

USE CASE 03 · FOR OBSERVABILITY LEADS · SRE · PLATFORM

SIX SYSTEMS. ONE **BINARY**. ONE BILL.

Logs, metrics, and traces arrive at the same data plane. One binary replaces the Splunk + Prometheus + Loki + Tempo + Mimir stack for teams that want unified observability without a multi-vendor bill.

02 · THE OBSERVABILITY STACK IS BROKEN

THREE PILLARS. SIX SYSTEMS. ONE ROOT CAUSE.

Observability teams pay for three data stores to answer one question: what happened. Logs are in one cluster, metrics in another, traces in a third, and the correlation query is a tab-switch between three UIs. AI agents expected to help with on-call cannot traverse the same evidence a human would.

COGNITIVE LOAD	SPL for logs · PromQL for metrics · TraceQL for traces. Three dialects for one incident. Junior SREs spend their first year learning the query languages before they can contribute to postmortems.
FAN-OUT COST	Ingest fans out to three stores. Three retention policies. Three compactors. Three backup schedules. The same event is stored three times in three shapes at three prices.
AGENT-UNFRIENDLY	On-call agents need one query surface. Three APIs with three semantics means the agent either learns an orchestrator or defers to a human. Either way the MTTR story stalls.
OPERATIONAL SPRAWL	12 config files. 6 Helm charts. 5 monthly bill lines. Every version upgrade touches three teams. On-call rotations split by telemetry type, not by ownership.

03 · UNIFIED AI DATA STACK

ONE ENGINE. THREE Pillars. ONE QUERY.

Logs, metrics, and traces into the same XERJ data plane. Prometheus remote_write on the metrics side (roadmap Q3), OTLP on the traces side (beta), ES Bulk on the logs side (GA). One query language across all three — ES DSL today, SQL roadmap Q4. Agents query the same surface humans do.

CONSOLIDATION

6 → 1**SYSTEMS
REPLACED**

Splunk · Prometheus · Tempo · Loki · Mimir · session store — one binary.

INGEST

80 K**DOCS/SEC
SUSTAINED**

Single-client, diurnal traffic, zero write-stalls. Scales horizontally on sharded ingest (roadmap Q3).

LANGUAGES

3 → 1**ES DSL
+ SQL ROADMAP**

One query dialect across logs + metrics + traces. Agents use one retriever.

COST

5 → 1**BILL LINES
PER MONTH**

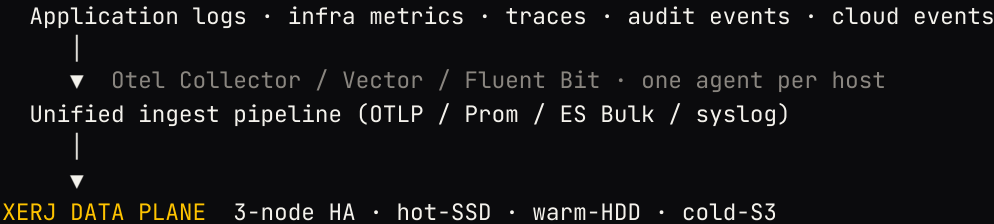
One vendor, one contract, one SLA, one audit surface, one on-call rotation.

04 · OBSERVABILITY BLUEPRINT

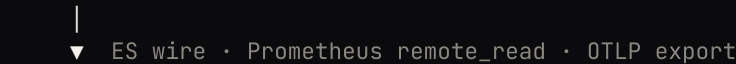
ONE PIPELINE. THREE PILLARS.

A three-node XERJ cluster serving logs, metrics, and traces for a mid-size engineering org. Same platform, one ingest surface, one retention policy framework, one dashboard layer.

