PERCONA

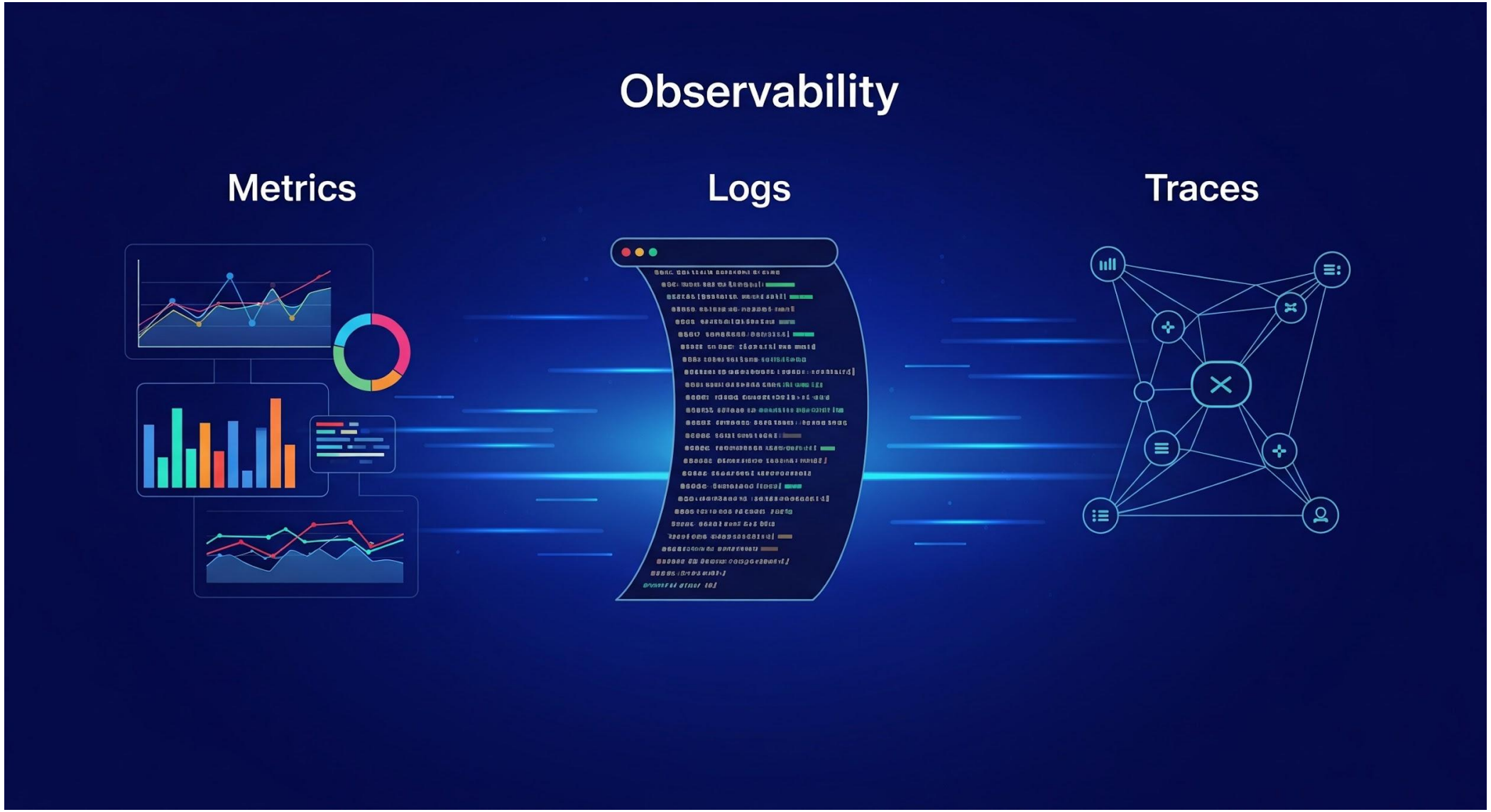
Logging in Percona Monitoring and Management

