

An ∞ -Topos Semantics for a Communist Collective of Computing Agents

Common ownership, resource-sensitive planning,
and higher coherence

October 13, 2026 · Preprint v1.0

Abstract

We give a mathematical specification of a collective of computing agents whose productive assets are held in common, whose members have equal constitutional standing, and whose resources are allocated by admitted need rather than by ownership claims. The collective is a structured object in an ∞ -topos of sheaves of spaces; irreversible executions form a separate internal $(\infty, 1)$ -category. Dependent types express ownership, resource feasibility, authorization, and preservation obligations. We prove an invariant-preservation theorem, a finite quorum-intersection theorem, an exact rational allocation theorem, and a descent theorem for coherently compatible local states. We also prove that deterministic anonymous allocation of one indivisible resource is impossible at a fully symmetric state, and express this obstruction as the absence of a section of an associated bundle over a classifying space. Univalence identifies equivalent structured presentations, while homotopy quotients retain their automorphisms. A four-agent example and an exact-arithmetic verification supplement make the finite fragment explicit. The construction supplies a semantics and conditional correctness results; political axioms, authenticated membership, resource measurements, and protocol progress remain separately stated assumptions.

Keywords: ∞ -topos; homotopy type theory; collective ownership; distributed governance; resource allocation; homotopy quotient.

1 Purpose, scope, and contribution

A collection of networked computers becomes a *collective* only after rules have been specified for membership, productive assets, access to results, and the authority to allocate resources. A graph records connectivity but does not by itself specify these institutions. A higher categorical semantics can express their composition and coherence, provided the institutional requirements are supplied as explicit mathematical data.

We use *communist collective* for the particular constitutional model in Definition 3.3: productive assets have a common owner, membership carries equal voting standing, and allocation is determined by admitted needs and available common capacity. This is an operational definition for computing agents, not an exhaustive definition of historical or political communism. Its distinction between productive contribution and distribution by need is motivated by Marx’s discussion in Part I of the *Critique of the Gotha Programme* [6]. An agent’s “need” here is an admitted resource requirement for a service or task; no claim about machine consciousness is required.

The paper has three levels. The *ambient semantics* is an ∞ -topos. The *constitution* is a record of types, operations, and propositions internal to that semantics. The *execution model* is a directed higher category whose permitted operations satisfy the constitution. None of these levels can be substituted for another: common ownership is not a colimit, a path is not a CPU instruction, and a sheaf condition is not a consensus algorithm.

The principal results are as follows.

1. A constitutional invariant is preserved by every finite certified execution, and transfers to an implementation through a step-preserving interpretation (Theorem 4.1).
2. A finite quorum condition rules out conflicting certificates under an explicit non-equivocation hypothesis (Theorem 5.2).
3. A closed rational formula maximizes the guaranteed fraction of admitted demand in a divisible-resource model (Theorem 6.3).
4. Coherent local certified states admit a global state with a contractible space of compatible gluings (Theorem 7.1). Mere local existence does not supply those coherences.
5. A symmetric indivisible allocation problem admits no deterministic anonymous selector; randomized allocation has a different codomain and avoids this particular obstruction (Theorem 8.1).

The background mathematics is established. Higher topos theory supplies descent and effective groupoid quotients [5]; homotopy type theory supplies dependent and identity types [10]; Shulman’s semantics supplies strict univalent universes [9]. Complete Segal objects provide the directed categorical layer [7]. Riehl–Shulman type theory is a relevant syntactic framework for directed higher categories [8]. Resource theories [1] and multi-resource fairness [3] provide related computational perspectives. The contribution is the specified combination of these structures and the explicit implications and obstructions for this constitutional model. The elementary allocation and quorum arguments are not claimed as new general optimization or consensus theorems.

2 Ambient semantics and the meaning of a witness

2.1 The site and its ambient topos

Fix a small Grothendieck site $(\mathcal{C}, J)$. An object U of $\mathcal{C}$ represents an observational context, such as a local view of a system and its available evidence. A morphism $V \rightarrow U$ specifies a restriction of view. The topology J records which families of views are declared to constitute a cover. Its choice is part of the model: a network link alone does not establish a covering family.

Let $\mathcal{S}$ denote the ∞ -category of spaces and set

$$\mathcal{X} = \mathrm{Shv}_{\mathcal{S}}(\mathcal{C}, J), \quad L_J : \mathrm{Fun}(\mathcal{C}^{\mathrm{op}}, \mathcal{S}) \rightleftarrows \mathcal{X} : i. \quad (2.1)$$

Here L_J is the accessible left exact sheafification functor and i is the fully faithful inclusion. Thus $\mathcal{X}$ is an ∞ -topos. We use ordinary sheaf descent, not an additional assumption of hypercompleteness. For the trivial topology this is a presheaf ∞ -topos; for the one-object terminal site it is $\mathcal{S}$.

An object $F \in \mathcal{X}$ gives a space $F(U)$ of admissible observations in each context. Points of $F(U)$ are presentations, paths compare equivalent presentations, and higher paths express coherences between those comparisons. These interpretations must be specified for each object; no physical equivalence is inferred merely by placing two descriptions in the same type.

We work with a hierarchy of sufficiently large universes and consider only universe-bounded data. The ∞ -topos can be presented so as to interpret dependent sums, dependent products, identity types, and strict univalent universes [9]. We use $\mathbf{1}$ for the terminal object, $\mathbf{0}$ for the empty type, $\mathbf{Prop}$ for mere propositions, and $\mathbf{Set}$ for 0-truncated types. A mere proposition has at most one point up to a contractible choice; a set has proposition-valued identity types.

