

# How to Coordinate an Action Across Many Parties Without a Central Coordinator

If you need three or more independent parties to agree on and commit to a single action, and you cannot stand up a central coordinator, a shared broadcast channel, or a globally synchronized ledger, the usual toolbox fights you at every turn. This guide describes an architectural approach for reaching a coordinated outcome among specific named parties by carrying the coordination state inside the messages themselves and resolving it against a dynamically scoped quorum. It describes an architecture disclosed in U.S. Provisional Application No. 64/049,409, not a shipping library. The home inventive step is the N-Party Coordination inventive step.

---

## What You Are Building

You have three or more independent parties. Each one is under its own control, on its own hardware, possibly under its own organization or jurisdiction. You need them to converge on a single coordinated outcome: a joint decision, a quorum vote, an auction award, a role-differentiated handoff, a custody transfer, or a multi-authority approval. And you need to do this without a server in the middle that every party trusts, without a shared database they all write to, and without a global clock or ledger keeping them in lockstep.

This is a genuinely hard shape. The parties are named and specific (not an anonymous swarm), the outcome must be attributable to who contributed what, and there is no coordinator you can appoint because appointing one recreates the trust and availability problem you were trying to escape.

The approach below is drawn from the N-Party Coordination inventive step as disclosed in U.S. Provisional Application No. 64/049,409, which treats multi-party coordination as a first-class architectural primitive rather than an application built on top of a consensus engine. You will be implementing it yourself; this is the architecture, not a package.

## **Why the Obvious Approaches Fall Short**

The standard tools each solve a different problem than the one you have.

**Blockchain consensus protocols** (proof-of-work, proof-of-stake, and Byzantine-fault-tolerant variants) are built to make *all* network participants agree about *shared global state*. That is far more than you need. Your parties are specific and named; you do not care what the rest of the world believes, and you do not want to pay for global agreement or wait for block-commit finality when the coordination only concerns the parties actually involved.

**Multi-signature and threshold-signature schemes** (m-of-n signing) will aggregate signatures for you, but they aggregate them cryptographically without attaching any authority-chain semantics. You get "n keys signed," not "which party, under what authority, contributing what role." The signatures do not tell you who was allowed to participate or what each participant was attesting to.

**Auction and voting protocols** exist off the shelf, but each one is specialized to its own pattern (a single-commodity auction, a public-choice vote) and does not preserve a governance chain or compose with other coordination patterns.

**Secure multi-party computation** gives you privacy-preserving joint computation, but it provides no coordination semantics: no roles, no admission rules, no outcome attribution.

The structural gap is the same across all of them: none carries the *coordination state* with the parties, none grounds the coordination in the specific named participants who are physically or logically present, and none preserves a per-participant record of who did what. A central coordinator papers over the gap by holding that state centrally, which is exactly the dependency you are removing.

## The Architecture

The core move is to stop treating coordination as something a coordinator *does* and treat it as something the *messages carry*. Each party emits a self-describing, self-authenticating unit of state, and any party that receives enough of these units can compute the outcome for itself. There is no session, no handshake, no registrar.

**1. Carried-state observations.** Every contribution is a governed observation: a structured message that carries, as fields, everything a receiver needs to evaluate it in isolation. As disclosed, an observation carries an authority-credential field (who is speaking and under what authority), a spatial-reference field and a temporal-reference field (where and when), a time-to-live field (how long the contribution stays valid), a payload field (the party's actual attestation), and a lineage field carrying provenance with cryptographic attestation. Contribution is fire-and-forget: the disclosed contribution mechanism completes upon emission and requires no acknowledgment, handshake, delivery confirmation, or registration with a central authority. This is what removes the coordinator. State lives in the observations, not in a server.

**2. Participant admission over N parties.** The coordination primitive ingests governance-credentialed observations from N authority-credentialed parties, where N is three or more. A participant-admission step decides who is in. Admission is not a

single-factor check: each contributed observation passes through a composite admissibility evaluation, and admission can be disposition-weighted so that a party's standing affects whether and how it participates.

