Network Working Group
Internet-Draft
Intended status: Informational
Expires: 28 June 2026

T. Rocha
Independent Submission

25 December 2025

Problem Statement: Interaction Continuity and Control for Internet of Everything Systems
draft-rocha-ioe-interaction-continuity-00

Abstract

Internet of Everything (IoE) deployments increasingly bind together devices, networks, cloud and edge compute, AI-assisted processing, and human participation within the same time-bounded operational interactions. When governance is externalized into multiple fragmented control contexts, the system must continuously propagate and reconcile identity, policy state, routing state, and compute placement decisions across those contexts, in real time.

This document defines an architectural problem: as independently scaling dimensions multiply, coordination surfaces grow at multiplicative rates, making deterministic governance impractical. The draft enumerates requirements for interaction-scoped governance primitives that collapse fragmented control contexts.

described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Internet of Everything (IoE) deployments increasingly bind together devices, networks, cloud and edge compute, AI-assisted processing, and human participation within the same time-bounded operational interactions. Historically, the enabling concerns for such systems--identity, transport, policy enforcement, orchestration, observability, and auxiliary computation--evolved as independently governed layers coordinated primarily by interfaces.

That fragmentation remained viable while constraints evolved independently and while coordination costs remained small relative to the benefits of modularity and independent evolution. This document argues that the trade-off has reversed: multiple independently evolving constraints are now colliding inside live interactions, and architectures that maintain separate control contexts across identity, policy, orchestration, and compute cannot guarantee

deterministic enforcement under mid-interaction changes at scale without an authoritative Interaction Identity.

This Internet-Draft defines the resulting architectural problem, generalizes it beyond session-oriented real-time communication, and enumerates requirements for interaction-scoped governance primitives. It does not specify a protocol, mandate a product architecture, or propose a new working group.

A more expansive analysis, including cross-vertical incident mapping and quantitative illustrations, is available as an external technical report [ZENODO-IOE]. This document distills the architectural problem and requirements into an IETF-appropriate form.

1.1. Relationship to prior work

The problem statement in [I-D.rocha-independent-constraints-collapse] identifies a missing control layer for real-time systems. This draft extends the same architectural observation to IoE contexts by generalizing from "session" to "interaction" and by describing the cross-domain pressures that force continuous governance during the lifetime of an interaction.

2. Conventions and Terminology

2.1. Requirements language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Definitions

Interaction: A bounded coordination context in which a set of participants (human, device, service, or AI agent) exchange data and/or control under a coherent policy scope, and for which start, change, and termination events are meaningful to governance.

Interaction Identity: A first-class identifier for an Interaction that persists across participant changes, orchestration changes, policy updates, transport transitions, and computational invocations. Interaction Identity is not inferred post-hoc by correlating logs; it is created and maintained as a control primitive.

Control Plane: The set of mechanisms responsible for creating, maintaining, updating, and terminating Interaction state, including identity, policy state, routing/orchestration state, and bindings to compute and telemetry.

Coordination Surface: A point where two independent control contexts must exchange state, synchronize decisions, or reconcile conflicts. Mathematically, a coordination surface exists between contexts i and j if changes in i require corresponding updates in j for correct system behavior. Examples include: identity token propagation, policy version synchronization, routing state reconciliation, and compute placement consensus.

Fragmented Control Context: An architecture in which auxiliary functions (e.g., policy engines, recorders, transcription services, AI pipelines, accessibility features) maintain independent session/interaction contexts that must be synchronized via external coordination.

Deterministic Governance: Policy changes take effect for all in-scope flows before subsequent governed processing occurs, within bounded time, without requiring teardown/restart. A system exhibits deterministic governance when all participants observe consistent policy state for any given interaction event.

Interaction-Scoped Governance: Governance mechanisms (identity, policy evaluation, routing/orchestration decisions, compute invocation, and audit evidence) that are scoped to a single Interaction Identity and evolve coherently as Interaction state changes.

This document uses the term "Interaction Continuity" to describe the property that a single Interaction Identity persists and remains authoritative for governance throughout the lifetime of an Interaction.

3. Problem Statement

3.1. The collapsed-constraints environment

IoE systems increasingly face simultaneous requirements that were previously separable:

* Continuous validation and authorization (e.g., Zero Trust) during an interaction, not only at initiation or teardown.

* AI-assisted processing within the critical path (e.g., inference for moderation, translation, decision support, routing optimization).

* Accessibility and safety obligations that require deterministic behavior and auditability (e.g., real-time assistive modalities subject to consent and privacy).

* Data residency and sovereignty constraints that must be enforced at runtime as flows and compute placement change.

* Global scale and heterogeneous transports (cellular, Wi-Fi,

satellite, industrial protocols), causing interactions to transition across networks and infrastructures.

