OpenSearchCon EUROPE

Alert Fatigue to Action

Observability as the backbone of practical AIOps with OpenSearch

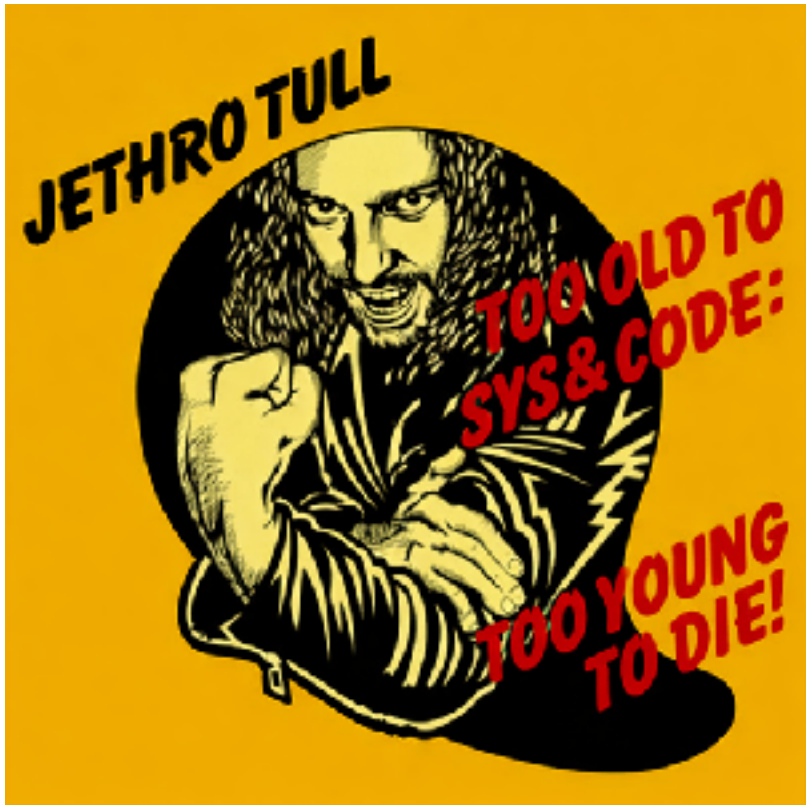
Stefano Pampaloni
CEO, Seacom Srl
VP, RIOS

OpenSearch Software Foundation AMBASSADOR 2026

Who I am

I’m no longer a developer.
I’m no longer a sysadmin.

President & CEO, Seacom Srl (Gruppo Itway) and
Member, OpenSearch Foundation
VP, RIOS — Italian Open Source Business Network
OpenSearch Ambassador



The passion never left.

The problem

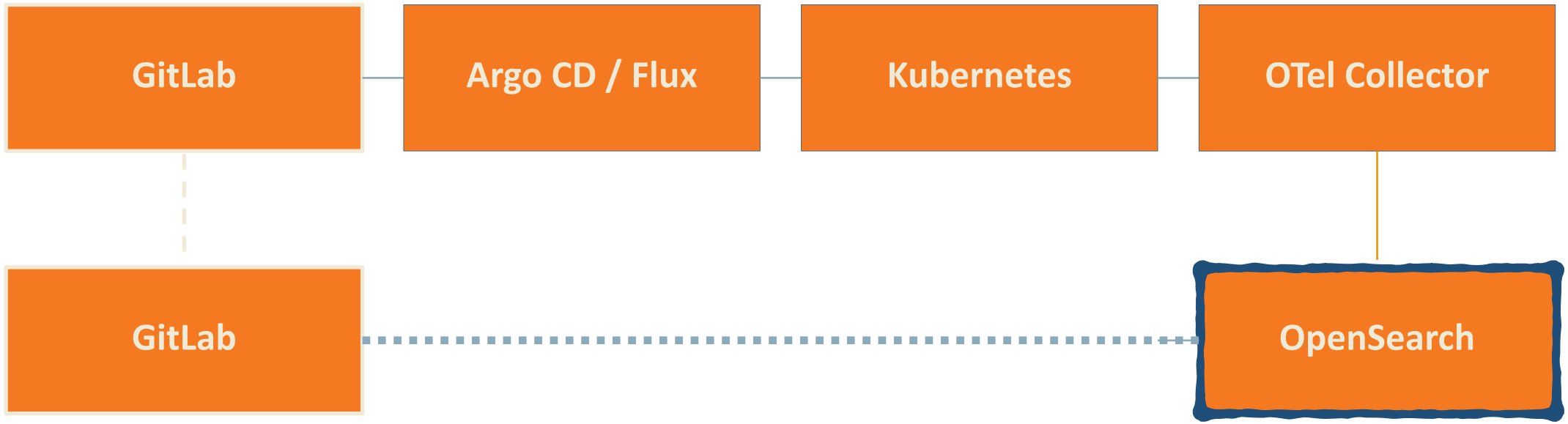
Alert fatigue is not a data problem.

