Pipeline Resilience Playbook

The Pipeline Resilience Playbook defines how the sensor-fusion pipeline recovers from the outages, data gaps, stale feeds, and partial failures that are normal operating conditions for environmental and IoT telemetry. A pipeline that only works when every sensor is healthy is not production-ready. This playbook codifies the ingestion monitoring posture, the procedures for handling source outages, the backfill discipline when connectivity resumes, and the stale-feed detection rules that prevent silent quality degradation from going unnoticed. It reuses the fallback patterns from s1-fallback-architecture-blueprint and connects to the alert emission rules in e5-alert-design-spec.

1. Ingestion Monitoring

The pipeline must continuously monitor its own health, not just the health of individual sensors. Operators who wait for an alert to discover the pipeline is broken have already failed silently.

1.1 Watchdog metrics

Metric	Definition	Warning threshold	Critical threshold
ingest_rate	Records received per minute per source	< 50% of expected rate for 3 consecutive windows	0 records for 2× emission_frequency
normalization_reject_rate	Fraction of records failing normalization	> 5% of window	> 20% of window
fusion_staleness_fraction	Fraction of measurement types with ≥ 1 stale source	> 25% of active types	> 50% of active types
alert_suppression_rate	Suppressed-to-fired ratio per rule	> 10:1 in a 1-hour window	> 50:1 (alarm is likely misconfigured)
duplicate_discard_rate	Duplicate records per source per window	> 5% of ingested records	> 20% (likely timestamp bug)
conflict_group_rate	Conflict groups per fusion window	> 2 per window per measurement type	> 5 (authoritative source conflict; governance issue)

Watchdog metrics are published to the pipeline monitoring channel in near-realtime. Any crossing of a critical threshold emits a severity: high alert via e5-alert-design-spec rule feed-gap-sensor or an equivalent pipeline-health rule.

1.2 Source heartbeat monitoring

Every source registered in e5-telemetry-source-map with an emission_frequency of hourly or faster must emit a heartbeat record at least once per 5 × emission_frequency_seconds. Heartbeat absence triggers the sensor-offline handling defined in e5-normalization-layer §4 and downgrades the source's confidence_band automatically. The downgrade is a provenance event, not a silent state change.

2. Outage Handling

An outage is any period during which a source is not delivering expected data. Outages are classified by scope:

Outage class	Scope	Immediate action
source-outage	Single source offline	Demote band, continue fusion with remaining sources, emit C6 payload at severity: medium with suggested_fallback: increase-monitoring-frequency
partial-outage	Multiple sources of the same measurement_type offline simultaneously	Demote composite band; if only low-band sources remain, halt alert firing for that type; emit severity: high to pipeline ops
pipeline-outage	Normalization or fusion layer itself unresponsive	Escalate immediately per s1-fallback-architecture-blueprint AwaitingHITL path; do not silently produce a stale operational view
connectivity-outage	Upstream network segment unavailable	Enter offline-mode operation per §2.1; emit a single outage event; do not flood retry attempts

2.1 Offline-mode operation

When a connectivity outage is detected (3 consecutive heartbeat misses on ≥ 50% of active sources), the pipeline enters offline mode:

1. The last-known-good operational view is retained with a stale_since_utc timestamp attached.

2. Alert firing is suspended for measurement_types whose only contributing sources are offline.

3. A single pipeline-connectivity-outage event is emitted to the escalation channel.

4. The pipeline polls for connectivity recovery at exponential backoff intervals: 30 s, 60 s, 120 s, 240 s, max 300 s.

5. On recovery, the pipeline does NOT fire all suppressed alerts retroactively. It resumes normal operation from the recovery timestamp and triggers a backfill procedure (§3) to fill the gap in historical records.

Reusing the s1-fallback-architecture-blueprint retry policy: per-step cap = 3, per-run cap = 6, backoff base = 2 s, max 60 s for individual source reconnection attempts. Connectivity outage polling uses the longer 30–300 s schedule above because mass-reconnection storms on a sensor network are a known reliability failure mode.

3. Backfill Procedure

When a source resumes after a gap, historical data may be available via the source's API or buffer. Backfill is a deliberate act, not automatic catch-up.

3.1 Backfill eligibility

A source is eligible for backfill if:

- The gap duration is ≤ 24 hours.
- The source provides a time-range query API or a local buffer.
- The confidence_band of the source is medium or higher at the time of backfill.

Sources with reliability_grade: C are not backfilled automatically; their historical data must be reviewed by an operator before injection.

3.2 Backfill steps

1. Log backfill intent. Record {source_id, gap_start_utc, gap_end_utc, triggered_by} in the provenance store before any data movement.

2. Fetch gap data. Query the source's buffer for records in the gap window.

3. Normalize. Pass all backfill records through the normalization layer with a backfill: true flag in the normalized record.

4. Tag and inject. Insert normalized backfill records into the historical store with the original timestamp_utc values. They are NOT re-fused into the live operational view.

5. Re-evaluate historical alerts. Run the alert rules against the backfilled fusion output for the gap period. Any critical or high alerts that would have fired are emitted as retroactive-alert events with status: informational — they do not trigger operational actions since the gap has already passed.

