Compute-Sharded Stream Processing for Petabyte-Scale Real-Time Cybersecurity Analytics

Conf42 Site Reliability Engineering (SRE) 2026

REAL-TIME ANALYTICS | PETABYTE SCALE | CYBERSECURITY

SPEAKER

Abhishek Suman

Senior Software Engineer
Microsoft Corporation

Abhishek works on large-scale distributed systems at Microsoft, with deep focus on real-time data pipelines, stream processing architectures, and cybersecurity analytics infrastructure. His work spans petabyte-scale telemetry ingestion, fault-tolerant compute design, and SRE-oriented system reliability.

Distributed Systems

Stream Processing

Cybersecurity Infra

Agenda

What we'll cover in the next 30 minutes

01

The Detection Latency Problem

Why batch-based SIEM architectures are failing at scale

02

Architecture Design

Compute-sharded stream processing — principles and components

03

Ingestion & Detection

Schema-agnostic pipelines and hybrid rule + ML detection

04

Multi-Region Resilience

Geographic fault tolerance and consistency guarantees

05

Validation & Case Studies

Measured outcomes and operational lessons learned

The Adversary Doesn't Wait for Your Batch Job

Modern attackers operate with speed and automation. Lateral movement, credential harvesting, and persistence establishment can complete in **minutes** often within a single batch processing window.

Detection Gap

Batch cycles introduce minutes-to-hours of latency a window attackers exploit fully.

Alert Delay

By the time a SIEM generates an alert, the threat actor may already have persistence.

Scale Mismatch

Petabyte-scale telemetry overwhelms architectures designed for gigabyte-era data volumes.

Why Traditional SIEM Architectures Break at Scale

Traditional SIEM platforms were designed for a different era. They rely on **scheduled batch cycles** that introduce structural latency not tunable latency, but architectural latency baked into every layer of the pipeline.

At petabyte scale, this creates compounding problems: ingestion queues back up, correlation windows grow stale, and alert fidelity degrades under load.

Core Architecture: Compute-Sharded Stream Processing

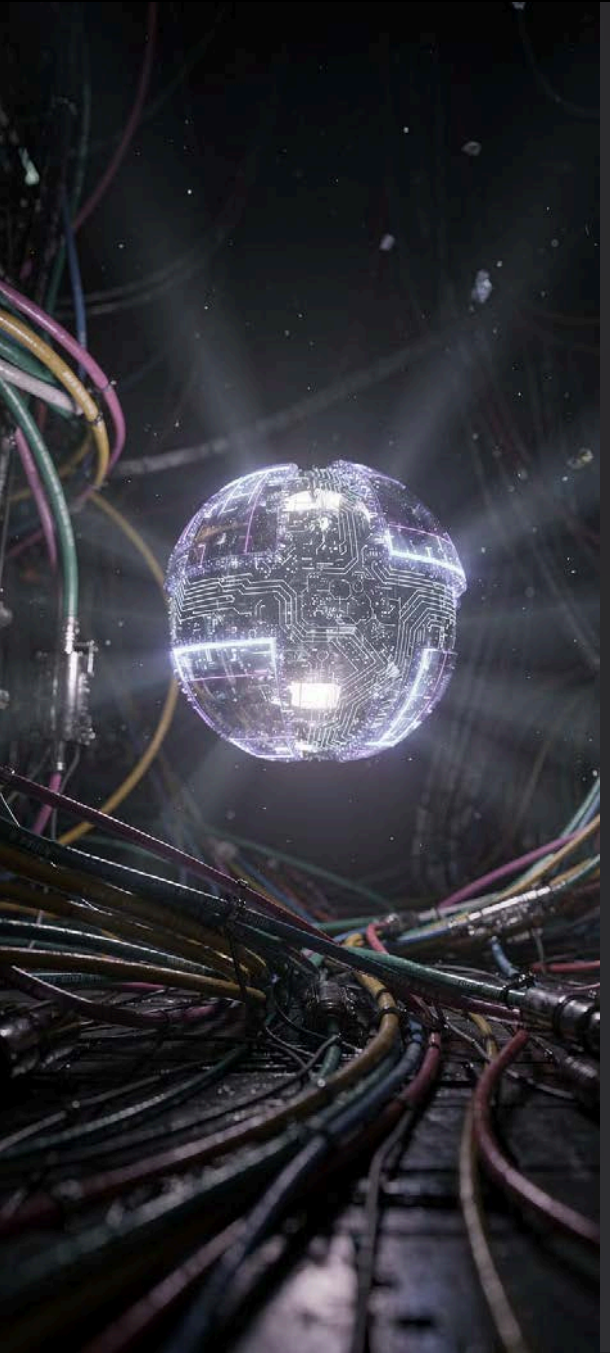
The architecture distributes **both data and computation** across independent processing nodes using consistent hashing. Each shard operates autonomously, eliminating centralized bottlenecks while enabling linear horizontal scalability as data volumes grow.