2.2 Internal logic

In the internal language, a family $B : A \rightarrow \mathcal{U}$ is represented by a dependent object over A . The type $\sum_{a:A} B(a)$ carries a value and its witness, while $\prod_{a:A} B(a)$ is a dependent function. The identity type $\mathrm{Id}_A(a, b)$ records identification with its possible higher structure. The propositional truncation $\|B\|_{-1}$ retains only the assertion of inhabitation.

For a type of plans $\mathrm{Plan}(s)$ at state s , the expressions

$$\prod_{s:S} \|\mathrm{Plan}(s)\|_{-1} \quad \text{and} \quad \prod_{s:S} \mathrm{Plan}(s) \quad (2.2)$$

have different meanings. The first does not in general produce the second. A supplied term of the second gives a uniform semantic choice; an executable scheduler additionally requires an effective presentation of that term and its inputs. Section 6 provides such an explicit term for a finite rational fragment.

A witness in context U is a generalized point $U \rightarrow F$, not necessarily a global point $\mathbf{1} \rightarrow F$. The construction does not assume that $\mathcal{X}$ has enough geometric points. Section 7 makes precise when local witnesses glue.

3 The constitutional object

Fix integers $m \geq 3$, $d \geq 1$, and $r \geq 1$. Let

$$A = \mathrm{Fin}(m), \quad R = \mathrm{Fin}(r), \quad B = (\mathbb{Q}_{\geq 0})^d,$$

interpreted internally. The type A records admitted members and R records productive assets. Coordinates of B have independently fixed units, for example core-hours and GiB-hours. Inequalities on B are componentwise; quantities with different units are never added to each other.

Membership is fixed within an epoch. A member is a constitutional identity, not an arbitrary process identifier: replicas of one member do not obtain additional votes. Admission and removal are separate certified transitions between epochs. The formal results below assume the advertised membership and epoch are authenticated.

3.1 State data and common ownership

Let $S : \mathcal{U}$ be a type of raw states. It carries the following fields, all natural under restriction of context:

$$\begin{aligned} o : S &\rightarrow (R \rightarrow \mathbf{1} + A), & \kappa, c, n, x : S &\rightarrow (A \rightarrow B), \\ b : S &\rightarrow B, & \mathrm{Read} : S &\rightarrow A \rightarrow K_{\mathrm{com}} \rightarrow \mathbf{Prop}. \end{aligned}$$

The fields are, respectively, productive ownership, contribution capacities, actual contributions, admitted needs, allocations, the available common budget, and access rights to a designated common

knowledge type K_{com} . Further fields may contain code, task specifications, measurements, and private local state.

Definition 3.1 (Common productive ownership). The ownership proposition at $s : S$ is

$$\text{Own}(s) := \prod_{r:R} (o_s(r) = \text{inl}(\star)). \quad (3.1)$$

The left summand names the collective. The alternative $\text{inr}(a)$ names an individual owner. A use permit for a member is a separate field and does not change o_s .

This is a statement about the modeled title and control policy. It does not prove that an external cloud provider, human operator, or legal system implements that policy. Other owner classes can be added to the principal type without changing the definition of common ownership.

Definition 3.2 (Basic constitutional invariant). Define the propositions

$$\text{Ability}(s) := \prod_{a:A} (0 \leq c_s(a) \leq \kappa_s(a)), \quad (3.2)$$

$$\text{Feas}(s) := \left(\prod_{a:A} 0 \leq x_s(a) \leq n_s(a) \right) \times \left(\sum_{a:A} x_s(a) \leq b_s \right), \quad (3.3)$$

$$\text{CommonKnowledge}(s) := \prod_{a:A} \prod_{k:K_{\text{com}}} \text{Read}(s, a, k), \quad (3.4)$$

$$\text{Inv}(s) := \text{Own}(s) \times \text{Ability}(s) \times \text{Feas}(s) \times \text{CommonKnowledge}(s). \quad (3.5)$$

The certified state type is $S_{\text{Inv}} := \sum_{s:S} \text{Inv}(s)$.

Every factor of $\text{Inv}(s)$ is a mere proposition, so $S_{\text{Inv}} \rightarrow S$ is a subobject. Common knowledge means a common access entitlement, not uninterrupted network delivery. The specified K_{com} need not include private inputs, credentials, or every member's internal state.

The ability inequality bounds demands placed on a contributor. It does not require maximizing every machine's utilization. Available capacity is separately measured. A simple epoch model is

$$b = b^{\text{ext}} + \sum_{a:A} c(a) - h, \quad 0 \leq h \leq b^{\text{ext}} + \sum_{a:A} c(a), \quad (3.6)$$

where h is the maintenance and reserve provision and b^{ext} is an admitted external supply. Other production functions may replace (3.6).

3.2 Equal standing and collective authorization

Let $\text{Proposal}(s)$ be a type of proposed control-plane operations at s , with a target $t_s(p) : S$. Each proposal includes an epoch identifier and an unambiguous content identifier. Endorsement and certificate types are defined over the total proposal type $\sum_{s:S} \text{Proposal}(s)$; the source state is suppressed in the notation when fixed. Fix a threshold q with

$$2 \leq q \leq m - 1. \quad (3.7)$$

Let $\text{Endorse}(e, a, p)$ be a type of authenticated endorsements by member a of proposal p in epoch e . It is evidence-bearing data, not merely a truth value. The type of certificates is

$$\text{Cert}_e(p) := \sum_{V \subseteq A} (q \leq |V|) \times \prod_{a \in V} \text{Endorse}(e, a, p), \quad (3.8)$$

where V ranges over decidable subsets, so repeated signatures from one member count once.