6. Log completion. Record {source_id, backfill_record_count, first_record_utc, last_record_utc, retroactive_alerts_found} in the provenance store.

3.3 Backfill conflicts

If backfill records conflict with records already in the historical store (same source_id, same timestamp_utc, different value), apply the same conflict resolution rules from e5-fusion-logic-map §5. Log the conflict and the resolution in the provenance store.

4. Stale-Feed Detection

A stale feed is subtler than an offline source. The source continues to emit records but the values are not changing, the timestamp is not advancing normally, or the data is a replay of prior values.

4.1 Staleness indicators

Indicator	Detection method	Staleness threshold
Value freeze	Standard deviation of fused_value over the last 10 fusion windows = 0 for a measurement type expected to vary	Applies for sources with expected variance > 0.01 units
Timestamp stall	timestamp_utc in consecutive records is identical or advancing at < 10% of expected rate	3 consecutive occurrences
Implausible repetition	Same raw_value (down to 4 decimal places) repeated ≥ 5 consecutive records	Any source
Divergence from network	Source value deviates from the fused network composite by > 3σ for ≥ 5 consecutive windows	Applies when ≥ 2 other sources are available

When any staleness indicator fires:

1. The source is flagged stale: true in the operational source registry.

2. Its confidence_band is automatically downgraded one notch.

3. A severity: medium alert is emitted to the pipeline monitoring channel.

4. The source owner is notified via the channel declared in e5-telemetry-source-map.

5. If the source is the sole authoritative source for a measurement_type used in a critical alert rule, the severity escalates to high and a Tier 2 escalation fires per s1-threshold-escalation-spec.

4.2 Stale-feed recovery

A stale feed is not automatically re-trusted when it starts varying again. Recovery requires:

1. An operator confirms the source is producing valid data (visual inspection or calibration check).

2. last_validated is updated in e5-telemetry-source-map.

3. The stale: true flag is cleared and confidence_band is restored.

4. A recovery event is logged to the provenance store.

5. Pipeline Health Dashboard

The pipeline resilience state is summarized in a health dashboard with these four status indicators:

Indicator	Green	Yellow	Red
Source coverage	≥ 80% of sources online and non-stale	60–79%	< 60%

Indicator	Green	Yellow	Red
Fusion completeness	All active measurement types have ≥ 1 medium+ source	Any type has only low-band sources	Any type has no contributing sources
Alert confidence	≥ 80% of fired alerts have confidence_band: medium or higher	60–79%	< 60%
Backfill queue	0 pending backfill tasks	1–2 tasks pending	≥ 3 tasks or task > 4 hours old

The dashboard is read by the operator at-a-glance before acting on any alert. An all-green dashboard means the operational view is trustworthy. Any yellow or red indicator means the operator should weight alert recommendations accordingly and may want to request manual verification before acting on suggested_fallback recommendations.

6. Failure Mode Cross-Reference

This playbook handles the following failure modes from s1-failure-mode-register:

failure_id	How this playbook addresses it
tool-timeout	Offline-mode operation + exponential backoff reconnection (§2.1)
stale-data	Stale-feed detection §4 downgrades band and escalates
schema-drift-input	Normalization layer quarantines; pipeline monitoring alerts on reject-rate spike (§1.1)
false-success-report	Backfill retroactive-alert step (§3.2) surfaces cases where a missed alert would have been critical
looping-retry	Reconnection uses the bounded retry policy from s1-fallback-architecture-blueprint; offline mode prevents retry storms
silent-degradation	Staleness indicators + watchdog metrics surface gradual quality decay before it silently corrupts operational decisions

7. Worked Example — Partial Outage Recovery (illustrative)

Scenario: 2026-05-29 19:00Z, a communications node serving sw-water-level-j1 and sw-soil-moisture-j2 (both reliability grade B) goes offline. sw-rain-gauge-nwfl-01 (grade A) and sw-tide-noaa-destin (grade A) remain online.

T+0 (outage start): Two heartbeat misses from both B-grade sources. No offline-mode trigger yet (< 50% of active sources offline).

T+2 min: Third miss. Both sources flagged sensor-offline. confidence_band demoted to low. Fusion continues with NOAA rain gauge and NOAA tide as primary contributors. Composite band for water-level drops to medium (sole high-band source is tide; rule 5 applies). A severity: medium C6 emission fires to s1-threshold-escalation-spec.

T+10 min: Connectivity node still unresponsive. Pipeline ops receive the Tier 1 async alert. They confirm a physical network outage and estimate 45-minute repair window.

T+55 min (connectivity restored): Both B-grade sources resume heartbeats. Backfill procedure triggered for the 55-minute gap. 55 minutes × 60 records/min × 2 sources = ~6,600 records fetched, normalized with backfill: true, and injected into historical store. Retroactive alert evaluation finds the water-level would have hit the pond-level-high threshold at T+22 min during a brief rain event. A retroactive-alert event (informational) is logged. No operational action is triggered since the event is historical.

T+56 min: Source band restored to medium after operator re-validation via last_validated update. Dashboard returns to all-green.