Logs · Metrics · Traces · Alerts
We have all of this.

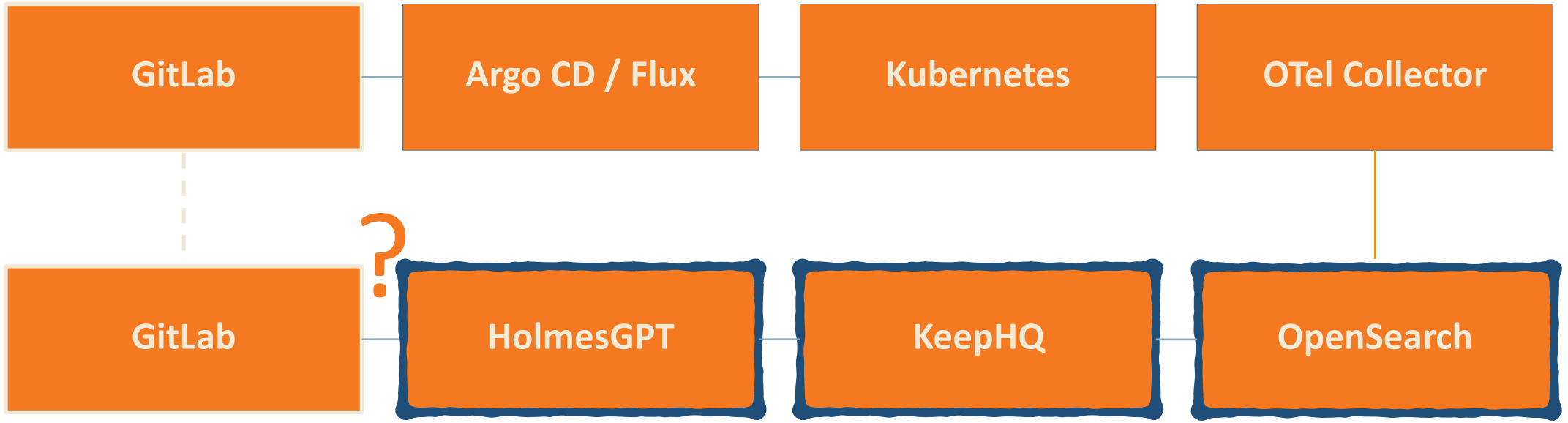
During incidents, engineers still correlate signals manually.
Under pressure. Relying on experience. Without shared context.

The data is there. The problem is fragmentation and an uncompleted circle

From GitLab to GitLab.



From GitLab to GitLab



OpenSearch as backbone

One platform. All signals. One query.

What it collects:

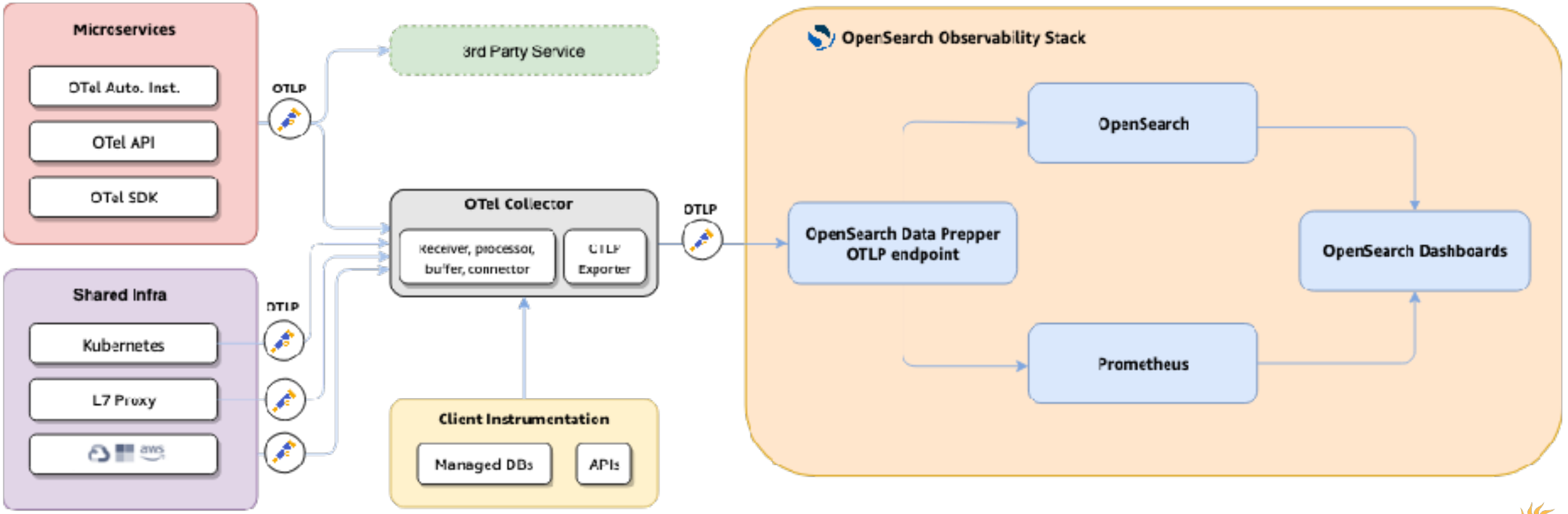
- Logs, metrics, traces via OTLP / Data Prepper
- Prometheus metrics via native Prometheus integration and OpenTelemetry-based ingestion
- Deployment context from pipeline annotations
- Runbooks and incident history as vector index