Alex Demidoff

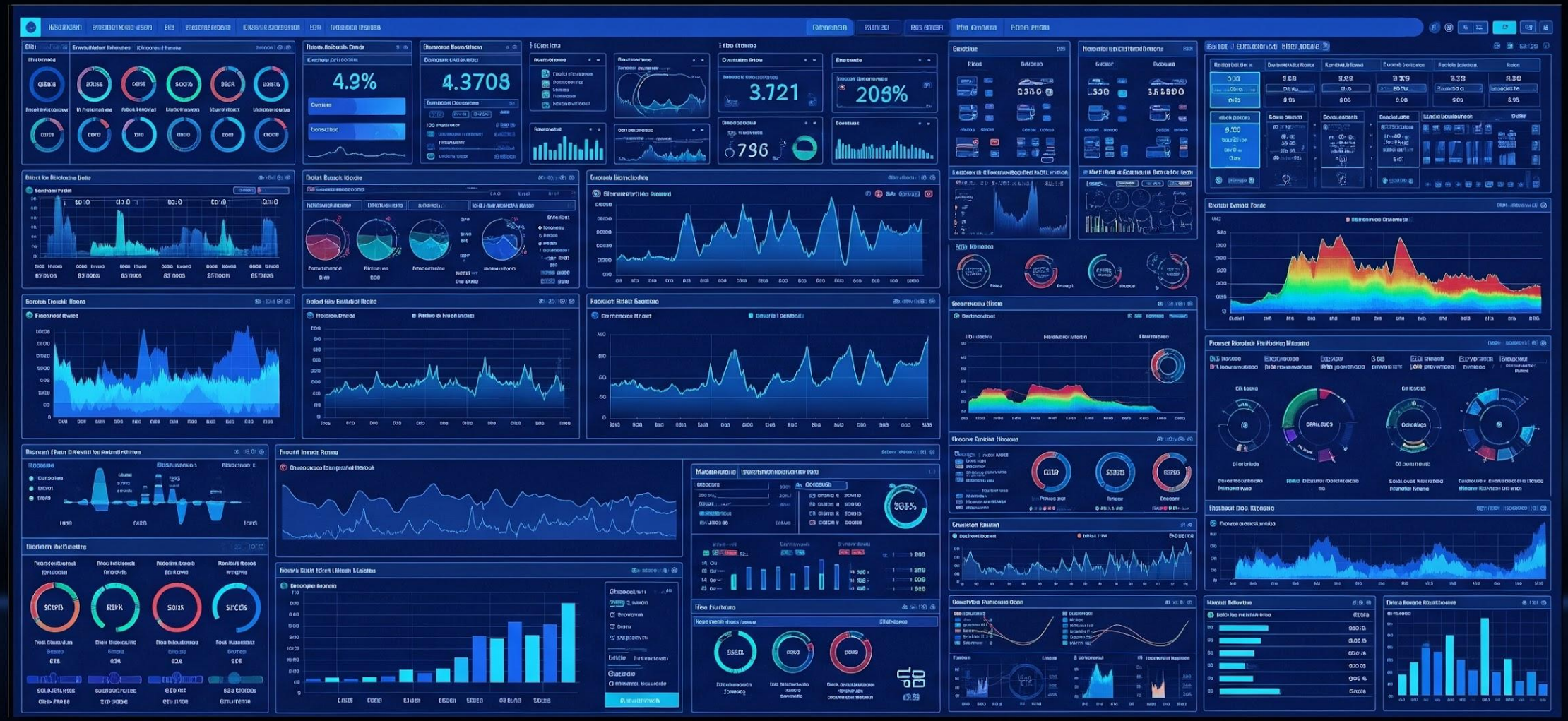
Erevan, August 2025

Agenda

1. Why do we need logging in PMM?
2. Architecture overview
3. Why embrace OpenTelemetry?
4. Demo
5. Cost and business benefits
6. HA-readiness



Metrics are hard to watch!



PERCONA

Solution - set up alerting and get notified.

Status quo

Lack of logging in PMM makes troubleshooting a real challenge. Grepping each individual log file to find the root cause does not scale.

Goal

PMM offers a turnkey logging solution integrated with alerting. Logs are accessible from a central place. Troubleshooting is made awesome.

