The decision test

A sensor is a claim about the future: that some decision, some day, will be made differently because this measurement exists. Four gates hold the claim, each with its own stop, and one sentence has to finish before any hardware is priced. Failing on paper is free; failing in a junction box is not.

RECORD

Four gates and the completion sentence — each gate with its answer and the stop it triggers

01 **Which decision?** A named, recurring decision — schedule the bearing renewal, adjust the setpoint, release the batch, dispatch the crew.
Stop: "visibility" and "insight" are not decisions; they are hopes with a subscription fee.

02 **Whose decision?** A named role that will actually receive the data, with the authority to act on it.
Stop: if the answer is "the team", the honest answer is nobody.

03 **What threshold or comparison?** The value, trend or difference that would change the decision.
Stop: if no value would change it, the measurement is decoration.

04 **What is it worth?** Roughly — in avoided cost, recovered capacity or reduced risk — because the chain that delivers the measurement has a cost, and the two numbers have to meet.
Stop: worth less than the chain costs.

05 **The completion sentence.** Written out, in full, before pricing: **"When this value crosses ___, [role] will ___ instead of ___."** If the sentence cannot be finished, the sensor is not an investment; it is a souvenir.

ASK

Four questions that keep a sensing programme a queue rather than a catalogue

– **Is this candidate on the queue, or in front of it?**
A sensing programme is a portfolio of named decisions sorted by value, and the waiting list is public: anyone may promote a candidate by naming the decision, the owner and the value. Coverage is a portfolio, not a percentage.

– **Was it promoted because the sensor is cheap?**
Then it has passed no gate. Cheap sensors still buy engineering time, network capacity, storage and attention — and the expensive links are all downstream of the hardware.

– **Which physical effect carries the earliest useful evidence for *this* decision?**
Start at the decision and walk backwards to the physics. Walked machine-first with a catalogue open, every question defaults to "more, better, everywhere" and the economics quietly drown.

– **What accuracy does *this* decision actually need, and can the location be reached?**
A trend feeding a scheduling decision tolerates offset error that would ruin a custody-transfer measurement. The sensor must also live where the effect is visible, and be wired, powered and maintained there.

THE BOUNDARY THIS CARD CARRIES

A measurement can pass all four gates and still fail later — the sensor may not resolve the effect, the data may arrive too late, the threshold may prove wrong. Those are honest engineering failures. A measurement that fails the gates now has nowhere to fail later, because it was never going to change anything. This card selects no sensor, sets no threshold or accuracy class, and authorises no installation.

The load statement

"It will cope" is not a method. Rate × size × the worst honest minute is — and each of the three inputs is measured rather than guessed, then debited against a headroom figure that already exists in writing. Networks do not fail at their average; they fail at their worst minute.

RECORD

Five measured inputs, then the arithmetic they produce

01 **Points per class.** Analogues, discretes and strings counted separately for the proposed flow, because change behaviour differs by two orders of magnitude between placid temperatures and chattering discretes.

02 **Measured change rate per point class.** From a representative day, not a datasheet and not a guess. For report-by-exception flows this is the rate; for polled flows it is samples per second per point.

03 **Bytes per message on the wire.** Payload plus protocol overhead, from a packet capture. Overhead differs sharply by protocol theory, and measured captures beat datasheet arithmetic every time.

04 **Peaking factor, with its source named.** The worst honest minute over the average — shift-change refreshes, batch transitions, alarm floods — obtained from flood tests and batch edges rather than assumed.

05 **Buffer drain configuration.** What each edge buffer does on reconnect, *on top of* live traffic. The drain rate is a configuration choice this arithmetic must set, not discover during the outage.

06 **The two results.** Steady rate = points × change rate × bytes. Worst minute = steady × peaking, plus the configured drain. Both recomputable from the rows above.

ASK

The numbers that turn arithmetic into a social contract

— **What is the site's written headroom policy?**
Headroom is policy, not leftovers. Every new flow's load statement is debited against a figure that is written down — and a link at 80% of policy headroom is full, whatever its utilisation graph says.

— **What ceiling has the automation owner set inside control zones?**
Obtained *before* design, not discovered at commissioning. The load statement is presented against it, and the after-state — controller scan margins, segment utilisation, worst observed minute — is measured and filed where the next project will find it.

— **What is the latency budget, stage by stage?**
Sampling delay, device processing, per-hop transit, broker or server processing, storage write, consumer refresh. Budgets are additive; the commonest discovery is that the network was innocent and the seconds live in scan intervals and refresh timers at the two ends.