SOURCES



LOGS	columnar · delta-ts · dictionary · ZSTD
METRICS	columnar · down-sampled rollups (roadmap Q3)
TRACES	span tree · service-graph join (roadmap Q4)
INDEXES	FTS + vector (for log-embedding similarity)



CONSUMERS

- Grafana unified dashboards (logs + metrics + traces panel-per-type)
- Kibana for log-centric workflows
- On-call runbooks · alerting via threshold + anomaly (roadmap Q4)
- SRE agents · runbook automation · RCA assistant

RETENTION

Logs	hot 14 d · warm 30 d · cold 90 d (object storage)
Metrics	raw 30 d · downsampled 1 y · yearly rollups archived
Traces	full 7 d · head-sampled 30 d · tail-sampled on error

05 · WHY THIS SCALES

ENGINEERING CHOICES FOR OBSERVABILITY.

Six decisions that make a single platform viable for the three pillars. Each is a property of the data plane, not an integration story.

COLUMNAR LOGS	Delta-of-delta timestamps · dictionary keywords · varint integers · ZSTD block compression. 85 MB per 1 M events — 2.8× smaller than ES on the same corpus. Makes long retention affordable.
SINGLE-PROCESS INGEST	Parsers live in the writer process. One fsync, one network hop, end-to-end back-pressure. No Kafka queue as a buffer, no Logstash pipeline to babysit, no separate ingest service to monitor.
LOCK-FREE READERS	Flush never blocks analyst queries. A 24-hour ingest surge writes through without stalling the dashboard. ArcSwap segment view swaps atomically; in-flight queries finish on the old view.
NO GC TAIL	Rust · jemalloc · no JVM. p99.9 tracks p50 plus cache-miss and disk I/O. Observability SLOs that require bounded latency become achievable without tuning.
EXPLAIN-PLAN	Every query — log search, metric aggregation, trace fetch — returns per-node cost. Capacity planning and cost attribution share one JSON shape.
PROMETHEUS METRICS NATIVE	XERJ's own /metrics endpoint is a first-class surface. Scrape XERJ with XERJ if the self-referential story appeals. Dogfooding is a demo, not a workaround.

06 · MEASURED OUTCOMES

OPERATIONAL SURFACE. COLLAPSED.

Every axis that shows up in a procurement review. XERJ on one platform vs. a Splunk + Prometheus + Tempo reference stack doing the same job.

DIMENSION	XERJ	SPLUNK + PROM + TEMPO
BINARIES TO DEPLOY	1	6 — 6×
CONFIG FILES	1	12 — 12×
QUERY LANGUAGES	1 (ES DSL)	3 (SPL + PromQL + TraceQL)
STORAGE BACKENDS	1	3 — 3×
RETENTION POLICIES	1 unified	3 separate
MONTHLY BILL LINES	1	5 — 5×
IDLE CLUSTER MEMORY	400 MB · 1 node	8.5 GB · 4-node ES baseline
COLD START	50 ms	~15 s JVM warm-up
OPERATIONAL IMPACT	One on-call rotation instead of three. Junior SREs productive in weeks, not quarters. One upgrade cycle instead of six.	
COST IMPACT	~80 % infrastructure cost reduction on consolidated log + search + vector workloads. Most of the saving is headcount: one team owns the data layer, not four.	

07 · YOUR OBSERVABILITY STACK

OTEL. PROMETHEUS. DROP-IN.

XERJ speaks the standard wire protocols. Every collector that already ships to ES or OpenTelemetry works as-is. Dashboards keep their queries.

CATEGORY	TOOLS	STATUS
COLLECTORS	OpenTelemetry Collector · Vector · Fluent Bit · Fluentd · Otel SDKs · Otel Operator	GA
METRICS IN	Prometheus remote_write · native pull scrape for ES metrics · Otel metrics via OTLP	ROAD Q3
TRACES IN	OTLP/HTTP · OTLP/gRPC · Zipkin adapter (via Otel Collector) · Jaeger adapter	BETA
LOGS IN	ES _bulk · Filebeat · Vector · Fluentd · direct JSON over HTTP · syslog (roadmap)	GA
DASHBOARDS	Grafana · Kibana · custom · unified logs-metrics-traces panels via Grafana 10+ mixed datasources	GA
ALERTING	Alertmanager-compatible rules · Grafana alerting · webhooks · PagerDuty · Slack · anomaly detection (roadmap Q4)	GA
EXPORT	OTLP/HTTP export to downstream systems · Prometheus remote_read · ES search API · long-term S3 archives	GA
IDENTITY	OIDC · SAML · AWS IAM · Azure AD · per-team index isolation with quotas	ROAD Q2
INCIDENT MGMT	PagerDuty · Opsgenie · Incident.io · ServiceNow · via webhook · runbook-aware alert enrichment (roadmap)	GA