What you can do:

- Cross-signal queries in one language (PPL / SQL)
- Semantic search on past incidents
- Anomaly detection — built-in ML
- **AI agents operating via MCP and more**

OpenSearch Observability Stack

OpenSearchCon EUROPE



https://observability.opensearch.org/

The architecture

Sources push signals in. Consumers query out. OpenSearch holds the shared context.

SOURCES
OTel Collector
Prometheus
GitLab CI
→ push signals

→

OpenSearch
logs · metrics · traces
deployment registry
runbooks · vector index
MCP → AI agents

→

CONSUMERS
KeepHQ
HolmesGPT
GitLab MR
← query context

KeepHQ

Alert control plane — enrichment from OpenSearch is the key capability.

Receive

- 120+ sources: Prometheus, OpenSearch alerts, Zabbix, Grafana…

Correlate & deduplicate

- Group by namespace, service, timeframe. Reduce noise before AI.

Enrich from OpenSearch

- Query deployment context, similar incidents, runbook matches, before the workflow runs. This is what gives AI agents operational context.

Trigger AI workflow

- KeepHQ calls HolmesGPT. Enriched context travels with the alert.

NOTE: KeepHQ was acquired by Elastic in 2025. Development has slowed.
The architectural pattern remains valid with alternative tools.

HolmesGPT

OpenSearchCon EUROPE

Investigation engine. Operates on data retrieved from OpenSearch.

Receives enriched alert from KeepHQ

- alert context already includes OpenSearch enrichment.

Queries OpenSearch directly via MCP toolset

- Retrieves correlated logs, spans, deployment records at investigation time, not pre-indexed context.

Queries Kubernetes API and Prometheus

- Multi-source reasoning. The agent executes live queries, not text retrieval.

Returns hypothesis + confidence to KeepHQ

- Root cause, suggested action, confidence score. **KeepHQ workflow opens the GitLab MR.**

confidence < 50% → escalate to human. Not auto-remediation.

Closing the loop

How does the system know where to open the MR? OpenSearch is the operational memory that makes this possible.

At deploy

Pipeline annotates K8s Deployment:

repo · file-path · branch

OpenSearch stores snapshot:

project_id · config values · timestamp

→

At incident

HolmesGPT reads annotations via kubectl

Queries OpenSearch for deployment record

Builds payload:

project_id · file_path · change

→

The workflow

KeepHQ uses GitLab provider with payload

Opens MR

author: keephq-bot

MCP + agentic AI

OpenSearch as a tool for AI agents.

MCP Server
OpenSearch exposes search, query, and retrieval as tools that LLM agents can call during reasoning.

Agents reason over live data
Not pre-indexed static context. The agent queries OpenSearch at investigation time — logs, metrics, deployment history, similar incidents.

RAG on operational knowledge
Runbooks and incident history stored as vectors. The agent retrieves relevant context before suggesting a remediation action.

Not RAG on logs. Not a chatbot. An agent that reasons over live operational data.

Takeaways

OpenSearchCon EUROPE

01 Noise reduction comes before AI.

An AI agent given 200 uncorrelated alerts is useless. KeepHQ enriched with OpenSearch context is a great advantage.

02 OpenSearch is the reasoning substrate.

Consistent schema, cross-signal queries, vector search, MCP exposure

03 Git governs what AI proposes.

Every action is a reviewable MR. Not self-healing. AI-suggested, OpenSearch-grounded, Git-governed, human-accepted.

Do we really need dedicated tools to RCA, Workflows etc?

Openclaw and similar agents can monitor, investigate and find solutions to problems

I've used the Claude Cose Agent Skills from https://observability.opensearch.org/docs/claude-code/ and in 2 minutes my agent became super powerful, then I added an MCP client to Opensearch too.

OpenSearchCon EUROPE

Questions?

References

github.com/keephq/keep
github.com/robusta-dev/holmesgpt
observability.opensearch.org

OpenSearchCon EUROPE

Contacts

Stefano Pampaloni