Identity Failure Chain Under Interruption

When interruption occurs, observation-based identity inference degrades, leading to identity ambiguity. In the absence of preserved identity authority, systems commonly compensate by creating new identities, reinitializing identity state, or overwriting preserved information. These actions propagate into duplication, drift, and distributed state divergence, degrading system correctness and triggering downstream recovery failures.

Purpose

This document describes the common architectural failure sequence that occurs in systems relying on continuous observation when interruption conditions arise.

It is explanatory, not prescriptive.

1. Interruption Occurs

A system operating under real-world conditions experiences interruption, including one or more of the following:

- Degraded telemetry
- Delayed or reordered data
- Partial unavailability
- Sensor occlusion or loss
- Asynchronous execution across components

Interruption is not an exceptional condition; it is a normal operating reality for distributed, cyber-physical, and autonomous systems.

2. Observation-Based Identity Inference Degrades

When continuous observation is assumed:

- Identity association depends on uninterrupted telemetry
- Confidence thresholds or inference heuristics degrade
- The system can no longer reliably associate incoming data with an existing identity instance

At this stage, the system has lost certainty, not the entity itself.

3. Identity Ambiguity Emerges

As inference confidence drops:

- The system cannot determine whether incoming signals correspond to:
    - an existing entity,
    - a new entity,
    - or a previously known entity that was temporarily unobserved

Identity ambiguity is an architectural condition, not a data quality or sensing issue.

4. Compensating Behaviors Trigger

In the absence of preserved identity authority, systems commonly respond by one or more of the following:

- Creating a new identity instance
- Reinitializing identity state
- Overwriting or superseding preserved state with unvalidated telemetry
- Allowing parallel subsystems to infer identity independently

These behaviors are often unintended and emerge from local correctness assumptions.

5. Identity-Level Failures Propagate

As a result, one or more identity failures occur:

- Duplicate identity instantiation
- Identity drift over time
- Incorrect re-association after recovery
- Distributed state divergence across components
- Reset cascades triggered by downstream inconsistencies

At this point, the system may still be operational but is no longer internally coherent.

6. Downstream System Correctness Degrades

Identity failures propagate into higher-level system behavior, including:

- Planning, control, or decision logic operates on inconsistent state
- Safety, performance, or mission guarantees erode
- Recovery actions compound rather than resolve the issue

Failures become harder to diagnose because they surface far downstream from the original interruption.

7. Manual Intervention Becomes Necessary

In mature systems, identity failures are often resolved through:

- Operator intervention
- Manual resets
- Offline reconciliation
- Ad hoc patches or heuristics

These interventions restore function temporarily but do not address the underlying architectural cause.

Key Observation

The root failure is not interruption.
The root failure is the absence of architectural mechanisms to preserve identity authority and state across interruption.

Without those mechanisms, interruption initiates a predictable and repeatable failure chain.

Relationship to SISMIQ Continuity Requirements

The SISMIQ Continuity Requirements define the minimum architectural behaviors required to break this failure chain by:

- Preserving identity authority across interruption
- Maintaining persistent internal identity representations
- Gating re-association upon recovery
- Preventing downstream identity duplication
- Ensuring continuity across distributed execution domains

Reference Use

This document may be used as:

- Architecture reference material
- Design review context
- Procurement or requirements justification
- Failure analysis framing
- Architectural discussion and failure analysis framing

It does not prescribe implementation techniques or system topologies.

Related Reference Material

SISMIQ Continuity Requirements (SCR)
Canonical architectural requirements for identity continuity under interruption.

Architectural Note: Identity Continuity Under Interruption
Contextual discussion of architectural implications and design tradeoffs.

Continuity Compliance Checklist
Self-assessment tool mapping system behavior to continuity requirements.

Procurement / RFP Language Pack
Specification-ready language for solicitations and contracts.

For architecture or specification discussion:

contact@sismiq.io