Every member has the same eligibility and contributes one unit to $|V|$. Capacities, ownership claims, and past contributions carry no additional voting weight. Equation (3.7) excludes a singleton certificate and permits a certificate omitting any specified member, if the other required endorsements are supplied. It does not imply that endorsements will be forthcoming.

Anonymity requires more than the absence of a field called “wealth.” Let $G = \text{Sym}(A)$ act on all member-indexed data by reindexing, including endorsements and proposals. The authorization and allocation rules are required to be G -equivariant. A mere renaming then cannot change a constitutional decision. Different capabilities, needs, and authenticated histories may still justify different operational treatment.

Precisely, a needs-oriented allocation rule is a G -equivariant map

$$\bar{\alpha} : B \times B^A \longrightarrow B^A \quad (3.9)$$

whose output is feasible for its input (b, n) and equals n whenever $\sum_a n_a \leq b$. Its state-dependent implementation factors as $\alpha = \bar{\alpha} \circ \langle b, n \rangle$. This excludes a separate personal-title or stake argument. It leaves the scarcity rule to be specified; the radial instance fixes that rule explicitly.

Definition 3.3 (A communist computing collective). A *communist computing collective* in $\mathcal{X}$ consists of the state and resource data above, an initial certified state, an anonymous equal-standing authorization rule (3.8), a needs-oriented allocation rule, and a directed execution structure satisfying the preservation obligations in Section 4. Productive title remains common and the designated common knowledge remains accessible by right. Allocation factors through admitted needs and common budgets, rather than individual titles or voting stakes. The *radial collective* is the instance using the allocation rule of Theorem 6.3.

The definition admits functional specialization and temporary coordination roles. It excludes an ownership class *within the represented constitution*; it does not assert equal capabilities, identical data, absence of organizational roles, or absence of power outside the model. A change that repeals these axioms is a transition to a different constitutional class, not a theorem that such change is physically impossible.

4 Directed computation and invariant preservation

4.1 Execution is not identity

An identity $s =_S t$ is invertible. A computation that consumes energy or irreversibly updates a record need not be. We therefore represent execution by an internal complete Segal object

$$\mathcal{D}_\bullet : \Delta^{\text{op}} \longrightarrow \mathcal{X}. \quad (4.1)$$

Here $\mathcal{D}_0 = S$ is the state object, $\mathcal{D}_1$ contains directed executions, and for $n \geq 2$ the Segal maps

$$\mathcal{D}_n \simeq \underbrace{\mathcal{D}_1 \times_S \cdots \times_S \mathcal{D}_1}_{n \text{ factors}} \quad (4.2)$$

encode coherent composition. Completeness says that the identity degeneracy identifies S with the subobject of equivalences in $\mathcal{D}_1$. Higher morphisms compare executions by specified equivalences;

all morphisms above dimension one are invertible in this $(\infty, 1)$ -categorical model. Noninvertible refinements between executions would require a richer, for example $(\infty, 2)$ -categorical, model.

This is a semantic definition of an internal higher category, not a claim that the entire simplicial coherence structure has already been encoded in ordinary book HoTT. A directed or two-level formalization supplies additional syntactic infrastructure; the approach of Riehl and Shulman [8] is one option. The finite data and proofs below use ordinary dependent-type constructions.

4.2 Certified primitive steps

Choose primitive steps $\text{Step}(s, t)$ whose finite composites present the executions of interest. The specification includes maps $\text{Step}(s, t) \rightarrow \mathcal{D}(s, t)$ and a coherent interpretation of concatenated traces, where $\mathcal{D}(s, t)$ is the fiber of $\mathcal{D}_1 \rightarrow S \times S$ at (s, t) . Coverage of an implementation requires the separate premise in Theorem 4.1. A control-plane step may implement a batch of computations under one approved plan. Each primitive step carries a proposal, the applicable certificate, an execution witness, and a preservation term. In dependent-type notation its required interface includes

$$\text{preserve} : \prod_{s, t : S} \text{Step}(s, t) \rightarrow \text{Inv}(s) \rightarrow \text{Inv}(t). \quad (4.3)$$

For example, with $\text{Run}(s, p)$ the type of runtime evidence and $e(s, p)$ the bound epoch, one may use the constructor type

$$\begin{aligned} \text{Step}(s, t) := \sum_{p : \text{Proposal}(s)} & \text{Cert}_{e(s, p)}(p) \times (t_s(p) = t) \\ & \times \text{Run}(s, p) \times (\text{Inv}(s) \rightarrow \text{Inv}(t)). \end{aligned} \quad (4.4)$$

For example, a plan-update constructor supplies a certified allocation for its target, preserves the ownership field, and preserves the common-read policy. More generally, a verifier must check these obligations for each accepted constructor. Type declarations alone are not proofs of arbitrary machine code.

Let $\text{Trace}_0(s, t) := (s = t)$ and recursively define

$$\text{Trace}_{n+1}(s, t) := \sum_{u : S} \text{Trace}_n(s, u) \times \text{Step}(u, t). \quad (4.5)$$

The primitive trace type keeps the certificate data even when a semantic interpretation later identifies equivalent executions. If an internal localization identifies traces, only equivalences compatible with the endpoints and the constitutional predicates may be used.