**3. Dynamically scoped quorum via proximity.** Instead of a fixed global membership, the relevant set of parties is scoped at coordination time. A spatial-proximity evaluator verifies participant spatial proximity and a temporal-proximity evaluator verifies participant temporal proximity, using mesh-derived coordinate and time references rather than assuming one global frame. This is what makes the quorum *dynamically scoped*: the parties that count are the specific named parties who are actually present in the relevant space-time window, not everyone on a network. Coordination is thereby grounded in physical (or logical) reality rather than in a global membership list.

**4. A coordination-pattern selector and role-differentiated attestations.** A governance-policy-defined coordination pattern specifies each party's role and the rules that turn contributions into an outcome. A role-differentiated attestation schema then specifies what content each role must contribute. One primitive covers consensus-required decisions, quorum-based resolutions, auctions, lead-and-follower formations, custody-transfer chains, federated aggregations, multi-authority approvals, and governed voting, all by parameterization rather than by writing a new protocol for each.

**5. An outcome function.** A coordination-outcome function evaluates the admitted, contributed observations to determine the coordinated outcome under the governance policy. Contributions can be authority-tier-weighted, so a higher-authority party can carry more weight than a lower one where the policy says so.

**6. Robustness mechanisms, unified.** The primitive integrates, rather than bolting on separately: a multi-round coordination engine for iterated ceremonies that converge to a terminal outcome; a Byzantine-robust mechanism that tolerates a governance-

policy-defined fraction of adversarial or failed participants; a partial-quorum and abandonment handler for ceremonies that never complete; and a dynamic-membership mechanism that lets a member be replaced mid-ceremony.

**7. Lineage as the record of truth.** A coordination-lineage recorder writes each participant admission, attestation, outcome determination, round transition, Byzantine event, abandonment, membership change, and cross-domain handoff into the governance-chain lineage field. There is no external ledger; the record travels with the coordination artifact and remains cryptographically attributable to each participant.

For coordination that crosses an authority boundary (one jurisdiction to another, one domain to another), the disclosed cross-domain coordination handoff mechanism transfers the operation from a relinquishing-domain authority to a receiving-domain authority while preserving governance, lineage, and coordination-state continuity, recording the transfer as a lineage-linked matched pair.

## **How to Approach the Build**

Work outward from the message, not inward from a server.

**Step 1: Define your carried-state observation.** Decide the concrete encoding of the fields above. The disclosure is explicit that the encoding is not itself the invention: fixed- or variable-length binary, TLV, CBOR, Protocol Buffers, FlatBuffers, Cap'n Proto, Avro, JSON, or XML are all viable as long as the enumerated fields are present. Pin down what your authority credential, spatial reference, temporal reference, TTL, payload, and lineage fields actually contain.

An illustrative interface sketch (faithful to the disclosed fields, not a library you can install):

```
Observation {  
  authority_credential // who, under what authority  
  device_hash          // continuity-based identity  
  spatial_reference    // where (mesh-derived or local frame)  
  temporal_reference   // when  
  time_to_live         // validity window  
  payload              // this party's role attestation  
  lineage              // provenance + attestation  
}
```

**Step 2: Choose your proximity references.** You need a spatial reference and a temporal reference that all parties can evaluate without a global clock or GPS dependency. The disclosure grounds these in mesh-derived coordinate and time frames; for your build, decide the equivalent (a local coordinate frame, a logical partition key, a cooperatively estimated time). This is what scopes your quorum, so it has to be something every candidate participant can independently check.

**Step 3: Implement admission as composite evaluation.** Do not gate on a single threshold. Score each incoming observation on multiple factors (authority, integrity, standing) and admit under a composite rule. Add disposition-weighting if some parties should be admitted more readily than others.

**Step 4: Encode the coordination pattern as policy, not code.** Write the pattern (consensus, quorum, auction, custody chain, and so on), the role assignment, and the outcome function as governance-policy parameters that a deployment supplies. The goal is that adding a new coordination pattern is a policy change, not a protocol rewrite.

**Step 5: Implement the outcome function against admitted contributions.** Compute the outcome purely from the admitted observations and their weights. Because the state is carried, any party with the same admitted set computes the same outcome; you do not need a coordinator to declare it.

