MOMENTEDGE WHITE PAPER | SEPTEMBER 2026

From Robot Data to Operational Understanding

Real-world robot fleets generate continuous telemetry, logs, video, and sensor data. The challenge is turning that growing record of system behavior into evidence engineers can use to find failures, investigate incidents, and improve performance.

As robot deployments grow in size and duration, teams accumulate a larger record of real-world system behavior. The engineering problem is increasingly what to retain, what deserves attention, how to reconstruct incidents, and how to turn field evidence into changes in the system

4.66M industrial robots in operational use worldwide in 2024 [1]	**+11%** growth in U.S. industrial robot installations in 2025 [2]	**$2.3M/hr** reported cost of an unproductive hour in automotive manufacturing [3]

EXECUTIVE SUMMARY

Robot data increasingly has to serve two jobs: explain what happened in the field and inform what changes next.

The number of robots operating in the physical world is rising quickly. The International Federation of Robotics counted 4.66 million industrial robots in operation in 2024, up 9% year over year; annual installations have exceeded 500,000 units for four consecutive years. Professional service robots are also expanding, with more than 199,000 units sold in 2024. [1][4]

Every deployed system creates operational evidence: telemetry, logs, video, perception outputs, sensor streams, task state, environmental context, and human interventions. Yet the useful evidence is usually sparse and the questions are not static. Some incidents are known in advance and can be captured efficiently with triggers. Others emerge only after robots encounter new environments, software states, edge cases, or combinations of conditions that nobody thought to instrument.

That creates a structural gap between data collection and data use. Teams can capture targeted data, record everything they can afford to retain, or inspect incidents after an operator flags them. Each approach is useful, but none by itself solves the full workflow from capturing an event to finding it later, understanding what happened, and using the result in engineering.

Robotics data workflows increasingly span four connected stages: targeted collection, fleet-wide event discovery, multimodal investigation, and feedback into engineering. Known conditions can drive precise data capture. Broader fleet data can reveal unexpected events. Relevant signals can then be assembled for investigation, and validated findings can inform future collection, operations, evaluation, simulation, and training.

The goal is not to replace robotics observability, storage, visualization, or simulation systems. It is to connect them around a higher-level question: how does a robotics team continuously turn deployment experience into evidence that can inform operations and engineering, without requiring enormous amounts of manual inspection?

01 | WHY THIS BECOMES A FLEET-SCALE PROBLEM

More deployed robots create more operational evidence

Robotics is becoming a form of physical AI: software is continuously perceiving, deciding, and acting in the physical world.

That changes what a “failure” looks like. It may be a software exception, but it may also be a perception miss, localization drift, degraded task performance, an unexpected interaction with the environment, or a task that technically succeeds while falling short of operational expectations.

The evidence behind those events is often distributed across modalities. A log may show that a subsystem entered an error state, while video reveals the physical interaction, localization traces expose the precursor, and task metadata shows the operational impact.

In physical AI systems, understanding what happened often depends on reconstructing that full context rather than reading any single signal in isolation.

Known events can be instrumented, but real-world deployments keep producing unknown ones.

Triggers remain one of the most efficient ways to collect targeted data. Momentedge Clipper is one example of this approach, an open-source ROS 2/Rust component for trigger-based data collection. When a team knows the failure condition, the relevant signals, and the time window it needs, event-driven capture can preserve high-value context without retaining every high-bandwidth stream continuously.

But triggers encode existing knowledge. They are strongest for events a team has already defined. As deployments expand, the set of meaningful conditions changes: robots enter new facilities, operators use them differently, payloads and layouts vary, software updates introduce new interactions, and weak signals may only become visible when they recur across a fleet. Collection logic cannot anticipate every future question.

In our interviews, many teams described a hybrid approach. They want highly targeted, trigger-based collection for known conditions, while also retaining broader, lower-fidelity data as a fallback.

That second category is often collected with no immediate use case in mind; effectively, “keep it in case we need it later.” In practice, this can mean large volumes of raw data are moved to cloud storage without a clear plan for how they will eventually be searched, interpreted, or used.

Manual investigation becomes harder to sustain at fleet scale

When an event is not cleanly instrumented, engineering teams often reconstruct it manually: find the relevant recording, line up timestamps, search logs, inspect video or sensor streams, compare the incident with normal operation, and determine which signals deserve deeper analysis. This work is intellectually valuable, but much of the search and correlation is repetitive.

