The AIOps Data Pipeline

Transforming Diverse Telemetry into Reliable Unified Data

White Paper

AAC

TABLE OF CONTENTS

The AIOps Data Pipeline: Transforming Diverse Telemetry into Reliable Unified Data

1. Introduction

In Part 1 of this series, we outlined how AIOps transforms IT operations from reactive firefighting to proactive incident prevention by applying machine learning and analytics across the full spectrum of IT operations telemetry data. Part 2 builds on that foundation and focuses on the AIOps data pipeline as the core system that connects information across IT domains. This data pipeline transforms diverse telemetry into reliable, unified data while providing the foundation for analytics, automation, and strategic foresight.

The following sections examine each stage of this pipeline, from data ingestion and centralization to transformation, quality assurance, and feature engineering, showing how raw operational signals become trusted intelligence that drives modern IT operations. Although many of these capabilities are now available out of the box through modern AIOps and observability platforms, practitioners must understand how these functions work, how to fine-tune them, and how to align them with organizational priorities and governance requirements.

2. Essential Data Types for AIOps

In the previous paper, we highlighted possible data-generating sources, including cloud applications, mobile devices, storage systems, laptops, servers, IoT endpoints, microservices, ITSM platforms, and SIEM solutions, to illustrate the breadth of today's IT ecosystem. This section moves the discussion from data sources to the kinds of data they produce, providing a high-level overview of the telemetry and records that feed AIOps pipelines (see **Figure 1**). Together, these streams supply the raw material for analytics, correlation, and automation, supporting functions such as anomaly detection, root-cause analysis, and predictive intelligence.

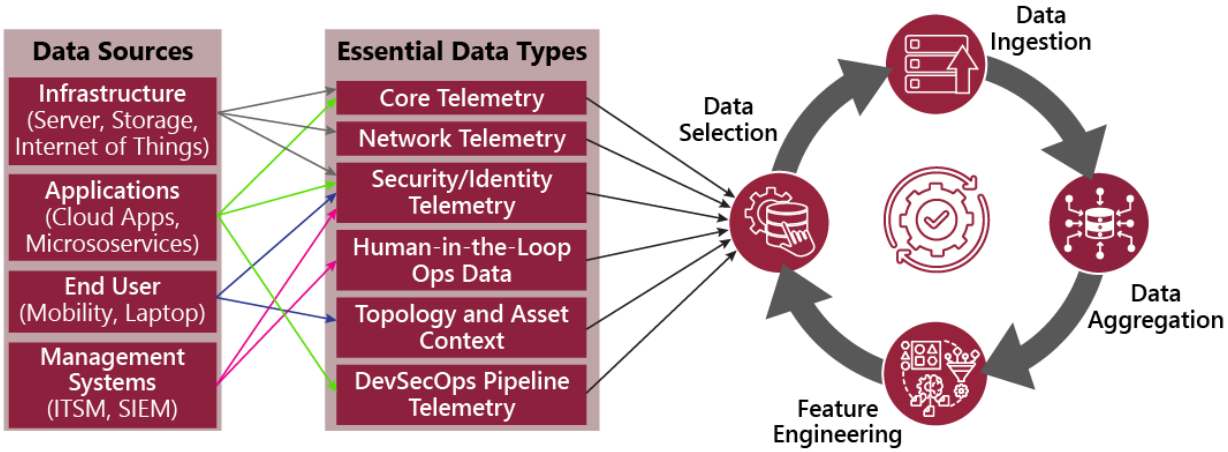

Figure 1: Essential Data Types

Core data categories include logs, metrics, traces, topology data, DevSecOps pipeline telemetry, and security feeds. In practice, organizations can also draw on additional inputs

such as hardware sensors, virtualization metrics, cloud provider service health dashboards, vendor advisories, and industry threat-intelligence reports. These supplemental sources extend visibility beyond enterprise boundaries and strengthen both operational resilience and cyber awareness.