**Step 6: Handle the unhappy paths explicitly.** Build the multi-round loop, a Byzantine tolerance fraction, a partial-quorum/abandonment path, and mid-ceremony membership replacement. These are first-class, not error handlers you add later.

**Step 7: Record everything to lineage.** Every admission, attestation, round transition, Byzantine event, and outcome goes into the lineage field with attestation. This is your audit trail and your dispute-resolution substrate, and it replaces the ledger.

**Step 8: Add cross-domain handoff only if you need it.** If coordination crosses an authority boundary, implement the relinquish/accept handoff as a lineage-linked pair with taxonomy translation between the two domains' authority contexts.

## **What This Does Not Give You**

This is an architecture, not a drop-in library. There is no package to install, no SDK, and nothing here "just works" on import. You implement the observation format, the proximity evaluators, the admission logic, the outcome function, and the robustness mechanisms yourself, against your own environment and threat model.

It is disclosed in a patent filing. It has not been presented here as a benchmarked or production-proven system, and this guide states no performance, latency, or throughput numbers because the disclosure grounds an architecture, not measured results. Any such numbers are yours to establish.

The approach assumes coordination among *specific named, credentialed parties* grounded in spatial and temporal proximity. If your problem is genuinely global agreement about shared state across an open, anonymous participant set, a consensus protocol is the right tool and this is not it. If your parties cannot be authority-credentialed, or cannot share any evaluable spatial or temporal reference, the proximity-scoped quorum has nothing to scope against, and the model does not apply.

The Byzantine tolerance is bounded by a governance-policy-defined fraction of faulty participants; beyond that fraction, ceremonies are expected to abandon rather than to produce a wrong outcome, and you must design for abandonment as a normal result.

## Disclosure Scope

The N-Party Coordination inventive step, including carried-state governed observations, proximity-scoped participant admission, governance-policy-parameterized coordination patterns and outcome functions, the multi-round, Byzantine-robust, partial-quorum, and dynamic-membership mechanisms, lineage-recorded outcomes, and cross-domain coordination handoff, is disclosed in U.S. Provisional Application No. 64/049,409. This guide is educational and describes an architectural approach a developer can build; it is not a shipping product, not a warranty of fitness or performance, and not an offer of software. Every mechanism described here traces to that filing; nothing beyond it is claimed.

---

## **N-Party Coordination** (</n-party-coordination>)

[All 40 steps → \(/inventive-steps\)](/inventive-steps)

Multi-party ceremonies grounded in physical proximity. Cross-domain authority handoff.

Provisional application

### **PRIMARY TECHNICAL DISCLOSURE**

- [N-Party Coordination: Physical-Proximity-Grounded Multi-Party Settlement \(/articles/n-party-coordination-physical-proximity-grounded-multi-party-settlement\)](/articles/n-party-coordination-physical-proximity-grounded-multi-party-settlement)

### **SECONDARY TECHNICAL**

- [Role-Differentiated Multi-Party Attestation \(/articles/n-party-coordination/role-differentiated-attestation\)](/articles/n-party-coordination/role-differentiated-attestation)
- [Proximity-Grounded Multi-Party Coordination \(/articles/n-party-coordination/proximity-grounded-coordination\)](/articles/n-party-coordination/proximity-grounded-coordination)

- [Coordination Pattern Plurality \(/articles/n-party-coordination/coordination-pattern-plurality\)](/articles/n-party-coordination/coordination-pattern-plurality)
- [Byzantine-Robust Multi-Party Coordination \(/articles/n-party-coordination/byzantine-robust-coordination\)](/articles/n-party-coordination/byzantine-robust-coordination)
- [Partial Quorum Handling \(/articles/n-party-coordination/partial-quorum-handling\)](/articles/n-party-coordination/partial-quorum-handling)
- [Dynamic Membership for Multi-Party Coordination \(/articles/n-party-coordination/dynamic-membership\)](/articles/n-party-coordination/dynamic-membership)
- [Cross-Pattern Coordination Composition \(/articles/n-party-coordination/cross-pattern-composition\)](/articles/n-party-coordination/cross-pattern-composition)
- [Cross-Domain Coordination Handoff \(/articles/n-party-coordination/cross-domain-handoff\)](/articles/n-party-coordination/cross-domain-handoff)
- [Intermodal Freight Coordination Embodiment \(/articles/n-party-coordination/intermodal-freight\)](/articles/n-party-coordination/intermodal-freight)
- [Airspace Handoff Coordination \(/articles/n-party-coordination/airspace-handoff\)](/articles/n-party-coordination/airspace-handoff)
- [Medical Patient Transfer Coordination \(/articles/n-party-coordination/medical-patient-transfer\)](/articles/n-party-coordination/medical-patient-transfer)