Theorem 4.1 (Preservation and implementation transfer). *Suppose $s_0 : S$ carries $i_0 : \text{Inv}(s_0)$ and the primitive steps carry (4.3). Then*

$$\prod_{n : \mathbb{N}} \prod_{t : S} \text{Trace}_n(s_0, t) \longrightarrow \text{Inv}(t).$$

Moreover, suppose an implementation has states O , a map $\rho : O \rightarrow S$, and an initial state o_0 with $\text{Inv}(\rho(o_0))$. If every implementation step $o \rightsquigarrow o'$ supplies a trace from $\rho(o)$ to $\rho(o')$, every finitely reachable implementation state satisfies $\text{Inv} \circ \rho$.

Proof. For length zero, transport i_0 along the given identity. For length $n + 1$, a trace consists of u , a length- n trace to u , and a primitive step $u \rightarrow t$. Apply the induction hypothesis to obtain $\text{Inv}(u)$, then (4.3) to obtain $\text{Inv}(t)$. For the implementation claim, concatenate the supplied finite traces and apply the first part, or induct directly on the number of implementation steps. $\square$

Corollary 4.2 (Non-appropriation in certified histories). *For any finitely reachable certified state t , asset $r : R$, and member $a : A$, the type $o_t(r) = \text{inr}(a)$ is empty.*

Proof. Preservation supplies $o_t(r) = \text{inl}(\star)$. A proposed identification with $\text{inr}(a)$ would identify the disjoint summands of $\mathbf{1} + A$, which is impossible. $\square$

For a complete Segal execution object, the full subobject on certified states has levels

$$\mathcal{D}_n^{\text{Inv}} = \mathcal{D}_n \times_{S^{n+1}} (S_{\text{Inv}})^{n+1}. \quad (4.6)$$

It is again a complete Segal object: the Segal equivalences restrict by pullback, and completeness restricts because Inv is invariant under identities. This construction selects endpoints satisfying the invariant; it does not replace the separate certificate requirement on arrows. In particular, an execution structure containing unauthorized arrows is not repaired simply by retaining safe endpoints.

Theorem 4.1 is a structural safety result. Its implementation premise fails if an unmodeled administrator can rewrite assets, if measurements are false, or if the runtime bypasses verification. Crashes, attacks, and resource loss must be included among implementation transitions whenever the theorem is applied to them.

5 Quorums, conflicting plans, and progress

Fix an epoch and let at most f of the m members be faulty. A binary relation $p \# p'$ declares proposals mutually incompatible within the same decision slot.

Assumption 5.1 (Authenticated non-equivocation). Certificates contain endorsements by the claimed distinct members, and an honest member never endorses both p and p' when $p \# p'$ in the same epoch and slot. Epoch and proposal identities are bound into every endorsement.

Theorem 5.2 (Quorum safety). *If $2q > m + f$, incompatible proposals cannot both carry valid certificates under authenticated non-equivocation. If, in addition, a certificate must be obtainable without any faulty member's cooperation, the structural requirement is $q \leq m - f$. An integer threshold satisfying both inequalities exists exactly when $m > 3f$.*

Proof. Let V, W be two signer sets of size at least q . Then

$$|V \cap W| = |V| + |W| - |V \cup W| \geq 2q - m > f.$$

Their intersection contains an honest member, contradicting non-equivocation for incompatible proposals. To obtain a certificate from honest members alone, their count $m - f$ must be at least q ; if they all endorse one proposal and their endorsements are delivered, that inequality is sufficient to assemble its certificate. Combining $2q > m + f$ and $q \leq m - f$ gives $m > 3f$. Conversely, if $m > 3f$, take $q = m - f$; then $2q > m + f$. $\square$

The last equivalence concerns the two displayed numerical inequalities. The separate equal-standing requirement (3.7) also excludes degenerate small collectives. The familiar choice $m = 3f + 1$, $q = 2f + 1$ satisfies all requirements for $f \geq 1$.

There is no liveness theorem here. Non-equivocation can prevent progress when honest members endorse different proposals; obtaining and maintaining the required locking discipline is a protocol

problem. An amendment or retry must specify its slot and locking rules, not simply reuse this static counting argument. The Fischer–Lynch–Paterson theorem rules out guaranteed termination of deterministic consensus in a fully asynchronous message-passing system with even one possible crash, under its stated consensus conditions [2]. An ∞ -topos semantics does not change that result. The topological study of distributed computability likewise supplies genuine task-dependent obstructions, rather than a universal gluing algorithm [4].

6 Resource sensitivity and allocation by admitted need

6.1 Information can be copied without copying its capacity

The cartesian diagonal $B \rightarrow B \times B$ copies a *description* of a resource quantity. It must not be interpreted as manufacturing that resource. We therefore retain a separate symmetric monoidal resource category, following the distinction between processes and resources developed in [1].

Definition 6.1 (The budget category). The category Res_d has objects $v \in (\mathbb{Q}_{\geq 0})^d$ and a unique morphism $v \rightarrow w$ exactly when $w \leq v$. Its tensor is addition, $v \otimes w = v + w$, with unit 0. A morphism permits discarding unused capacity; it does not permit creating additional capacity.

Lemma 6.2 (No free duplication of a positive budget). *A morphism $v \rightarrow v \otimes v$ exists in Res_d if and only if $v = 0$.*

Proof. Such a morphism is the inequality $2v \leq v$. In each nonnegative rational coordinate this is equivalent to $v_k = 0$. $\square$

An allocation $x : A \rightarrow B$ is resource-feasible precisely when there is a resource morphism $b \rightarrow \sum_a x(a)$, together with the individual demand bounds. If computations genuinely produce new usable capacity, that production must appear in an extended process model or the next measured budget. The budget category is a small accounting model, not a semantics of all physical production.