Objectives

Log collection & processing

Collect logs from various sources, in different formats. Support on-the-fly transformations

Log storage

Persist logs in a structured, queryable format with proper schema management

Log search & filtering

Provide flexible querying capabilities for log retrieval and analysis. Support full-text search across log messages and metadata

Dashboard integration

Seamless integration with dashboards for log visualizations

Configurable log retention

Ability to configure the log retention via SQL

Integration with alerting

Support alerting based on log events and notify users via available channels (e.g., email, slack, etc.)

Scalability

Handle large volumes of log data efficiently

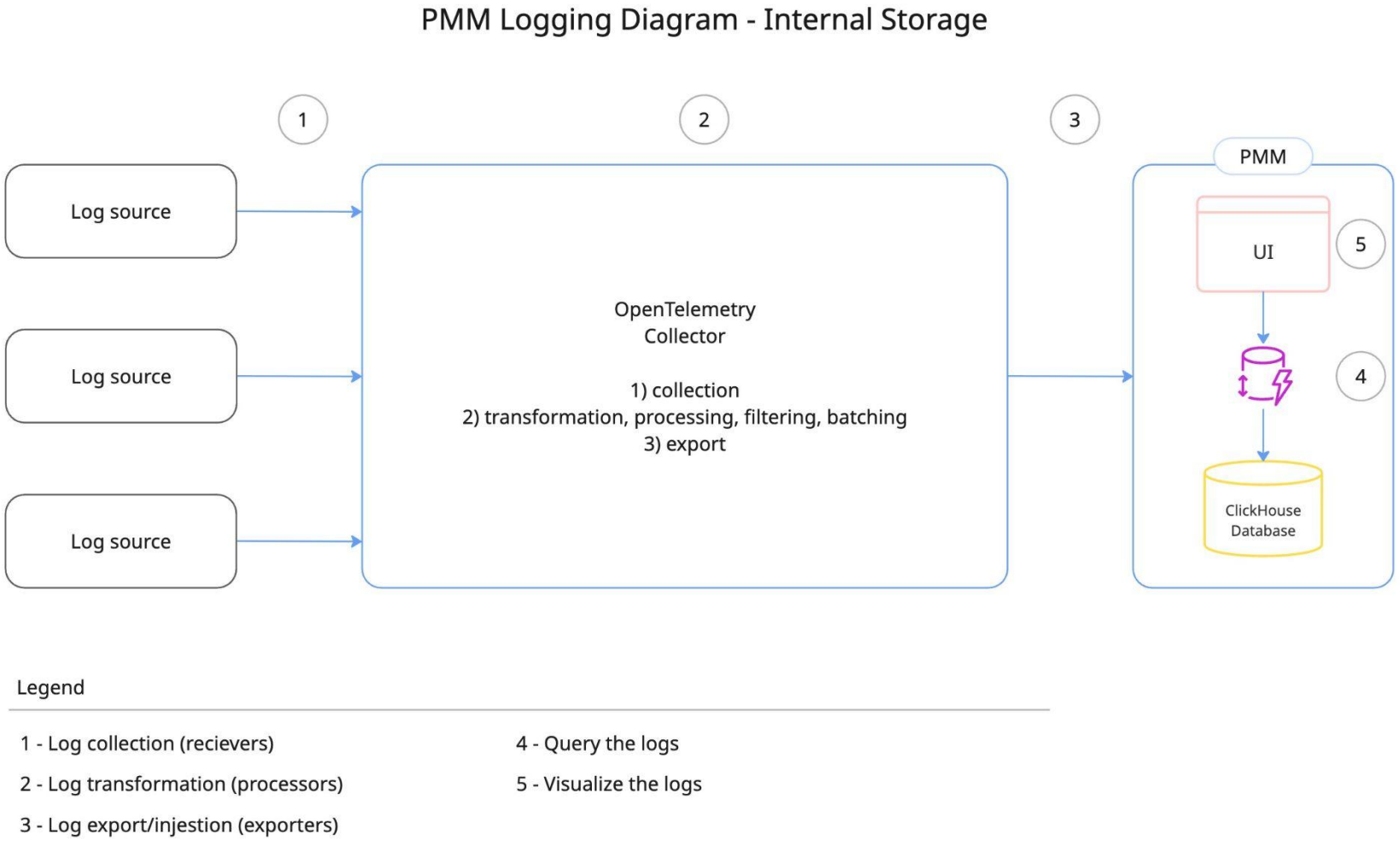
Developer-friendliness

Easy extension and modification architecture

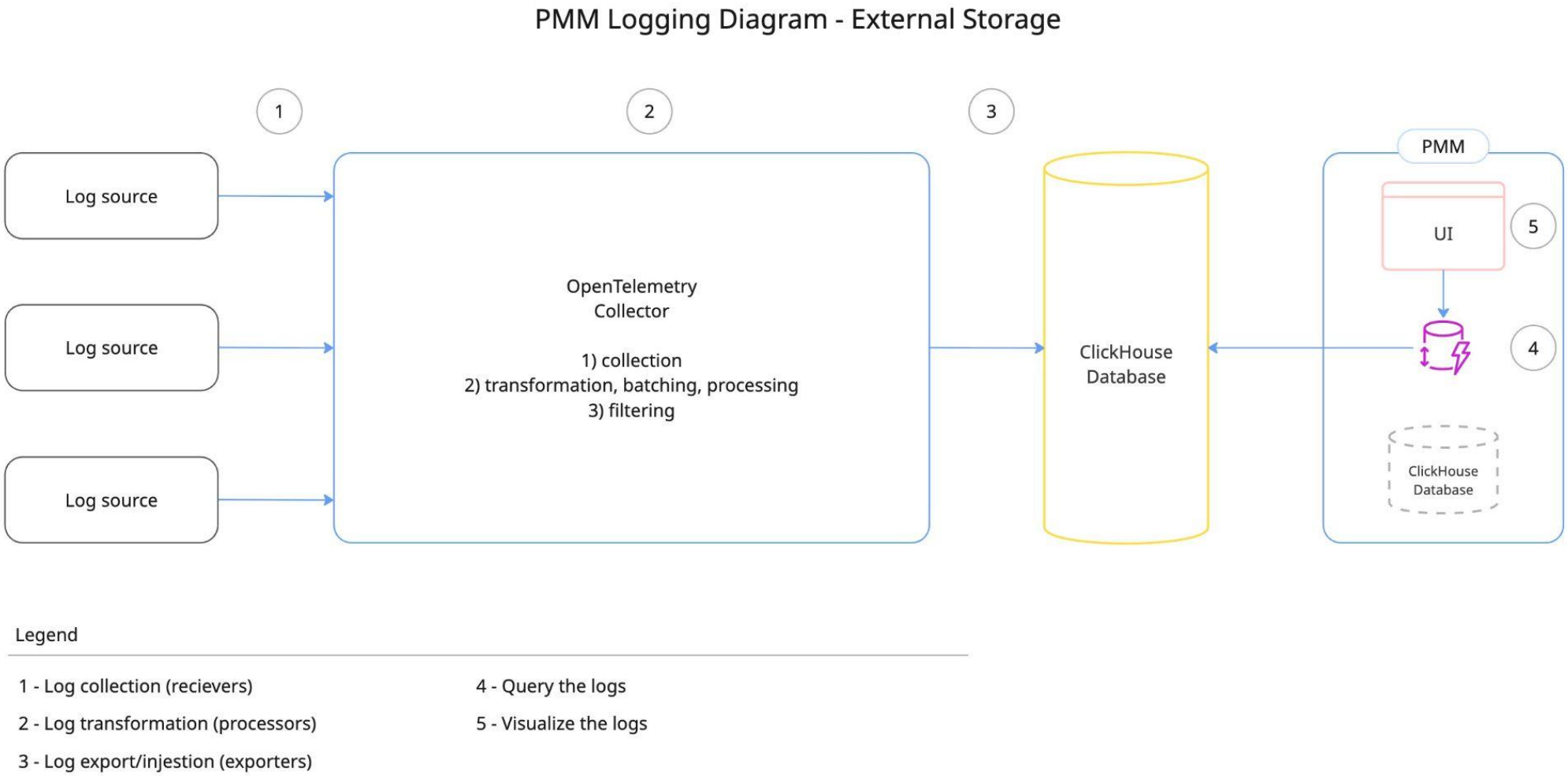
Open source

Stay true to open source principles

Architecture - Internal Storage



Architecture - External Storage



Architecture - Otel Collector Diagram

PERCONA

What's Open Telemetry?