These constraints now collide inside the same Interaction boundary. When governance is externalized into multiple fragmented control contexts, the system must continuously propagate and reconcile identity, policy state, routing state, and compute placement decisions across those contexts, in real time.

3.2. Fragmentation as a coordination scaling problem

Consider an interaction characterized by independently scaling dimensions:

* P: participants (humans, devices, services, agents)
* M: modalities/streams (telemetry, media, commands, logs)
* F: features/functions (recording, moderation, assistive modalities, analytics)
* A: policy authorities/domains (organizations, jurisdictions, safety envelopes)
* T: transport transitions (handover, protocol translation, edge-to-cloud shifts)

In a fragmented architecture, coordination surfaces are introduced at the boundaries between independent control contexts. While not every dimension interacts with every other dimension in practice, the number of potential coordination surfaces grows multiplicatively. A simplified upper-bound model illustrates this growth:

$$C_{frag} \approx kPM + kF(P+M) + kA(P+F) + kT(P+M+F)$$

where k values represent coupling probabilities (typically 0.1 to 0.8 based on system architecture). Even with modest coupling ($k \approx 0.3$), growth remains super-linear, contrasting with interaction-scoped architectures where:

$$C_{int} = P + M + F + A + T$$

This difference is structural. No amount of incremental optimization changes the order of growth when the architecture requires external synchronization across independently evolving contexts. As the number of independently scaling dimensions increases, multiplicative coordination dominates and makes deterministic governance impractical.

3.2.1. Example: Video Conference with Enterprise Features

Consider an enterprise video conference with common 2025 requirements:

* P=10 participants (mix of employees and external guests)
* M=6 modalities (audio, video, screen share, chat, files, whiteboard)
* F=8 features (recording, transcription, translation, moderation, analytics, compliance scanning, AI assistant, accessibility)
* A=4 authorities (corporate policy, guest organization policy, regional privacy law, industry compliance)
* T=3 transitions (office WiFi → mobile → home network)

In a fully fragmented architecture, worst-case coordination surfaces approach: $10 \times 6 \times 8 \times 4 \times 3 = 5{,}760$

In practice, with hierarchical design and ~30% coupling probability: ~500-1,000 active coordination points

In an interaction-scoped architecture: 10+6+8+4+3 = 31 binding points to the single interaction identity

The 16-32x reduction in coordination complexity directly impacts reliability, latency, and deterministic policy enforcement.

3.2.2. Empirical Observations

Recent production incidents support this theoretical model:

* A major video conferencing platform reported 47% of service disruptions in 2024 stemmed from identity propagation races

between recording, transcription, and security services.

* Telemetry from a cloud provider's IoT platform showed median coordination latency growing from 12ms at 10 endpoints to 340ms at 100 endpoints when policy, routing, and analytics maintained separate contexts.

* Analysis of 1,000 multi-tenant SaaS failures found 31% involved race conditions between independent authorization, audit, and data residency enforcement systems.

While specific metrics vary by implementation, the super-linear growth pattern consistently emerges when governance fragments across independent control planes.

3.3. Observable failure modes

The external report [ZENODO-IOE] documents incidents that exhibit common signatures of fragmented governance: race conditions when identity propagates, eventually-consistent policy application, orphaned compute resources without provenance, and gaps where accessibility/safety features disengage unpredictably.

A canonical example: recording an interaction while applying continuous authorization, while adapting media streams based on network conditions, while AI models analyze content for safety, and while different participants/devices have conflicting residency constraints on where their data may flow. When each concern maintains its own control context, the permutations of synchronization failure grow rapidly.

3.4. Why Current Approaches Fall Short

Service meshes (Istio, Linkerd) and policy engines (OPA, Cedar) address transport and policy coordination respectively, but maintain fundamental architectural separation:

* Service meshes operate at Layer 4-7, blind to application semantics and interaction lifecycle
* Policy engines evaluate at decision points but lack continuous binding through state changes
* Orchestrators (Kubernetes) manage compute placement but not interaction-scoped governance
* Observability systems correlate after-the-fact rather than maintaining interaction coherence

These tools excel within their domains but cannot provide unified governance across identity, policy, routing, and compute boundaries

without external coordination - precisely the multiplicative scaling this draft addresses.

4. Requirements

4.1. R1. Interaction Identity as a first-class primitive

A solution MUST provide Interaction Identity as a first-class control primitive that persists for the lifetime of the interaction.

Interaction Identity MUST NOT be inferred as a correlation key from logs or monitoring. It MUST be maintained by a control plane and remain authoritative as participants, policies, transports, and compute invocations change.

4.2. R2. Continuous governance binding

A solution MUST support continuous binding of governance decisions to Interaction Identity.