6.2 A constructive allocation rule

Fix a budget $b \in B$ and admitted need vectors $n_a \in B$. The model in this subsection assumes divisible resources, a common planning interval, and no additional placement, deadline, or integer constraints. A service vector is $u \in [0, 1]_{\mathbb{Q}}^A$, with allocation

$$x_{ak} = u_a n_{ak}. \quad (6.1)$$

Thus u_a is the fraction of member a 's admitted bundle that is supplied. Define

$$A^+ = \{a \mid n_a \neq 0\}, \quad N_k = \sum_{a:A} n_{ak}, \quad D^+ = \{k \mid N_k > 0\}. \quad (6.2)$$

For an inactive member put $u_a = 1$, meaning that its zero requirement is fully met. A service vector is feasible if $\sum_a u_a n_{ak} \leq b_k$ for each coordinate. Its guaranteed service is

$$\sigma(u) = \begin{cases} \min_{a \in A^+} u_a, & A^+ \neq \emptyset, \\ 1, & A^+ = \emptyset. \end{cases} \quad (6.3)$$

Theorem 6.3 (Exact radial allocation). *Set*

$$\lambda^*(b, n) = \min \left(\{1\} \cup \left\{ \frac{b_k}{N_k} : k \in D^+ \right\} \right), \quad u_a^* = \begin{cases} \lambda^*, & a \in A^+, \\ 1, & a \notin A^+. \end{cases} \quad (6.4)$$

Then $x_{ak}^* = u_a^* n_{ak} = \lambda^* n_{ak}$ is feasible and

$$\sigma(u^*) = \max\{\sigma(u) : u \text{ is feasible}\} = \lambda^*. \quad (6.5)$$

The formula is a total finite rational algorithm and is equivariant under every permutation of members. All admitted needs can be met simultaneously if and only if $N_k \leq b_k$ for every k .

Proof. All entries of the finite set in (6.4) are nonnegative and it contains 1, so $0 \leq \lambda^* \leq 1$. If $N_k > 0$, then $\lambda^* \leq b_k/N_k$, hence

$$\sum_a x_{ak}^* = \lambda^* N_k \leq b_k.$$

If $N_k = 0$, nonnegativity forces every $n_{ak} = 0$, so the sum is zero. Individual bounds $0 \leq x_{ak}^* \leq n_{ak}$ follow from $0 \leq \lambda^* \leq 1$.

Let u be any feasible service vector and put $\mu = \sigma(u)$. Every active member satisfies $u_a \geq \mu$, and inactive members contribute no demand. For every $k \in D^+$,

$$\mu N_k \leq \sum_a u_a n_{ak} \leq b_k.$$

Therefore $\mu \leq b_k/N_k$ for every active coordinate, and $\mu \leq 1$, giving $\mu \leq \lambda^*$. If there are active members, all of their u_a equal λ^* ; if there are none, both sides equal 1. This proves optimality.

The construction uses finite sums, decidable zero tests, division by explicitly positive rationals, and a nonempty finite minimum. It is therefore total and exact. A permutation of members preserves each N_k , hence λ^* , and reindexes x^* ; this proves equivariance. Finally, if $N \leq b$, choosing $u_a = 1$ meets every need. Conversely, any allocation meeting every need has total demand N and must satisfy $N \leq b$. $\square$

Corollary 6.4 (Independence from individual contribution claims). *Two states with the same admitted need function and the same available budget receive the same radial allocation, even if their individual contribution histories differ. If $n_a = n_{a'}$, then $x_a^* = x_{a'}^*$, regardless of their capacities. If $\lambda^* > 0$ and $n_a \neq 0$, then $x_a^* \neq 0$, including when $c_a = 0$.*

Proof. The allocation formula depends only on (b, n) . Equal need vectors are multiplied by the same scalar; a nonzero nonnegative need vector has a positive coordinate, which remains positive under multiplication by $\lambda^* > 0$. $\square$

Contribution can affect the total budget through (3.6); the corollary excludes *individual entitlement weights* beyond that effect. An arbitrary policy could hide ownership priorities inside its admitted-needs function. Consequently, the method for admitting n is part of the constitution and must be audited separately. Truthful revelation of private demand, social welfare maximization, and incentive compatibility are not consequences of the formula.

The rule chooses one max–min optimizer. It need not be the unique optimizer or a lexicographically max–min allocation; some coordinates can have unused capacity. If work has placement or indivisibility constraints, feasibility must be strengthened and this formula may cease to apply. Dominant Resource Fairness [3] uses a different normalization and objective; its properties cannot be transferred to the present rule without proof.

6.3 The dependent-type interface

Let

$$\text{Allocation}(b, n) := \sum_{x: A \rightarrow B} \left(\prod_a x(a) \leq n(a) \right) \times \left(\sum_a x(a) \leq b \right), \quad (6.6)$$

$$\text{Service}(b, n) := \sum_{u: A \rightarrow [0,1]_{\mathbb{Q}}} \left(\prod_{a \notin A^+} u_a = 1 \right) \times \left(\sum_a u_a n_a \leq b \right). \quad (6.7)$$

The nonnegativity of x is built into B . Theorem 6.3 supplies actual terms

$$\text{allocate} : \prod_{b: B} \prod_{n: A \rightarrow B} \text{Allocation}(b, n), \quad (6.8)$$

and a corresponding service term together with a proof that every feasible service vector has guarantee at most λ^* . This is an untruncated construction. No choice principle or general optimization oracle is invoked. Because it is defined by finite rational operations, it can be interpreted internally and respects context restriction. An arbitrary semantic section, by contrast, need not arrive with executable code.