The OpenTelemetry Protocol (OTLP) is a standardized protocol designed for collecting and transmitting telemetry data (metrics, logs, and traces) within the OpenTelemetry framework. It defines how this data is encoded and transported between different components like applications, collectors, and backends, ensuring interoperability and flexibility in observability systems.

Key Components

- PMM Server: Central component helping collect and visualize metrics and logs and manage the inventory
- ClickHouse: High-performance columnar database for log storage
- OpenTelemetry Collector: Agent for gathering, processing, and sending logs
- ClickHouse Datasource: Grafana plugin to query the logs

ClickHouse Features

- **Compression**: Reach choice of compression algorithms for different data types
- **Indexing**: Bloom filters and specialized indexes for fast queries
- **Partitioning**: Date-based partitioning for performance
- **TTL Management**: Automatic data retention (3-day default)

Database Schema (OpenTelemetry Standard)

```
CREATE TABLE otel.logs (
    `Timestamp` DateTime64(9) CODEC(Delta(8), ZSTD(1)),
    `TimestampTime` DateTime DEFAULT toDateTime(Timestamp),
    `SeverityText` LowCardinality(String) CODEC(ZSTD(1)),
    `ServiceName` LowCardinality(String) CODEC(ZSTD(1)),
    `Body` String CODEC(ZSTD(1)),
    `LogAttributes` Map(LowCardinality(String), String) CODEC(ZSTD(1)),
    -- Additional OpenTelemetry fields and indices...
) ENGINE = MergeTree
PARTITION BY toDate(TimestampTime)
ORDER BY (ServiceName, TimestampTime, Timestamp)
TTL TimestampTime + toIntervalDay(3)
```

Demo time!

Competitive advantages

Cost benefits

- **Reduced Storage Costs**: Minimal disk usage compared to alternatives
- **Lower Infrastructure Requirements**: Efficient resource utilization
- **No Licensing Fees**: Fully open source (Apache 2.0)
- **Reduced Operational Overhead**: Automated retention and management

Business benefits

- **Faster Troubleshooting**: High-performance queries enable rapid issue resolution
- **Scalable Growth**: Architecture grows with business needs
- **Standards-Based**: Future-proof with industry-standard protocols
- **Developer Productivity**: Streamlined development and testing cycles
- **Enterprise Ready**: Production-grade reliability and performance

Comparison with Loki and VictoriaLogs

	ClickHouse	Loki	VictoriaLogs
Disk Usage	🟢 Low	🔴 High	🟢 Low
Performance	🟢 High	🔴 Low	🟢 High
Scalability	🟢 High	🟡 Moderate	🟡 Moderate
Query Language	🟢 SQL	🟡 LogQL	🟡 LogsQL
Licensing	🟢 Apache 2.0	🟡AGPLv3	🟢 Apache 2.0

HA Compatibility

PMM can function in HA mode. With little to no modification we can make this solution highly available.

Why implement logging in PMM?

🎯 **Performance**: Up to 15x better performance than alternatives

💾 **Efficiency**: Minimal disk usage with maximum capabilities

🔧 **Flexibility**: OpenTelemetry standard ensures broad compatibility

📈 **Scalability**: Capable of handling enterprise-scale deployments

🚀 **Developer Experience**: Streamlined development and testing

💰 **Cost-Effective**: Open source with enterprise-grade features

What about AI?

Logs are a great data source to be consumed by AI, which will make PMM even more awesome :)

Resources

- Documentation:

  https://docs.percona.com/percona-monitoring-and-management/
- OpenTelemetry: https://opentelemetry.io
- ClickHouse: https://clickhouse.com
- Source Code: https://github.com/percona/pmm/pull/4230