Consistent Hashing: Distributing Work Without Centralized Bottlenecks

Why Consistent Hashing?

Consistent hashing assigns both data streams and compute workloads to nodes on a virtual ring. When nodes are added or removed, only a minimal subset of keys remaps enabling elastic scaling without full reshuffling.

- Predictable, deterministic shard assignment
- Minimal rebalancing overhead during scale-out events
- Fault-isolated: a failed node affects only its key range
- Enables stateful stream correlation per shard without global coordination



Schema-Agnostic Ingestion: Onboard Any Source, Instantly

Security telemetry is inherently heterogeneous firewall logs, EDR events, cloud audit trails, network flows, identity signals. Requiring predefined schemas creates integration friction and slows threat coverage expansion.

Dynamic Schema Inference

The ingestion layer infers structure at runtime, eliminating upfront schema engineering for new data sources.

Faster Source Integration

New telemetry sources can be onboarded without pipeline halts or schema migration cycles.

Normalization at Ingest

Lightweight normalization at the ingestion boundary ensures downstream processing consistency without heavy preprocessing.

Continuous Stream Processing vs. Batch Cycles

Structural Latency Eliminated

Batch systems process data in discrete scheduled windows every event waits until the next cycle runs. Continuous stream processing evaluates every event as it arrives, enabling correlation and detection within seconds rather than minutes.

The latency gap is not marginal it is **an order of magnitude**, and it compounds across multi-hop detection chains.

Hybrid Detection: Rules + Machine Learning

Two Complementary Detection Layers

No single detection strategy covers the full threat landscape. This architecture combines deterministic rule-based detection with ML-driven anomaly identification to maximize coverage breadth and fidelity.

Rule-Based Detection

Known attack signatures, compliance checks, threshold-based alerts. Low latency, high precision for defined threat patterns.

ML Anomaly Detection

Behavioral baselines, statistical deviation, unsupervised clustering. Catches novel behaviors and zero-day techniques without predefined signatures.

Multi-Region Deployment: Resilience at Global Scale

Geographic distribution is not just a reliability strategy — it is a compliance and performance necessity for global enterprise security operations.

Geographic Resilience

Regional failure isolation ensures that a single region outage does not compromise global detection coverage.

Localized Processing

Data residency and sovereignty requirements are met by processing telemetry within designated regional boundaries.

Cross-Region Consistency

Distributed consensus mechanisms maintain detection rule and state consistency across all active regions without tight coupling.

Fault Tolerance

Automatic shard re-assignment and regional failover ensure processing continuity under node or zone-level failures.

Validation Results: What the Architecture Delivers

Validation across deployment environments demonstrated consistent, measurable improvements over batch-based SIEM baselines across all key operational dimensions.

Latency Reduction

Detection latency reduced by an order of magnitude compared to scheduled batch cycles

Sustained Throughput

Processing throughput remained stable under increasing workload — no degradation at scale

Resource Utilization

Improved compute utilization per unit of telemetry processed versus centralized architectures

Incident Response

Measurable reduction in mean time to respond across case study deployments

SRE Takeaways: Building This in Production

Key architecture principles and operational lessons for SRE teams building real-time cybersecurity pipelines at scale.

1

Design for Horizontal Scale from Day One

Vertical scaling hits ceilings quickly at petabyte volumes. Sharding strategy must be a first-class design decision, not a retrofit.

2

Decouple Ingestion from Processing

Schema-agnostic ingestion layers allow data source expansion without pipeline freezes critical for maintaining threat coverage velocity.

3

Instrument Everything, Alert on SLOs

Processing lag, shard rebalance duration, and detection pipeline depth are SRE-grade SLIs. Define SLOs before production rollout.

4

Test Failure Modes, Not Just Happy Paths

Node loss, region partition, and backpressure cascades must be chaos-tested. Resilience is not emergent it is engineered.

Thank You!

Abhishek Suman · Senior Software Engineer, Microsoft Corporation

Conf42 Site Reliability Engineering (SRE) 2026

Detection latency is an architecture decision. Build systems that are fast enough to matter when it counts.