3. Data Ingestion Patterns

After identifying data sources, organizations must reliably collect and transport telemetry into the AIOps pipeline. This challenge requires solutions that capture information in real time, normalize it, and stream it to storage or analytics systems. These solutions broadly fall into several categories: agent-based collectors, monitoring systems, streaming and messaging frameworks, and cloud-native streaming services, each with distinct strengths, limitations, and use cases. Together, they form the data plumbing of AIOps, ensuring logs, metrics, events, and traces flow continuously and at scale into downstream analytics. **Figure 2** depicts the most commonly used data ingestion patterns.

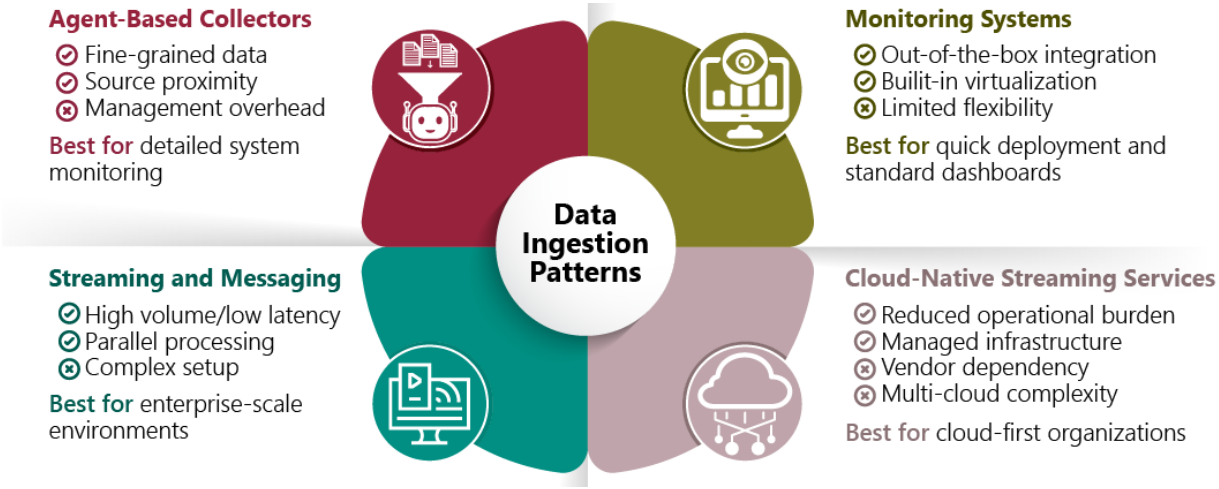

Figure 2: Data Ingestion Patterns

4. Centralizing Data for Easy Access

Data ingestion moves information from source systems into durable storage, where organizations can later transform and analyze it. At this point, structure remains secondary: data typically stays in its raw form (logs, events, blobs) until shaped further downstream. Many organizations employ multiple repository types, using data lakes to hold varied telemetry in its native format and data warehouses for cleaned, organized data used in reporting and analytics.

While many platforms offer open-source or low-cost options (e.g., PostgreSQL, MongoDB, Cassandra, Apache Iceberg), the real challenge lies in storage and transport economics. As telemetry grows to terabytes or petabytes, storage bills escalate rapidly. In cloud

environments, ingress and egress charges become hidden cost drivers. Common I/O optimizations in analytics applications such as AIOps can generate unexpected costs, ranging from thousands to millions of dollars annually, due to storage API calls.

For this reason, effective centralization strategies balance repository choice with strict retention and tiering policies, keeping only the most valuable data "hot" while archiving or pruning the rest.

5. Data Aggregation – Transforming Data into a Unified Format

Data collected from diverse IT and security systems rarely arrives in a uniform or analysis-ready state. Metrics tend to be structured, logs are semi-structured, tickets often remain unstructured, and events may stream in real time at high velocity. For analytics and automation to work properly, organizations must align incoming data into formats that other systems can understand and use. Skipping this step results in inconsistent, incomplete, and far less reliable AIOps outputs.