08 · DAY-TWO OPERATIONS

ONE BINARY. ONE RUNBOOK.

Day-two operations are where observability platforms usually lose teams. XERJ is engineered for rolling restart, hot-reload config, and segment-level backup from day one.

METRICS	Prometheus on /metrics. 13 counters / histograms / gauges across ingest, query, flush, cache. Per-instance registry — no global-singleton label collisions in multi-tenant setups.
HEALTH / READINESS	GET /healthz for liveness · GET /readyz holds until WAL replay completes — rolling restart fleet-safe by default.
LOGS	Structured JSON to stdout. One event per line. Correlation ID on every request. Tail with jq, ship with any log-forwarding agent.
UPGRADES	Rolling restart supported. Segment + WAL formats are forward-compatible across minor versions. One-minor-version window for rollback. Helm rolling update + HPA work out of the box.
BACKUP · RESTORE	Snapshots to S3 / GCS / Azure Blob. Incremental · encrypted · restorable at segment granularity. <code>xerj snapshot create list restore</code> .
CAPACITY PLANNING	~80 K events/sec sustained per node · diurnal pattern absorbed · write-stalls rare under documented sizing. Scale horizontally from 3 to 12 nodes as throughput grows.

09 · FROM PILOT TO CONSOLIDATION

FIVE PHASES. ONE YEAR.

A reversible rollout that swaps each pillar one at a time. Logs first (highest volume, clearest cost win), then traces, then metrics.

01	PILOT WEEKS 1–2	Single node · route one log stream (NGINX / app logs) to XERJ via Vector · keep existing SIEM primary · compare ingest cost, query latency on Kibana side-by-side.
02	SHADOW WEEKS 3–6	3-node HA cluster · dual-write all log streams · Grafana reads from both. Log dashboard A/B. OTLP traces beta testing on internal services.
03	MIGRATE LOGS QUARTER 2	Cut over logs primary. Decommission Splunk / Loki log index. First bill line removed. Retention policies unified. Kibana or Grafana as single log UI.
04	MIGRATE TRACES QUARTER 3	OTLP traces primary. Tempo decommissioned. Service-graph materialization lands. Runbook agents retrieve across logs + traces in one hybrid query.
05	MIGRATE METRICS QUARTER 4	Prometheus remote_write in primary. Mimir / Thanos long-term store retired. Three pillars, one engine. Single alerting config. One on-call rotation.

REVERSIBLE · ES + PROMETHEUS + OTLP MEAN NO LOCK-IN AT ANY PHASE

10 · NEXT STEPS

RUN IT. ON YOUR TELEMETRY.

A two-week pilot on one log stream. We ship the binary, Grafana dashboard JSON, and the ingest + query scripts. You bring one log stream and a Grafana baseline p95.

OBSERVABILITY CONTACT

HELLO@XERJ.AI

XERJ.AI · XERJ.COM · XERJ.DEV

WHAT YOU GET	Private binary · Otel Collector config · Vector config · Grafana dashboard JSON · Prometheus scrape examples · synthetic 24-hour ingest corpus.
WHAT YOU BRING	One production log stream · one Grafana dashboard · a p95 latency baseline for at least one existing query.
COMPANION BRIEFS	xerj-tech-brief.pdf (architect) · xerj-usecase-security.pdf for SIEM modernization · xerj-usecase-ops-intel.pdf for operational intelligence