— **Which stage blows the budget?**
The right question once the stages are written down. "The dashboard feels slow" is not a finding.

THE BOUNDARY THIS CARD CARRIES

This card supplies no headroom fraction, no peaking factor and no bytes-per-message figure — those come from your process, your network and your own captures, and a number borrowed from another site is not evidence about yours. Network design, segmentation, and any change to control-zone traffic are engineering activities under site change control and the automation owner's authority.

The timestamp audit

A timestamp is three claims wearing one number: which moment it records, which clock produced it, and in what representation it is stored. Most systems answer all three by accident. One row per device family, and one deliberate interruption test at the end.

RECORD

The four audit columns — what each records, and the answer that should worry you; walk them per device family

AUDIT COLUMN	WHAT THE ROW RECORDS	THE ANSWER THAT SHOULD WORRY YOU
Stamped when	Source-stamped by the device, or arrival-stamped by a poller or historian.	Store-and-forward telemetry arrival-stamped. The gap runs to hours.
Which clock	The device's own, a site time server, or a GPS-disciplined source — and how far it might be from true.	A commissioning laptop's clock, five years stale, on a device used to order events.
Which representation	UTC stored, local rendered, zone named in every export leaving the site.	Naked local times — an ambiguous, repeated hour every autumn.
Drift monitored?	Drift alarms on every source, clock state recorded beside the quality codes.	"Time sync is configured." Nobody watches whether it still works.

ASK

Four design rules, and the test that finds forgeries

- **Stamp at source wherever the device can.**
  Where a poller must stamp instead, record *that fact* in the point's metadata, so consumers know the weakness of the claim.
- **Carry both stamps when they differ.**
  Source time and arrival time answer different questions — when did it happen, when did we know — and buffered telemetry needs both.
- **Never let a conversion re-stamp.**
  A gateway that replaces source time with "now" has forged the birth certificate. Test for it at commissioning; it is common and silent.
- **Buffer a source deliberately, then look at what arrives.**
  Interrupt the link, let the buffer fill, restore it — and see whether the drained values carry their observation times or the moment the link returned.
- **Monitor the clocks like the utility they are.**
  A clock that fails quietly corrupts everything it stamps, plausibly. An investigation should be able to ask "was time healthy?"

THE BOUNDARY THIS CARD CARRIES

Timestamps are the one piece of context that cannot be repaired later, which is why this card is an audit rather than a remedy. It prescribes no synchronisation tier and no accuracy figure: the ordering accuracy each data class needs is a site engineering decision, and changing time sources or device settings is change-controlled work.

The retrieval modes

What comes back from a query is not the data. It is a retrieval mode applied to whatever survived compression — and a consumer who does not know which mode they asked for is being confidently misled.

RECORD

Four modes, what each is honest for, and where each one lies

MODE	WHAT IT RETURNS	HONEST FOR	LIES WHEN
Raw / actual	The stored points, as kept.	Investigations, model training, audit.	You forget that compression already thinned it.
Interpolated	Values computed on a regular grid between stored points.	Aligning tags for calculation and export.	The signal was stepped, or the gap was an outage, not a quiet period.
Aggregate	Average, minimum, maximum or count per window.	Dashboards, KPIs, long-window trends.	The consumer mistakes the average for the excursion; min and max were discarded.
Stepped / last-known	The value held until the next stored point.	Discrete states and report-by-exception data.	Applied to a genuinely continuous signal.

ASK

Three consumer rules, and two fields the tag has to carry

- **Steps as steps.**
  Report-by-exception and discrete data are step functions and must be retrieved stepped. Interpolating them invents transitions that never happened.
- **Gaps as gaps.**
  An outage interpolated into a smooth line is the chain's most dangerous lie. Quality codes and gap markers let retrieval say "unknown here" — the consumer has to ask for them.
- **Investigate on raw — knowing even raw is post-compression.**
  Trends may live on aggregates for speed; investigations, models and arguments with money go back to the stored points.
- **What is this tag's compression setting?**
  Lossless encoding or a configured deviation, recorded in the point's metadata — or it is a rumour.
- **Was the deviation derived, and when was it last examined?**
  From measured noise and named decision sensitivity, per tag class, re-examined when the tag's duty changes. The tag that fed a monthly report tolerates a deviation the same tag feeding a model cannot.

