service can optimize a metric that fails to distinguish the desired histories.

8. Authority and robust behavioral capability

Let Θ_A be actor A 's deployable configurations, $B(\theta) \neq \emptyset$ allowed responses, and $Z(\theta, b) \neq \emptyset$ allowed disturbances, specified independently of the desired conclusion. With P fixed, define robustly enforceable properties

$$\begin{aligned} \mathcal{O}(\theta) &= \{\text{hist}(P, \theta, b, z) : b \in B(\theta), z \in Z(\theta, b)\}, \\ \mathcal{G}_A &= \{D \subseteq H : \exists \theta \in \Theta_A, \mathcal{O}(\theta) \subseteq D\}. \end{aligned}$$

Proposition 8.1 (authority monotonicity and limits). If $\Theta_A \subseteq \Theta_{A'}$ and all other data are unchanged, then $\mathcal{G}_A \subseteq \mathcal{G}_{A'}$. Moreover, $D \in \mathcal{G}_A$ precisely when some allowed configuration's entire outcome set is contained in D . Existence of a desirable response alone is insufficient.

Proof. The same witness θ remains available to A' . The second assertion expands the universal quantifiers. If every configuration has an allowed counter-response outside D , none witnesses membership. $\square$

Operators, designers, advertisers, participants, and regulators may control different parameters. Server ownership need not confer authority over every reward; writing code need not authorize deployment. Testing social influence requires actual authority, response behavior, and measured outcomes. Stochastic responses require an explicit probability or expectation criterion.

9. Evidence, limitations, and reproducibility

The accompanying standard-library Python artifact exhaustively explores a finite reservation scheduler, checks invariants, and analyzes fair cycles for response progress. It contrasts weak fairness of component ticks with fairness of actual service actions; checks replica merge counterexamples and metric fibers; and computes the incentive threshold and replicator trajectories. Synthetic parameters are exactly those in Section 6. No user telemetry, current platform internals, or causal estimates of human behavior are used.

Finite exploration verifies only the encoded finite system; it does not verify production services or replace imported impossibility proofs. No proof-assistant verification is claimed. Source links were checked on 16 October 2026; access evidence and results accompany the sources.

Admissible executable protocols delimit guaranteed services; authorized configurations and participant responses delimit robust behavioral influence. Desirable outcomes require additional properties that metrics may fail to classify.

Authorship and automation. Anonymous is the responsible human author, who supplied the thesis and authorized submission. OpenAI Codex assisted with formalization, references, drafting, artifacts, and document checks. The work remains subject to editorial and independent mathematical review.

References

- [1] B. Alpern and F. B. Schneider. *Defining liveness*. IPL 21(4):181-185, 1985. [Report](#).
- [2] M. Abadi and L. Lamport. *The existence of refinement mappings*. TCS 82(2):253-284, 1991. [Report](#).
- [3] M. J. Fischer, N. A. Lynch, and M. S. Paterson. *Impossibility of distributed consensus with one faulty process*. JACM 32(2):374-382, 1985. [Paper](#).
- [4] C. Dwork, N. Lynch, and L. Stockmeyer. *Consensus in the presence of partial synchrony*. JACM 35(2):288-323, 1988. [Paper](#).
- [5] M. Herlihy and N. Shavit. *The topological structure of asynchronous computability*. JACM 46(6):858-923, 1999. [Paper](#).
- [6] P. Bailis, A. Fekete, M. J. Franklin, A. Ghodsi, J. M. Hellerstein, and I. Stoica. *Coordination avoidance in database systems*. PVLDB 8(3):185-196, 2014. [Paper](#).
- [7] J. Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170, Princeton, 2009. [Author version](#), Sections 5.5.6, 6.1, 6.3: truncation, descent, and topoi.
- [8] P. Covington, J. Adams, and E. Sargin. *Deep neural networks for YouTube recommendations*. RecSys 2016, 191-198. [Google Research](#).
- [9] J. Perdomo, T. Zrnic, C. Mendler-Dunner, and M. Hardt. *Performative prediction*. PMLR 119:7599-7609, 2020. [Paper](#).
- [10] N. Ghani, J. Hedges, V. Winschel, and P. Zahn. *Compositional game theory*. LICS 2018:472-481. [Preprint](#). Our elementary response-set model does not implement open games.