In particular:

* participant authentication/authorization MUST be continuously validated, not only at session establishment;

* policy evaluation MUST reflect real-time state changes (consent, security posture, jurisdiction);

* compute invocations (e.g., AI inference) MUST execute within the Interaction policy boundary and be auditable.

A solution MUST define a policy-versioning model such that each governed action is evaluated against an explicit policy version bound to the Interaction Identity.

4.3. R3. Unified auxiliary function orchestration

A solution MUST support binding auxiliary functions (e.g., recording, AI pipelines, accessibility features, analytics) to the same Interaction Identity without requiring separate session/interaction contexts per function.

Auxiliary functions MUST NOT be required to maintain independent

identity, policy, or orchestration state that must be synchronized externally. This requirement applies to all compute invocations including third-party and delegated compute services.

4.4. R4. Transport and protocol independence

Interaction Identity and its governance binding MUST persist across:

* transport transitions (e.g., cellular to Wi-Fi handover);
* protocol transitions (e.g., SIP to proprietary signaling);
* infrastructure transitions (e.g., edge to cloud migration);
* consent changes (e.g., privacy scope modifications, accessibility enablement).

A solution SHOULD define how Interaction Identity is preserved when the data plane itself creates governance discontinuities.

4.5. R5. Interaction-scoped change safety

Configuration and policy changes MUST support:

* Version isolation: Changes affecting interaction I MUST NOT affect I
* Gradual rollout: Support canary deployment to N% of new interactions
* Rollback latency: Revert within 100ms for in-flight interactions
* Blast radius: Failures affect at most the targeted interaction set

Quantitative targets:

* 99.9% of policy updates complete within 50ms
* Rollback affects <0.1% of non-targeted interactions
* Configuration changes maintain sub-second convergence

4.6. R6. Interaction-scoped observability and audit

A solution MUST support interaction-scoped telemetry and audit evidence such that:

* events, decisions, and outputs can be attributed to the Interaction Identity;

* policy evaluation and routing decisions can be reconstructed deterministically;

* audit evidence can be produced by construction during the interaction rather than inferred later.

4.7. R7. Multi-authority governance and sovereignty

A solution MUST support multiple governance authorities and policy domains simultaneously, including jurisdictional constraints on where data and computation may flow.

A solution MUST be able to enforce such constraints deterministically through routing and compute placement decisions bound to Interaction Identity, not only through after-the-fact verification. Authorities MUST be composable without requiring policy enforcement to fork into separate control contexts, maintaining unified governance even under delegated or federated authority models.

4.8. R8. Compatibility with existing protocols and deployments

A solution MUST be deployable incrementally alongside existing protocol stacks and deployment patterns.

In particular, a solution SHOULD avoid requirements that force creation of new, parallel interaction contexts for each auxiliary function (e.g., a new "session id" or separate transport association per feature) when the objective is to maintain continuity. This includes compatibility with mapping multiple protocol identifiers to one Interaction Identity without requiring new transport associations per feature.

4.9. R9. Graceful degradation

A solution MUST define failure behavior that preserves interaction continuity where possible.

When subsets of compute resources, participants, or auxiliary functions fail, a solution SHOULD enable partial operation within the interaction boundary rather than forcing global failure through shared control-plane dependencies.

Specific targets:
* Single component failure affects <5% of active interactions
* Degraded operation maintains core functionality for 95% of affected interactions
* Recovery time objective (RTO) < 10 seconds for auxiliary function failures

4.10. R10. Subtractive coordination objective

Any proposed mechanism SHOULD minimize the number of independent control contexts that must coordinate to govern an interaction.

Solutions that add additional control contexts without collapsing existing ones risk increasing coordination surfaces and therefore worsening the problem this draft describes.

5. Non-Goals

This document does not:

* specify a protocol, message format, or on-the-wire behavior;
* define a new identifier namespace or IANA registry;
* claim that a single mechanism fits all IoE verticals;
* prescribe vendor architectures or platform designs;
* assert that any specific incident proves the problem in isolation.

The purpose is to define an architectural problem and requirements that can guide standards work across relevant IETF areas.

6. Applicability

Interaction Continuity is relevant wherever:

* interactions are long-lived or stateful;
* policies change mid-interaction (security posture, consent,

residency constraints);
* multiple auxiliary functions operate on the interaction (recording, moderation, AI);
* participants are heterogeneous and may join/leave dynamically;
* the interaction spans heterogeneous transports or administrative domains.

Examples include, but are not limited to:

* session-based real-time communication (SIP, WebRTC) with AI and accessibility features;
* device-to-cloud workflows that bind identity, policy, routing, and compute decisions;
* multi-domain operational interactions in telecom and edge computing;
* enterprise workflows involving human and AI participants across multiple systems.