For the radial collective, one may strengthen (3.5) to

$$\text{Inv}_{\text{rad}}(s) := \text{Inv}(s) \times (x_s = \lambda^*(b_s, n_s)n_s).$$

The preservation theorem applies to this stronger predicate whenever each primitive step supplies the corresponding preservation term.

7 Descent and the obstruction to uncoordinated gluing

A morphism $e : U \rightarrow V$ in $\mathcal{X}$ is an *effective epimorphism* when its augmented Čech nerve realizes to V . Write

$$U^{[n]} = \underbrace{U \times_V \cdots \times_V U}_{n+1 \text{ factors}}.$$

A covering family in the site induces such an effective epimorphism after passing to sheaves. The nerve records overlaps, triple overlaps, and their higher compatibility conditions.

Theorem 7.1 (Descent of certified states). *For any effective epimorphism $U \rightarrow V$ and certified state object S_{Inv} ,*

$$\text{Map}_{\mathcal{X}}(V, S_{\text{Inv}}) \simeq \lim_{[n] \in \Delta} \text{Map}_{\mathcal{X}}(U^{[n]}, S_{\text{Inv}}). \quad (7.1)$$

For a specified point of the right-hand limit, the space of global states together with an identification of their restriction with that point is contractible.

Proof. Effectiveness gives $V \simeq \text{colim}_{\Delta^{\text{op}}} U^{[\bullet]}$. The representable contravariant functor $\text{Map}_{\mathcal{X}}(-, S_{\text{Inv}})$ carries this colimit to the limit in (7.1). A homotopy fiber of an equivalence over a specified point is contractible. $\square$

This is an application of ∞ -topos descent [5], not a new consistency algorithm. For 0-truncated states, the data reduce to local states agreeing on overlaps. For general states they include

identifications on double overlaps, homotopies on triple overlaps, and the remaining coherences. Pairwise agreement up to unspecified homotopy is insufficient.

The theorem applies only when all relevant requirements, including global resource accounting, are represented in S_{Inv} and in its restriction maps. Two nodes that independently promise the same exclusive GPU do not obtain a valid combined reservation merely because each promise is locally plausible. If the overlap data do not record that exclusivity, the model has omitted the constraint. If they do, the attempted local states may fail to define a point of the right-hand limit.

Example 7.2 (Local witnesses without a global choice). Take $\mathcal{X} = \text{Shv}_{\mathcal{S}}(S^1)$ and let T be the sheaf of local sections of the double covering $p : S^1 \rightarrow S^1$, $p(z) = z^2$. Every sufficiently small arc admits a section, so $T \rightarrow \mathbf{1}$ is an effective epimorphism and $\|T\|_{-1} \simeq \mathbf{1}$. A global section would be a continuous map $g : S^1 \rightarrow S^1$ with $p \circ g = \text{id}$. Taking degrees gives $2 \deg(g) = 1$, a contradiction. Thus T is locally inhabited but has no global section. The missing object is a coherent compatible selection, not a proof that local sections exist.

The example distinguishes sheaf gluing from an unrestricted choice principle. It also explains why local certificates of feasibility cannot silently be promoted to a globally implemented plan.

8 Anonymous choice and indivisible resources

We now isolate a second obstruction, using finite sets in $\mathcal{S}$. Let a group G act transitively on a finite member set A with $|A| \geq 2$. Let Z be a set of inputs with a G -action. A deterministic selector for one indivisible resource is a function $f : Z \rightarrow A$; it is *anonymous* when $f(gz) = gf(z)$.

Theorem 8.1 (No deterministic anonymous selector at a symmetric input). *If $z_0 \in Z$ is fixed by every $g \in G$, no anonymous selector $f : Z \rightarrow A$ exists.*

Proof. Equivariance would give $f(z_0) = f(gz_0) = gf(z_0)$ for every g . Thus $f(z_0)$ would be a fixed point of the action on A . A transitive action on a set with at least two elements has no fixed point: choose another element and an element of G carrying $f(z_0)$ to it. $\square$

Corollary 8.2 (The associated bundle does not split). *For the preceding finite transitive action, the bundle*

$$A // G = EG \times_G A \longrightarrow BG \tag{8.1}$$

is an effective epimorphism of spaces with nonempty fiber A , but has no section.

Proof. The base BG is connected and the fiber is nonempty, so the map is surjective on connected components, hence an effective epimorphism in spaces. Its section space is the homotopy fixed-point space $\text{Map}_G(EG, A)$. Since EG is connected and A is discrete, any such map is constant. Equivariance would require its constant value to be a G -fixed point, which does not exist. $\square$

This obstruction is compatible with equal membership. An execution can choose a coordinator after receiving nonsymmetric ballots, a verifiable random seed, or an agreed rotation history. Those data change the input; the theorem concerns a deterministic equivariant rule on an unchanged fully symmetric input. Ordered member identifiers also break symmetry if the rule privileges their order.

There is a canonical invariant element in a different output type, the finite rational probability simplex

$$\text{Prob}_{\mathbb{Q}}(A) = \left\{ p : A \rightarrow \mathbb{Q}_{\geq 0} \mid \sum_a p(a) = 1 \right\} : \quad p(a) = \frac{1}{|A|}.$$

Uniform randomization therefore gives a symmetric *distribution* of assignments. Each realized assignment still names one recipient; it is not an invariant deterministic assignment. Repeated service guarantees require assumptions about the random source, scheduling, and time horizon. The divisible-resource formula of Section 6 avoids this obstruction because its codomain permits equal fractional allocations.