At a small scale, this is manageable. At fleet scale, it competes with feature development, reliability work, customer support, and model improvement. The cost is not only engineer hours. Some issues are investigated late or not at all, recurring problems remain unresolved longer, and useful field examples never reach the teams building tests, simulations, or training datasets.

The operational stakes can be material

The exact cost of a robot incident varies enormously by application, and not every robot problem causes production downtime. Still, industrial benchmarks show why faster detection and diagnosis matter. Siemens' 2024 downtime study estimated that unplanned downtime consumed 11% of revenue across the 500 largest companies represented in its analysis, and placed the cost of an unproductive hour in automotive manufacturing at $2.3 million. [3] These figures should be treated as large-enterprise benchmarks rather than a universal robot ROI model, but they illustrate the value of shortening the path from abnormal behavior to action.

542K	4.66M	38K
industrial robots installed globally in 2024 [1]	industrial robots operating worldwide [1]	U.S. industrial robot installations in 2025 [2]

02 | THE MISSING DATA LOOP

Robotics needs collection and analysis to inform each other

Most robotics data workflows are organized around individual stages: record data, upload it, visualize it, debug an incident, create an evaluation set, generate synthetic data, or retrain a model. Those capabilities are important, but they increasingly depend on information flowing between stages.

A fleet produces a stream of evidence about where the system struggles. Investigation turns some of that evidence into knowledge: which signals were predictive, which conditions were relevant, which behaviors were actually harmful, and which examples should be retained. That knowledge should influence future collection and prioritization. Over time, those findings can make future collection more selective and future analysis more targeted.

COLLECT	SURFACE	INVESTIGATE	IMPROVE
Preserve targeted, high-value context at the edge when the event is known.	Identify incidents, degradation, anomalies, and unusual behavior across broader fleet data.	Assemble the cross-modal evidence engineers need to understand behavior and narrow root cause.	Feed validated findings into operations, evaluation, simulation, model improvement, and future collection.

1. Collect: be selective when you can

For known conditions, the best data architecture may be deliberately selective. Rolling buffers, event-driven recording, local filtering, and configurable upload policies can retain the seconds or minutes around a meaningful event while controlling bandwidth and storage. The objective is selective collection: preserve enough context to answer the question without assuming every byte has equal future value.

2. Surface: make the unknown searchable

Selective capture cannot solve events that were never defined. Fleet analysis provides a complementary path: historical or continuously uploaded data can be analyzed to prioritize periods that deserve attention and surface events or patterns that were not explicitly instrumented and might otherwise require manual discovery. "Surfacing" is broader than anomaly detection; it can include explicit failures, degraded performance, repeated operational friction, or behavior that is unusual relative to a task or environment.

3. Investigate: turn an event into evidence

Incident detection is only useful if the team can move quickly into investigation. The core workflow is contextual assembly: retrieve the relevant telemetry, logs, video, sensor data, task state, and surrounding time window; align them; and give engineers a coherent starting point. Root-cause analysis is therefore an important use case, but it is not necessarily a single automated answer. In many robotics systems, the practical objective is to reduce the search space and shorten the path to a defensible engineering conclusion.

4. Improve: return field experience to development

Once an incident has been understood, the data can have a second life. High-value examples can become regression tests, evaluation sets, failure taxonomies, simulation scenarios, or training inputs. This matters as physical AI increasingly combines real-world and synthetic data. NVIDIA describes modern robot-learning workflows as drawing on data from real robots, synthetic environments, and other sources, while simulation is used to train and validate policies and generate diverse synthetic datasets. [5][6] The implication is not that real-world data replaces simulation; it is that the most useful simulation and training loops need a disciplined way to learn from what actually happens in deployment.

> **The useful output is therefore not raw data volume, but an increasingly structured relationship between real-world events, the evidence that explains them, and the engineering actions that follow.**

Connecting data capture with engineering workflows

Robotics teams already have important infrastructure: ROS and ROS 2 recording, MCAP or rosbag storage, cloud object stores, observability systems, visualization tools, databases, simulation environments, and internal analytics. A useful new layer should not require replacing all of them. It should make the data moving through those systems easier to prioritize and act on.

A working system needs five capabilities

Multimodal ingestion and alignment. Robotic behavior rarely makes sense from one stream. The system needs to ingest or reference telemetry, logs, ROS topics, video, perception outputs, sensor streams, and operational metadata while preserving timing relationships.

Event and segment identification. Instead of treating a recording as the unit of work, the system should identify meaningful segments: failures, anomalies, degraded behavior, rare conditions, or operator-defined events. The output is a prioritized set of moments with enough surrounding context for inspection.

