

How many sources did you actually have?

Systems degrade quietly. That is the part that hurts.

AT A GLANCE

The problem. Nothing failed loudly. Sources dropped away one at a time, the system kept producing an answer, and the answer looked exactly as confident as it had when everything was working.

What this does. Ties the system's state to a declared count of sources genuinely taking part, and binds the roster of who was present — and who was not — into the record of the decision.

Who it is for. Integrators, safety and systems engineers, and programme buyers who have to state what a system does when it is running on less than it was designed for.

Maturity. Implemented in a company-funded reference implementation and exercised in simulation. **Not yet proven in the field:** no hardware-in-the-loop, no field trial, no installed base.

Status. Covered by US provisional patent application **64/149,078**, filed 6 September 2026. One of four applications filed 2026; the fourth, on consensus over categorical judgements, has its own paper. Patent pending; none examined or granted.

The failure nobody writes a report about

Ask an engineer who has investigated an incident what surprised them, and a version of this often comes back: the system had been running degraded for a while, and the output gave no sign of it. Not a crash, not an alarm — a system doing what it was built to do, on fewer inputs than it was designed around, in the same format and with the same apparent authority as before.

This is the second failure mode, quieter than the first. The first is a source that reports something wrong; this one is a source that **leaves** — permission revoked, a feed stopped, a unit offline, a stream an operating-system update stopped providing, a sensor present but silent for an hour. Any one is ordinary. The question is what the system does next.

Why "graceful degradation" is not the same as knowing

Degrading gracefully is a design virtue, and it is not the issue. The issue is that it can be **silent**: the answer keeps arriving, formatted the same way, driving the same automation, with nothing in it distinguishing "computed from everything we have" from "computed from what was left."

Some systems do expose this. Health telemetry, quality and validity flags, reported covariance and a designed-in minimum-configuration state are established practice. The problem is not universal and we do not claim nobody has addressed it.

What is worth asking of any particular multi-source system is narrower. How many sources were genuinely contributing to *that* answer, and is the number recorded with it or only inferable from separate logs? Does behaviour change when the count drops? And if nothing was received from an expected source, does anything in the record of that decision say so? The last is most often missing, because logs generally record what arrived.

What a participation count does and does not establish

It establishes how many sources took part — a different question from how much independent evidence stood behind the answer. Four feeds from one upstream, four models sharing one corpus, four instruments on one reference: each is four participants and considerably less than four independent pieces of evidence, and counting participants does not reveal that.

The count establishes participation, which is a checkable record of what was contributing. Its implications for evidential strength depend separately on the deployment's source dependencies — engineering work done beforehand, which this records participation against rather than performs.

What this capability does

It counts participation, not installation. The state the system is in depends on how many sources are genuinely taking part at that moment — not how many are fitted, configured or nominally present.

It ties the state change to that count. Falling below what the deployment declared is a change of state, declared in advance, with a defined consequence — not a warning appended to an otherwise unchanged answer.

It records who was there. The roster is bound into the record of the decision: what was contributing, what was expected and absent, carried with the decision rather than reconstructed afterwards from separate logs that may not agree.

An expected source from which nothing was received is a recorded fact, distinguishable from one whose input arrived and was rejected. The record shows that nothing reached the system, not that nothing was transmitted.

An illustrative scenario

This is not a customer story. No deployment described here has taken place.

A platform is designed around four sources its integrator has established to be independent, and its deployment declares that at least three must be participating.

Over one working day, two things happen. During routine maintenance a permission is changed and one feed stops. Separately, a second source stays connected but nothing usable is received from it for forty minutes — present, and not contributing. The platform is now running on two participating sources, still producing output that looks unchanged.

Here the count is part of the state. Falling from four to two crosses the declared boundary of three, and what happens at that boundary was decided before the day began rather than improvised by whoever is on shift. The roster — two participating, one revoked, one silent — is bound into each decision in that interval, so what the system was working with is in the record rather than a later reconstruction.

Why this matters commercially

"The system degrades gracefully" is an assertion about behaviour. "The state is tied to a declared participation count, and the roster is recorded per decision" is a property an assessor can check, which turns *what was it actually running on?* into a question with a recorded answer. That is evidence that can feed a safety argument, not a safety argument in itself, and nothing here has been assessed against any standard.

What this is not

- **Not a monitoring dashboard.** The count participates in the decision rather than sitting alongside the output as telemetry.
- **Not a measure of independence.** It counts participants. Whether they are independent of each other is an assumption of the deployment, and this does not verify it.
- **Not a replacement for redundancy.** A system down to one source is down to one source however well that is recorded.
- **Not a safety certification.** Nothing here has been assessed against DO-178C, ISO 26262, IEC 61508 or any comparable standard.
- **Not yet proven in the field.** Results to date are from simulation. No hardware-in-the-loop, no field trial, no installed base.
- **Not a granted patent.** Four US provisional applications are on file. None has been examined and none is granted.

A first conversation

Bring a nonconfidential example — a case where the system turned out to be running on less than it was designed for, particularly one nobody noticed at the time. We will work through whether this architecture would earn its place on your platform and what a joint evaluation would need to establish. Recorded data, if it ever helps, comes later and under terms you set. A first call is a technical discussion, not a pitch, and we will say when the answer is that this would not have helped.

info@rvtechconsulting.com · RV Technology Consulting, Las Vegas, Nevada

Covered by US provisional patent application 64/149,078, filed 6 September 2026. Four US provisional patent applications filed 2026. Patent pending; none examined or granted. The scenario above is illustrative and describes no actual deployment. This document describes capability only and is not a technical specification.