Condition monitoring as an early-warning trigger, not a guarantee

WHITEPAPER · CONDITION MONITORING

Why reading an asset's real signatures gives the maintenance team a better trigger to act on — and why that is a more honest claim than "we predict failure."

Abstract. The pitch around condition monitoring usually overreaches: it promises to *prevent* failures, to *eliminate* downtime, to *predict* what will break. None of that survives contact with a real floor. What condition monitoring actually delivers is narrower and more useful — a better *trigger*. It reads the signatures an asset already emits — vibration spectra on rotating equipment, particle contamination and water saturation on hydraulic and lubrication oil — and raises a flag when one of those signatures crosses a threshold the maintenance team defined. The team still owns the decision, the discipline, and the work order. This paper argues that the honest framing — early-warning trigger, not guarantee — is also the more defensible engineering position.

The problem: a schedule is not a condition

Most maintenance programs run somewhere between two patterns, and neither one looks at the asset. **Reactive** maintenance waits for the bearing to seize or the hydraulic pump to fail, then responds. **Calendar-based** maintenance swaps the oil filter and services the bearing at fixed intervals — whether the oil is degraded or not, whether the bearing is worn or not. Both patterns work. Neither is reading the asset's real condition.

The cost of acting on a schedule instead of a condition runs both ways. Service too early and you spend labor and parts on a component that was fine. Service too late and the schedule's next interval arrives after the failure it was meant to prevent. A calendar is a proxy for wear, and proxies drift: duty cycles change, loads change, contamination arrives off-schedule. The interval that was right last year is guesswork this year. What the team is missing is not more discipline — it is a signal that comes from the asset itself.

The idea: a real signature is a better trigger

Condition monitoring replaces the proxy with the real thing. Rotating equipment emits a vibration signature; as a bearing degrades or a shaft falls out of alignment, that signature changes in ways that are measurable before the failure event. Hydraulic and lubrication oil carries a contamination signature — particle counts, water saturation — that worsens as filtration fails or seals let water in. These are not predictions. They are the present condition of the asset, read directly.

The pivotal design choice is what you do with that reading. The signature does not declare a failure; it crosses a **threshold the maintenance team defines**. The team sets the limit per asset, informed by its own reliability judgment and the equipment's failure modes. The platform watches continuously and raises a flag the moment a signature crosses. That is the whole mechanism: ongoing measurement against a human-set limit, on a schedule set per machine class. *Same maintenance team. Same workflows. A better trigger.*

Why "trigger" is the honest word — and the right one

Calling this a trigger rather than a guarantee is not modesty for its own sake. It is an accurate description of what threshold-based detection can and cannot do.

- **It is an early warning on the failure modes the signature reveals.** A vibration spectrum surfaces bearing wear, imbalance, misalignment, looseness, and cracks; an oil-contamination reading surfaces particle ingress, water saturation, and oil degradation. It catches what those signatures expose — and makes no claim about a failure mode that emits no signature the instrument is watching. Honesty about the boundary is what makes the in-boundary claim credible.

- **It is detection, not prophecy.** The flag fires because a measured signature crossed a limit *now* — not because a model forecast a future. "The asset is telling you this bearing is degrading, against the threshold you set" is a claim you can audit. "Our AI predicts what will fail" is a claim a reliability engineer will rightly distrust, because it hides the mechanism.
- **The decision stays with the team.** A trigger improves the quality of the moment the team chooses to act. It does not replace the team's reliability engineering — the failure-mode hypotheses, the threshold semantics, the root-cause interpretation — and it does not replace the CMMS that owns the work order. It feeds them a better-timed signal.

How Elpis does it

Two instruments read the two signatures. **VAS** (Vibration Analyser System) runs on the **mDAQ** acquisition platform, mounted at rotating equipment — pumps, motors, gearboxes, fans, compressors, conveyors, structural components — and captures vibration signatures — with measurement schedule and analytics (FFT, order analysis, spectrum, failure-mode mapping) configured per machine class — that turn a waveform into a recognizable signature. **E-IDOS** measures hydraulic and lubrication oil-health — solid particle contamination, water saturation, oil flow — and logs results to the **ISO 4406 / NAS 1638** oil-cleanliness report codes that maintenance teams already read. E-IDOS is a **standalone, sensor-agnostic appliance**: it works with HYDAC, Parker, MP Filter, and Argo-Hytos sensors, so the team isn't locked to one sensor vendor's contamination input.

In every case the threshold is the team's to set, and the trigger fires against it. On the rotating-equipment side, VAS signals feed EREMOS V2 today, where a threshold-crossing becomes a persistent alarm and an incident workflow that survives shift handoffs. On the oil-health side, E-IDOS operates today as a standalone reliability instrument — on-board HMI, thermal printer, BLE Android app, email reports — so the technician reads condition data and acts locally. *Streaming E-IDOS into EREMOS V2 is near-term roadmap, not current behavior.* For the capability detail see **/capabilities/condition-monitoring**; for the outcome narrative see **/solutions/predictive-maintenance**.

What this is *not*

- **Not a guarantee, and not failure-prevention.** It is an early-warning trigger on the failure modes its signatures reveal. It does not promise zero downtime, does not catch every failure, and does not prevent failures — it gives the team a better, earlier moment to act.
- **Not "AI predicts what will fail."** Detection is threshold-based on real, present condition signatures, not a machine-learning forecast. Where AI appears in the platform it is decision-support, outside the deterministic data path — it never raises the trigger.
- **Not a replacement for the team or its tools.** The CMMS stays the system of record for work orders, parts, scheduling, and labor; the reliability function still owns the engineering judgment. The platform improves trigger quality and workflow; the discipline remains the team's.
- **Not a certification.** ISO 4406 and NAS 1638 are the oil-cleanliness report codes E-IDOS logs against — a shared vocabulary for reporting condition, not an Elpis credential.

Takeaways

1. A schedule is a proxy for wear; a real signature is the condition itself — and a far better trigger to act on.
2. The mechanism is honest and auditable: continuous measurement against a **threshold the maintenance team defines**, not a model's forecast.
3. The trigger is an early warning on the failure modes its signatures reveal — explicitly not a guarantee, not failure-prevention, not "AI predicts failure."
4. VAS reads vibration on rotating equipment (on mDAQ); E-IDOS reads hydraulic/lubrication oil-health (standalone, sensor-agnostic, ISO 4406 / NAS 1638). The team, its CMMS, and its reliability discipline stay theirs.