Shown in **Figure 3**, unifying data typically involves several key stages that address specific challenges in preparing raw telemetry for analysis. These include normalization, cleaning, parsing, aggregation, filtering, and enrichment.

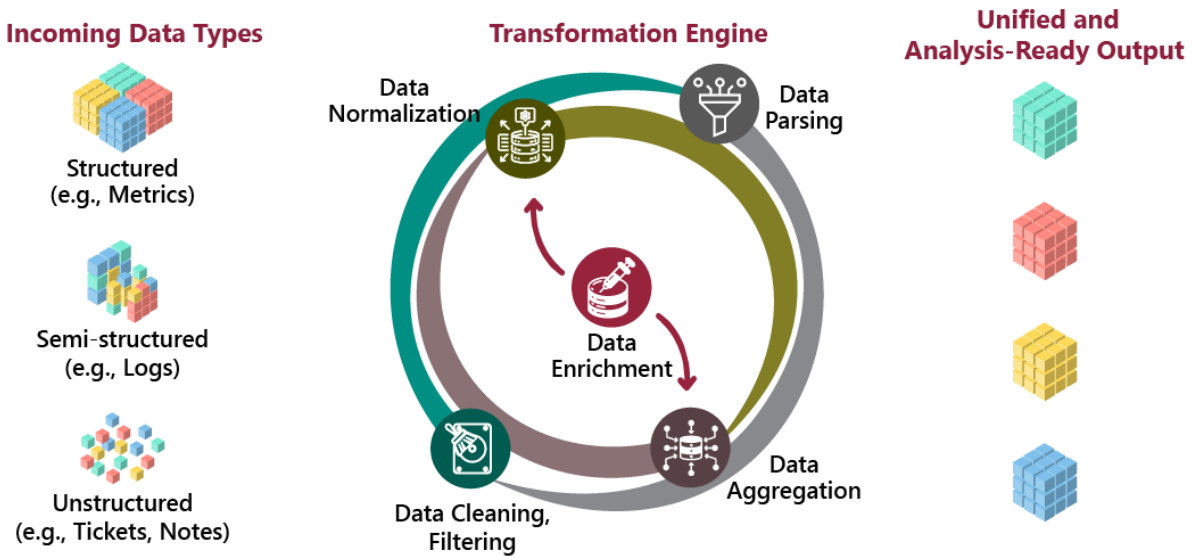

Figure 3: Data Aggregation as a Key Element in Transforming and Enriching Data

Table 1 (next page) summarizes these transformation stages, outlining their purpose and how they contribute to creating consistent, high-quality datasets for machine learning and operational analytics.

The AIOps Data Pipeline: Transforming Diverse Telemetry into Reliable Unified Data

Process	Description
Normalization	Normalization aligns data fields across sources by applying standard schemas. Frameworks such as OpenTelemetry and Open Cybersecurity Schema Framework (OCSF) for security provide widely adopted templates, while lightweight converters (JSON, CSV, etc.) handle generic formats. This ensures consistent interpretation of data—such as latency reported in milliseconds by one system and in seconds by another.
Cleaning	Cleaning removes duplicates, corrects malformed entries, and standardizes encodings or timestamps. Platforms like Logstash, Fluentd, and Apache NiFi include pre-built processors that handle these tasks automatically, reducing the risk of false anomalies caused by data errors.
Parsing	Parsing extracts structured fields such as timestamps, user IDs, or error codes from semi-structured logs. Common solutions include Logstash Grok patterns, Filebeat modules, and OpenTelemetry log processors, all of which provide ready-to-use parsers for popular systems. Parsing transforms free-text logs into searchable, filterable, and correlatable events.
Aggregation	Aggregation transforms raw records into higher-level indicators such as uptime percentages, latency distributions, or error counts per service. Tools like Prometheus recording rules, Elasticsearch aggregations, and SQL queries in modern data warehouses highlight trends and reduce noise for faster decision-making.
Filtering	Filtering eliminates low-value or redundant data, such as verbose debug logs or routine heartbeat events. Using built-in filters in Fluent Bit, Filebeat, or Kafka stream processors, organizations reduce storage and processing costs while maintaining focus on high-value signals.
Enrichment	Enrichment adds missing context such as user identifiers, asset tags, or topology relationships to make events actionable. Integrations with ServiceNow/CMDB, threat-intelligence feeds, or OpenTelemetry attributes provide this context with minimal customization.