9 Univalence, structured equivalence, and collective moduli

9.1 Equality of representations

Univalence asserts that the canonical map $(X = Y) \rightarrow \text{Equiv}(X, Y)$ is an equivalence for small types. It does not make all agents identical or turn any two governance rules into the same rule. A representation must carry its structure across an equivalence.

Proposition 9.1 (Structured identity). *Let $F : \mathcal{U} \rightarrow \mathcal{U}'$ be a family of structures on small types. For $(X, s), (Y, t) : \sum_{T:\mathcal{U}} F(T)$, univalence gives*

$$((X, s) = (Y, t)) \simeq \sum_{e:\text{Equiv}(X, Y)} (\text{transport}_F(\text{ua}(e), s) = t). \quad (9.1)$$

Proof. The identity type of a dependent sum is equivalent to

$$\sum_{p:X=Y} (\text{transport}_F(p, s) = t).$$

Replace $X = Y$ with $\text{Equiv}(X, Y)$ using univalence and its inverse ua . Dependent sums preserve equivalences of the indexing type with the corresponding transported fibers. $\square$

For a collective, F includes the ownership, resource, governance, and execution data, with their laws; multiple carrier types can be handled by the corresponding iterated dependent record. The transport condition is essential. An equivalence of underlying storage formats that does not preserve certificates and budgets is not an equivalence of collective structures. See [10] for identity types and univalence; the proposition is a direct use of these standard rules.

9.2 Forgetting labels while retaining automorphisms

Fix a universe-bounded, explicitly labelled presentation of collective data and let $M \in \mathcal{X}$ be its object of models. In particular, agent labels are fixed parameters rather than already identified by univalence. Suppose $G = \text{Sym}(A)$ acts coherently on M by reindexing all member-dependent fields and laws. Define

$$\text{CollectiveModuli} := M \parallel G := |[n] \mapsto M \times G^n|. \quad (9.2)$$

The displayed simplicial object is the action groupoid. Groupoid objects are effective in an ∞ -topos [5, Theorem 6.1.0.6], so this homotopy quotient exists with its action identifications and all their coherences. It changes the treatment of labels, not the ownership of any asset.

Proposition 9.2 (Symmetries survive the quotient). *If $\mathcal{X} = \mathcal{S}$, M is a discrete set, and G is a discrete group, the component of $M \parallel G$ containing $m \in M$ is equivalent to $B\text{Stab}_G(m)$. Consequently*

$$\pi_0(M \parallel G) = M/G, \quad \pi_1(M \parallel G, [m]) \cong \text{Stab}_G(m),$$

and its homotopy groups in degrees greater than one vanish.

Proof. The action groupoid on the orbit of m is connected. Its full one-object subgroupoid at m has automorphism group $\text{Stab}_G(m)$. Inclusion is fully faithful and essentially surjective, hence an equivalence of groupoids; realization gives the asserted equivalence of spaces. The homotopy-group statements follow from the classifying space of a discrete group. $\square$

Thus the orbit set records a class of presentations, while the homotopy quotient additionally records their automorphisms. If M has higher structure arising from program equivalences or proof data, the quotient can retain that higher structure as well. One must not quotient again by symmetries already built into the chosen identity types. The univalent structured-type presentation and the explicit labelled action presentation are alternative constructions to be compared, not blindly superimposed.

The numerical example below is 0-truncated. It establishes a concrete instance of the finite theory, not a claim that every allocation problem needs higher homotopy. Higher semantics matters when identifications, automorphisms, and their coherent descent are part of what the system must retain.

10 A four-agent collective

Consider one epoch with four admitted members, two divisible resource coordinates, and one common productive asset. The resource units are core-hours and GiB-hours. There is no external supply. Let actual contributions equal the capacities in Table 1, and reserve $h = (4, 4)$ for maintenance. Total contribution is $(12, 16)$, leaving $b = (8, 12)$.

Member	Capacity and contribution $\kappa_a = c_a$	Admitted need n_a	Allocation x_a^*
a_1	$(8, 4)$	$(1, 2)$	$(2/3, 4/3)$
a_2	$(4, 8)$	$(3, 2)$	$(2, 4/3)$
a_3	$(0, 4)$	$(4, 4)$	$(8/3, 8/3)$
a_4	$(0, 0)$	$(4, 8)$	$(8/3, 16/3)$
Total	$(12, 16)$	$(12, 16)$	$(8, 32/3)$

Table 1: Need-based service with heterogeneous productive contribution. Each ordered pair uses the two stated resource units.

The exact service guarantee is

$$\lambda^* = \min \left\{ 1, \frac{8}{12}, \frac{12}{16} \right\} = \frac{2}{3}.$$

All four members receive two thirds of their admitted bundle. Member a_4 receives a positive allocation despite contributing no capacity; member a_1 contributes the most CPU capacity but receives the smallest CPU allocation because its admitted need is smallest. The remaining memory budget is $12 - 32/3 = 4/3$. CPU is the bottleneck. The two-thirds guarantee cannot be increased for all members while retaining the same budgets and need bundles.

Take $q = 3$ and $f = 1$. Then $2q = 6 > 5 = m + f$ and $q = m - f$. Any two quorum sets intersect in at least two members, one of whom is honest under the fault bound. No member can authorize a plan alone, and any three members can supply the required signer set if they endorse the same plan.