Investigation context. For each event, engineers need the evidence assembled around it: what changed before the incident, what the system believed, what the physical environment showed, and what happened afterward. This is where data processing becomes operationally useful and not merely searchable.

Human judgment in the loop. Not every anomaly is a failure, and not every important event can be interpreted from data alone. Engineers and operators provide the domain context needed to confirm significance, distinguish expected behavior from real problems, and determine what action should follow.

Feedback to collection and development. Confirmed incidents and valuable examples should update what teams retain, which triggers they define, which patterns they search for, and which examples flow into tests, evaluation, simulation, or training.

Edge and cloud are complementary, not competing architectures

Some robotics data should be handled close to the robot. High-bandwidth streams, constrained connectivity, privacy requirements, and the need for pre-event context all favor local buffering and selective capture. Other questions require a fleet-wide view: comparing incidents across deployments, searching historical data, identifying recurring patterns, and learning which events are important enough to influence future collection.

This suggests a hybrid architecture. The edge protects context and controls data movement. Fleet-side analysis creates visibility across time and deployments. Knowledge flows back toward the edge as teams learn what deserves to be captured or retained more aggressively.

Momentedge Clipper demonstrates the known-event side of the loop

Momentedge Clipper is an open-source ROS 2/Rust collection component designed around event-triggered recording. It preserves the context immediately before and after a trigger in MCAP, making it possible to capture targeted evidence when a team already knows what condition matters.

Clipper represents the known-event side of the workflow. Broader fleet analysis addresses the complementary case: events that were not predefined. Once those events are understood, the findings can inform future triggers, retention policies, evaluation sets, and development workflows.

04 | ENGINEERING USES AND LONGER-TERM IMPLICATIONS

The same data workflow supports several engineering tasks

Fleet data can be used to identify incidents that were not explicitly flagged, reconstruct what happened around a known event, accelerate root-cause investigation, and extract useful examples for evaluation or training.

These tasks share a common requirement: isolate a meaningful slice of real-world robot behavior and assemble enough surrounding context to interpret it.

What changes as field evidence accumulates

- Faster incident response: engineers start with a prioritized event and assembled evidence rather than a blank search across raw recordings.
- Fleet-level pattern discovery: repeated weak signals or low-frequency issues can become visible across robots, sites, software versions, or operating conditions.
- Adaptive collection: confirmed incidents can inform new triggers, retention rules, or capture policies at the edge.
- Structured evaluation data: real failures and edge cases can become regression and evaluation sets rather than disappearing into archives.
- A tighter sim/real feedback loop: field examples can help teams decide which scenarios to reproduce, augment, or stress-test in simulation and synthetic-data pipelines.

Over time, teams also build structured knowledge about their systems: which behaviors matter, which signals help explain them, what context engineers use to validate an incident, and which actions follow from different outcomes.

That accumulated knowledge can improve both collection and analysis. Repeated incidents can inform new triggers, confirmed patterns can help prioritize future events, and validated examples can flow into testing, evaluation, simulation, or training.

The system becomes more useful as it gets better at connecting field behavior to engineering decisions, rather than simply accumulating more raw data.

SOURCES AND NOTES

Sources

[1] International Federation of Robotics. World Robotics 2025: Industrial Robots (2025). https://ifr.org/worldrobotics/report-2025

[2] International Federation of Robotics. US Robot Industry Returns to Double Digit Growth (2026). https://ifr.org/ifr-press-releases/news/us-robot-industry-returns-to-double-digit-growth

[3] Siemens / Senseye. The True Cost of Downtime 2024 (2024). https://assets.new.siemens.com/siemens/assets/api/uuid:1b43afb5-2d07-47f7-9eb7-893fe7d0bc59/TCOD-2024_original.pdf

[4] International Federation of Robotics. World Robotics: Service Robots See Global Growth Boom (2025). https://ifr.org/

[5] NVIDIA. Robotics Simulation (2026). https://www.nvidia.com/en-us/use-cases/robotics-simulation/

[6] NVIDIA. Robot Learning in Simulation (2026). https://www.nvidia.com/en-us/use-cases/robot-learning/

Note: Industry figures are included to establish the scale and direction of robotics deployment and the economic importance of operational reliability. They do not imply that all industrial downtime is caused by robots or that Momentedge would eliminate the cited costs. Product-direction statements in this paper describe Momentedge's thesis and intended architecture, not guaranteed customer outcomes.