Table 1: Data Aggregation and Transformation Stages

6. Data Selection – Ensuring Data Quality

Ingesting data without quality controls overwhelms AIOps platforms with noise, inconsistencies, and blind spots. Reliable insights depend on disciplined data hygiene that ensures accurate, complete, and consistent information enters the pipeline:

- **Accuracy** – Values such as CPU usage or response time must reflect actual system performance
- **Completeness** – Organizations must capture logs, metrics, and traces without gaps that could distort the operational picture
- **Consistency** – Timestamps, formats, and measurement units must align across sources to enable correct data correlation

Practical safeguards include filtering out corrupted records, aligning units and timestamps, and checking for missing or duplicate entries before data advances downstream. These

measures prevent systems from chasing false anomalies or overlooking genuine issues, ensuring analytics and automation rest on a solid, trustworthy foundation.

6.1 Handling Noisy or Missing Data

IT data rarely arrives in pristine condition. Metrics fluctuate, logs contain errors, and entire data streams occasionally fail. Left unaddressed, these issues mislead AIOps systems, creating false positives or masking genuine anomalies. Maintaining data quality demands continuous monitoring for accuracy, completeness, and consistency across all sources.

In practice, organizations implement several safeguards when managing noisy or incomplete data:

- **Noise Management** – Organizations filter or smooth unstable readings so anomalies reflect real system conditions rather than random fluctuations
- **Gap Detection** – Teams identify when data stops flowing or when records disappear from expected intervals
- **Recovery Strategies** – Systems automatically reconstruct incomplete datasets where possible or flag gaps for investigation when they cannot recover data

These measures ensure that insights derived from AIOps rely on reliable, representative information rather than statistical noise.

6.1.1 Data Quality Management Framework

To consistently address noise, gaps, and inconsistencies in operational data, AIOps implementations require a practical data quality management framework. This framework establishes guardrails across the ingestion pipeline, ensuring analytics and automation are using reliable inputs rather than transient errors or missing information. Key elements of this framework include:

- **Defining Data Quality Baselines** – Establish normal behavior patterns for each data source to quickly recognize when metrics drift or behave unexpectedly
- **Integrating Automated Validation** – Implement simple checks at collection points to flag incomplete records, broken formats, or delayed data arrivals
- **Monitoring Collection Health Dashboards** – Track ingestion activity and source connectivity to ensure smooth pipeline operation and proper data delivery
- **Reviewing Normalization and Enrichment Rules** – Periodically revisit transformation logic as systems change to confirm correct field mapping and ensure enriched data reflects current reality
- **Documenting and Governing** – Maintain clear records of data source ownership, issue resolution procedures, and quality standards that data must meet before use

7. Feature Engineering

Feature engineering transforms operational data into structured, model-ready variables that represent system performance, workload behavior, and operational events in measurable form. In AIOps, this critical step converts logs, metrics, and traces into quantitative features that machine learning models can analyze effectively. Well-engineered features expose meaningful operational patterns, reduce noise, and standardize diverse data types, enabling more accurate anomaly detection, faster root-cause analysis, and clearer, more actionable insights across complex environments. After creating these features, teams must identify which data attributes provide the most analytical value and should be retained for modeling.