To exhibit an actual model, work first in $\mathcal{X} = \mathcal{S}$. Choose a singleton common knowledge type with universal read permission, the common owner for the asset, the displayed resource data, and a valid three-member certificate for this plan. To include the required relabellings, put $G = \text{Sym}(4)$ and take states $S = G \times \{0, 1\}$. At (g, i) , use the g -reindexed resource data; i records whether the plan has been published. Left multiplication on G gives the relabelling action. A finite control category has the identity arrows and one certified arrow $(g, 0) \rightarrow (g, 1)$ for each g , carrying the reindexed certificate. The initial state is $(1_G, 0)$. Both endpoints of every arrow satisfy Inv_{rad} , and relabelling preserves the transition rule. There are no nonidentity isomorphisms; the discrete simplicial nerve is therefore a complete Segal space. This realizes the constitutional and safety specification from the supplied data, without establishing live distributed implementation under failures.

For a nontrivial label symmetry example, instead take the discrete family of numerical states with four members, identical needs and contributions, uniform governance eligibility, and no distinguishing history. Such a state has stabilizer $\text{Sym}(4)$. Its component in this state's action quotient is $B\text{Sym}(4)$. Equal fractional service is invariant under this symmetry; selecting one member for an indivisible assignment invokes the obstruction in Theorem 8.1.

11 Interpretation, limits, and further formalization

The ambient ∞ -topos supplies a logic of context-dependent data and coherent identification. Common ownership, equal standing, and the admission of needs are chosen constitutional requirements. Safety follows from verified preservation and authenticated evidence. Allocation optimality follows from a specified objective and feasible set. Neither the political requirements nor universal implementation success follows from the axioms of an ∞ -topos.

The model separates several obligations that an implementation must discharge:

1. *Membership and authentication.* Distinct members must correspond to the intended constitutional identities. Permitting unrestricted replica creation would invalidate the interpretation of equal voting power and the assumed fault bound.
2. *Resource measurement.* Capacity and need data require an admitted measurement or verification procedure. An allocation certificate proves an implication from these inputs, not their physical accuracy.
3. *Execution refinement.* The implementation must supply the step interpretation required by Theorem 4.1, including failures that fall within the claimed assurance boundary.
4. *Coordination and availability.* Quorum safety, delivery, progress, and fair service over time are separate properties. A consensus protocol must supply its own timing and fault assumptions.
5. *Institutional scope.* Equal formal rights do not establish equal effective influence, truthful need reports, or the absence of external control. These require additional models and evidence.

There are several precise extensions. Time-indexed budgets and temporal service bounds would distinguish instantaneous fairness from starvation freedom. Indivisible jobs require a richer feasible-plan type and explicit symmetry-breaking data. Dynamic membership requires a certified relation between successive member types and quorum-overlap conditions across configurations. Formalization of the execution layer requires directed higher type theory or an explicit model of internal complete Segal objects. More general production can replace the ordered budget category

by a symmetric monoidal category of resource transformations. Each extension changes a stated assumption or adds a proof obligation; none is implied by increasing the categorical dimension.

The resulting representation is therefore a structured collective *inside* an ∞ -topos, equipped with its own directed execution category and internal proof obligations. Its finite fragment has a direct constructive interpretation, while its higher structure retains equivalences and coherence that a plain set of states would discard.

A Exact-arithmetic supplement and verification status

The accompanying source package contains `verify.py`, a standalone Python 3 program using rational arithmetic from `fractions.Fraction`. It implements (6.4), reproduces Table 1, verifies the ownership-independent allocation comparisons, and checks boundary cases with zero demand and zero capacity. A finite exhaustive check compares the resulting guarantee with all service vectors in $\{0, 1/2, 1\}^3$ for all three-member, two-resource demand matrices in $\{0, 1\}^{3 \times 2}$ and budgets in $\{0, 1, 2\}^2$. There are 576 such input instances. The program also enumerates the four-member quorum intersections and checks the symmetry statement on the fully symmetric four-member input.

These computations check the implementation of the finite formulas and the displayed example. They are not substitutes for the universal mathematical proofs and do not mechanize the ∞ -topos or HoTT development. The manuscript contains conventional mathematical proofs, not a proof-assistant-checked formalization. The build instructions and the checked scope are supplied in `README.md`; the bibliography includes primary-source links.

References

- [1] Bob Coecke, Tobias Fritz, and Robert W. Spekkens. A mathematical theory of resources. *Information and Computation*, 250:59–86, 2016. Author preprint: <https://arxiv.org/abs/1409.5531>.
- [2] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, 1985. <https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>.
- [3] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*. USENIX Association, 2011. <https://www.usenix.org/conference/nsdi11/dominant-resource-fairness-fair-allocation-multiple-resource-types>.
- [4] Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM*, 46(6):858–923, 1999. <https://cs.brown.edu/~mph/HerlihyS99/p858-herlihy.pdf>.
- [5] Jacob Lurie. *Higher Topos Theory*, volume 170 of *Annals of Mathematics Studies*. Princeton University Press, 2009. Author’s electronic edition. <https://people.math.harvard.edu/~lurie/papers/HTT.pdf>.
- [6] Karl Marx. Critique of the Gotha programme, 1875. Part I. English text, Marxists Internet Archive. <https://www.marxists.org/archive/marx/works/1875/gotha/ch01.htm>.

- [7] Charles Rezk. A model for the homotopy theory of homotopy theory. *Transactions of the American Mathematical Society*, 353(3):973–1007, 2001. <https://arxiv.org/abs/math/9811037>.
- [8] Emily Riehl and Michael Shulman. A type theory for synthetic ∞ -categories. *Higher Structures*, 1(1):147–224, 2017. Corrected author version: <https://arxiv.org/abs/1705.07442>.
- [9] Michael Shulman. All $(\infty, 1)$ -toposes have strict univalent universes, 2019. arXiv:1904.07004. <https://arxiv.org/abs/1904.07004>.
- [10] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. <https://homotopytypetheory.org/book/>.