THE BOUNDARY THIS CARD CARRIES

Lossy compression settings are measurement decisions made in advance, not storage decisions — excursions inside the deviation vanish, and the lost detail is unrecoverable. This card names no deviation, ratio or product default, because a setting derived from another plant's noise is not evidence about yours. Historian configuration changes belong to the system owner under site change control.

The data contract

An interface is where meaning dies, and a contract is how it survives. Five clauses gather facts the earlier layers already established and put them at the boundary where they have always been lost — written, versioned, owned on both sides.

RECORD Five clauses, what each gathers, and the defect class it retires

CLAUSE	WHAT IT GATHERS	DEFECT CLASS IT RETIRES
Identity	Which points, named how, resolving to which assets in the model.	Joining the wrong pump.
Semantics	Quantity kind, unit and its single authority, calibrated and plausible range, quality vocabulary and mapping.	Unit assumptions of the Mars Climate Orbiter class.
Shape	Payload schema, encoding, retrieval semantics wherever history is served.	Payload drift; interpolation lies.
Tempo	Update behaviour, step semantics, latency class, timestamp claims — whose clock, which moment.	Step data read as continuous; clock confusion.
Stewardship	The named owner each side, the change process, the notice period for breaking changes.	Silent breaking change.

ASK What keeps a contract from becoming shelfware

- **Is the payload validated at the boundary?**
  Schema validation rejects a malformed message loudly at entry, not three consumers downstream.
- **Does anything check delivered points against the contracted list?**
  Automated point-list checks, on a schedule. Without them the list is an aspiration.
- **Does the payload carry a contract-version field?**
  Both sides can then detect drift as it happens, rather than excavating it afterwards.
- **Has every consumer stated its policy for non-good quality?**
  Exclude, hold last, or degrade its own output flag. A consumer with no stated policy has the policy of pretending everything is fine.
- **Who converts, and does the conversion declare itself?**
  Units have one authority per point, propagated and never re-guessed. Any layer that converts says so; no layer converts twice.

THE BOUNDARY THIS CARD CARRIES

The contract records what a number *is*; it does not establish that any value is correct. Quality propagates and is never laundered — an average of mostly-bad samples is a bad average and must say so. This card sets no units, ranges, quality mappings or notice periods, and authorises no interface, schema or namespace change.

The chain health signals

The pipeline that watches the plant is itself plant. It fails stuck, full, partial, slow, silent, confident and plausible — and defaults to silence while every dashboard keeps rendering, politely, from stale values. Seven signals make those failures announce themselves.

RECORD Seven signals, what each announces, the discipline it closes

HEALTH SIGNAL	WHAT IT ANNOUNCES	DISCIPLINE IT CLOSES
Heartbeats and last-value age	Stuck acquisition; stale values served as fresh.	Acquisition, reporting obligation.
Buffer depth and drain events	Outages absorbed versus data at risk.	Edge buffering, load arithmetic.
Ingest lag and queue depth	Stores failing slowly, unwatched.	Storage and retention design.
Clock offset per source	Time failing plausibly, corrupting everything downstream.	Timestamp discipline.
Schema violations, quality-code rates	Context drifting; quality laundered en route.	Data contracts, master data.
Certificate days-to-expiry	Trust lapsing on a schedule anyone could have read.	Hardening the data path.
Staleness banners at consumers	Degradation declared, not hidden.	Contract tempo clause, displays.

ASK The design questions behind the panel — and the field that closes it

- **Where does consequence earn redundancy?**
  Fleet-feeding hubs and compliance-critical flows justify redundant brokers, clustered stores and dual paths. The convenience dashboard does not.
- **Are the buffers sized honestly?**
  Sized honestly, buffering turns most transport failures into latency rather than loss — the reason buffer health leads the list.
- **Does each consumer declare its own degradation?**
  A dashboard that renders stale data *and says so* is an instrument; one that does not is a hazard.
- **Are the health signals published into the chain itself?**
  Same namespace under a system branch, same store, same displays, same notification discipline.
- **Who answers when one goes amber?**
  A named chain operator with the runbook and the authority to act. The last field on this card is a person's name.

THE BOUNDARY THIS CARD CARRIES

Chain failures compose — a clock fault corrupts everything downstream of it — and they default to silence, which is why the announcement has to be engineered rather than hoped for. This card lists signals to publish; it sets no thresholds, availability targets or on-call obligation, and authorises no change to a system in service.