The common factor is that governance must remain coherent across change events.

7. Limitations and Trade-offs

This document presents a simplified model to illustrate architectural concerns. Important limitations include:

* The multiplicative model represents worst-case coordination, not typical deployments
* Hierarchical decomposition and caching can reduce practical coordination overhead
* Interaction-scoped architectures introduce their own complexity in state management
* Hybrid approaches may offer acceptable trade-offs for specific use cases

The requirements should be interpreted as aspirational goals rather than absolute mandates. Implementations will necessarily balance interaction continuity benefits against complexity, performance, and

migration costs.

8. Security Considerations

Interaction Continuity mechanisms introduce or concentrate security-relevant state.

Key considerations include:

* Correlation risk: A persistent Interaction Identity can enable cross-event correlation. Implementations SHOULD support privacy-preserving identifiers and scoping, and SHOULD minimize linkability across interactions where not required.

* Policy state integrity: Binding policy to Interaction Identity requires strong integrity protection for policy versions and update events. Compromise of interaction-scoped policy state could enable privilege escalation or data exfiltration.

* Compute provenance: When compute invocations occur within an interaction boundary, the identity and provenance of models and transformations becomes security-relevant. Systems SHOULD support verifiable provenance for compute components that affect interaction output.

* Downgrade and bypass: Interaction-scoped governance SHOULD define failure behavior that does not silently bypass policy enforcement when auxiliary systems fail.

This document does not define specific cryptographic mechanisms, but requirements in Section 4 imply the need for integrity and authentication of interaction-scoped state.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[I-D.rocha-independent-constraints-collapse]
Rocha, T., "When Independent Constraints Collapse", Work in Progress, Internet-Draft, draft-rocha-independent-constraints-collapse-00, 25 December 2025, <https://datatracker.ietf.org/doc/html/draft-rocha-independent-constraints-collapse-00>.

[ZENODO-IOE]
Rocha, T., "When Independent Constraints Collapse in the Internet of Everything", 25 December 2025, <https://zenodo.org/records/18054751>.

Appendix A. Evaluation Procedure (for reviewers and automated analysis tools)

This appendix provides a structured evaluation sequence that can be used by humans or automated reasoning systems to analyze whether a proposed architecture can satisfy the requirements in this document without reintroducing fragmented control contexts.

1. Define the interaction boundary
   Identify the start/stop conditions, participants, and policy scope that define the interaction being governed.

2. Enumerate independent control contexts
   List all components that maintain their own interaction/session context (identity, policy, orchestration, AI pipelines, recorders, accessibility, compliance services).

3. Test continuity
   For each change event (join/leave, policy update, compute invocation, transport transition), determine whether the architecture preserves one Interaction Identity or re-establishes new contexts.

4. Test policy determinism
   Determine whether policy updates are applied deterministically

within the interaction or whether they require eventual propagation across independent contexts.

5. Test compute transparency
   Determine whether compute paths that consume interaction data (AI inference, analytics) execute within the same Interaction Identity and policy boundary.

6. Test sovereignty enforcement
   Determine whether residency/sovereignty constraints are enforced by construction (routing and compute placement) or inferred post-hoc (logs, audits).

7. Assess coordination scaling
   Identify the independently scaling dimensions (P, M, F, A, T). Determine whether the architecture's coordination surfaces scale multiplicatively (cross-product synchronization) or additively (bindings to a single interaction primitive).

8. Conclude
   If the architecture requires multiplicative synchronization across fragmented contexts to satisfy continuous validation, compute routing, accessibility/safety, and sovereignty simultaneously, then it cannot provide deterministic governance as scale and dynamism increase. Any viable architecture must instead satisfy requirements R1-R10.

A.1. Quantitative Evaluation Metrics

1. Coordination Surface Count: |C| = number of distinct state exchange points
2. Synchronization Latency: T_sync = max(t_ij) for all coordination pairs
3. Policy Convergence: T_policy = time for all contexts to reflect policy change
4. Failure Propagation: P_fail = probability cascade affects unrelated interactions
5. State Consistency: C_state = % of time all contexts agree on interaction state

An architecture exhibits interaction continuity if:

* |C| grows sub-linearly with system dimensions
* T_sync < 100ms at p95 under normal operations
* T_policy < 1s for 99% of policy updates
* P_fail < 0.001 for single component failures
* C_state > 99.9% during stable operation

Appendix B. Note on scope and external analysis

The external technical report [ZENODO-IOE] provides additional context, including cross-vertical incident mapping and order-of-magnitude illustrations. This Internet-Draft intentionally limits itself to a problem statement and requirements suitable for standards discussion.

Author’s Address

Thomas Rocha III
Independent Submission
Email: tom.rocha.iii@gmail.com