7.1 Techniques for Feature Selection

Feature selection identifies which variables carry the most useful information for analysis while reducing noise and redundancy. Key approaches include:

- **Expert-Guided Selection** – Engineers apply operational or domain knowledge to retain data elements meaningful for diagnosing or predicting behavior
- **Statistical Evaluation** – Analysts use quantitative measures to assess how strongly each variable correlates with, or explains, outcomes of interest
- **Data-Driven Weighting** – Teams examine correlations and response patterns within the data to determine which variables contribute most to overall variation or system changes
- **Iterative Refinement** – Data scientists test different variable subsets to balance model accuracy, interpretability, and computational cost

7.2 Dimensionality Reduction

Even after careful feature selection, datasets often remain vast and complex, sometimes containing hundreds or thousands of variables. High dimensionality increases processing costs, complicates visualization, and obscures meaningful signals within background noise. Reducing dimensionality addresses these challenges by compressing data into a smaller set of representative attributes that still capture key patterns and relationships.

In practice, this process takes many forms, from summarizing related measurements into composite indicators to applying statistical or algorithmic techniques that identify the underlying structure of large datasets. The goal remains consistent: retain what matters most for analysis while minimizing redundancy and computational overhead. Simplified representations make modeling faster and more efficient while improving interpretability, enabling analysts and automated systems to focus on trends that genuinely influence performance or risk.

The AIOps Data Pipeline: Transforming Diverse Telemetry into Reliable Unified Data

Principle	Implementation
Start Simple	Group similar metrics together, combining several related latency measures into a single view before implementing more sophisticated statistical techniques.
Monitor Data Growth	Track the number of active variables in your datasets; rapid expansion often signals redundant or low-value data collection.
Periodically Review Features	Regularly evaluate your metrics and eliminate those that no longer help you understand or improve the environment.
Balance Interpretability with Efficiency	Select simplification techniques that preserve transparency for operational teams while reducing computational burden.

Table 1: Rules of Thumb for Managing Data Dimensionality

8. The Role of Generative AI and Retrieval-Augmented Generation in Operational Intelligence

Generative AI is reshaping AIOps by fundamentally altering how operations teams engage with expanding data volumes. **Table 2** highlights several techniques and benefits to boosting AIOps using generative AI and retrieval-augmented generation (RAG).

Technique	Benefit
Semantic Log Interpretation	Understands log data based on meaning rather than just keywords or patterns
Faster Incident Triage	Accelerates problem identification and resolution by quickly extracting relevant information
Natural-Language Querying of Operational Systems	Enables operators to interact with operational systems using plain, everyday language
Automated Summarization	Generates concise overviews of complex incidents and their resolution paths
Historical Incident Retrieval	Quickly identifies and surfaces similar past issues to inform current troubleshooting
Knowledge-Driven Correlation	Leverages existing organizational knowledge to identify relationships between events

Table 2: Generative AI and RAG Transform Operational Insights

Generative AI focuses its power on the vast, complex pool of unstructured and semi-structured operational data, including logs, alerts, tickets, knowledge articles, and historical incident records. Language models can read through raw log entries, extract important details,

summarize long incident threads, and relate new alerts to previously encountered issues. Converting these various sources of unstructured text into clear, searchable information enables teams to triage incidents and identify root causes more quickly.

Retrieval-augmented generation enhances this analytical strength. When users pose queries, RAG retrieves and uses only the most pertinent information, such as specific logs, runbooks, configuration snippets, or details from prior incidents. This crucial function keeps both human operators and automated systems focused on precise contextual information, eliminating the burden of manually shifting through millions of log lines or documents.

9. Conclusion

The success of any AIOps effort begins long before algorithms run. It depends on how well organizations gather, clean, and connect data. When teams choose meaningful data sources, design solid ingestion processes, and unite telemetry under a common architecture, they create conditions for accurate analytics and dependable automation. Without this backbone, even the most sophisticated machine learning models succumb to blind spots, false positives, and missed opportunities.

In the following papers, we will build on this foundation to explore the higher-order capabilities of AIOps:

- Event Correlation
- Predictive Intelligence and Root-Cause Analysis
- Continuous Improvement and Model Evolution
- Building a Lean AIOps Stack with Open-Source Tools