## **APPLICATIONS · GENERAL**

- [Coalition Defense Coordination Across National Authorities Without a Shared Data Fabric \(/articles/n-party-coordination/coalition-defense-coordination\)](/articles/n-party-coordination/coalition-defense-coordination)
- [Care Team Coordination Across Hospitals, Specialists, and Home Care: A Credentialed Handoff Substrate \(/articles/n-party-coordination/healthcare-team-coordination\)](/articles/n-party-coordination/healthcare-team-coordination)
- [Multi-Stakeholder Supply Chain Coordination \(/articles/n-party-coordination/multi-stakeholder-supply-chain\)](/articles/n-party-coordination/multi-stakeholder-supply-chain)
- [Multi-Institution Research Collaboration Without Surrendering Institutional Sovereignty \(/articles/n-party-coordination/academic-research-collaboration\)](/articles/n-party-coordination/academic-research-collaboration)
- [Multi-Site Clinical Trial Coordination Across Sponsors, CROs, Sites, and IRBs \(/articles/n-party-coordination/clinical-trial-multi-site\)](/articles/n-party-coordination/clinical-trial-multi-site)
- [Multi-Agency Humanitarian Aid Coordination Across Agencies, Governments, and Donors \(/articles/n-party-coordination/humanitarian-aid-coordination\)](/articles/n-party-coordination/humanitarian-aid-coordination)
- [Incident Command System Multi-Agency Coordination Software for NIMS Compliance \(/articles/n-party-coordination/incident-command-system-coordination\)](/articles/n-party-coordination/incident-command-system-coordination)
- [Cross-Utility Grid Coordination: Settling Interchange Across ISO, RTO, and Balancing-Authority Seams \(/articles/n-party-coordination/smart-grid-multi-utility\)](/articles/n-party-coordination/smart-grid-multi-utility)

## **APPLICATIONS · SPECIFIC**

- [Epic EMR Care Coordination vs Payload-Bound N-Party Governance \(/articles/n-party-coordination/epic-emr-care-coordination\)](/articles/n-party-coordination/epic-emr-care-coordination)
- [NATO FMN vs Governed N-Party Coalition Coordination \(/articles/n-party-coordination/nato-fmn\)](/articles/n-party-coordination/nato-fmn)

- [Project Maven and JADC2 vs Cross-Authority Coordination Substrate \(/articles/n-party-coordination/project-maven-jadc2\)](/articles/n-party-coordination/project-maven-jadc2).
- [Salesforce Commerce Cloud vs Governed N-Party Coordination \(/articles/n-party-coordination/commerce-cloud-coordination\)](/articles/n-party-coordination/commerce-cloud-coordination).
- [HL7 FHIR vs Governed N-Party Coordination: The Layer Above Exchange \(/articles/n-party-coordination/hl7-fhir\)](/articles/n-party-coordination/hl7-fhir).
- [Oracle Fusion Cloud vs Governed N-Party Coordination \(/articles/n-party-coordination/oracle-fusion-coord\)](/articles/n-party-coordination/oracle-fusion-coord).
- [Governed Multi-Party Supply Chain Beyond SAP S/4HANA: N-Party Coordination \(/articles/n-party-coordination/sap-supply-chain\)](/articles/n-party-coordination/sap-supply-chain).
- [ServiceNow Alternative for Cross-Organization Governed Coordination: N-Party Co-Signature Beyond the Now Platform \(/articles/n-party-coordination/servicenow-coordination\)](/articles/n-party-coordination/servicenow-coordination).

---

[N-Party Coordination overview → \(/n-party-coordination\)](/n-party-coordination)