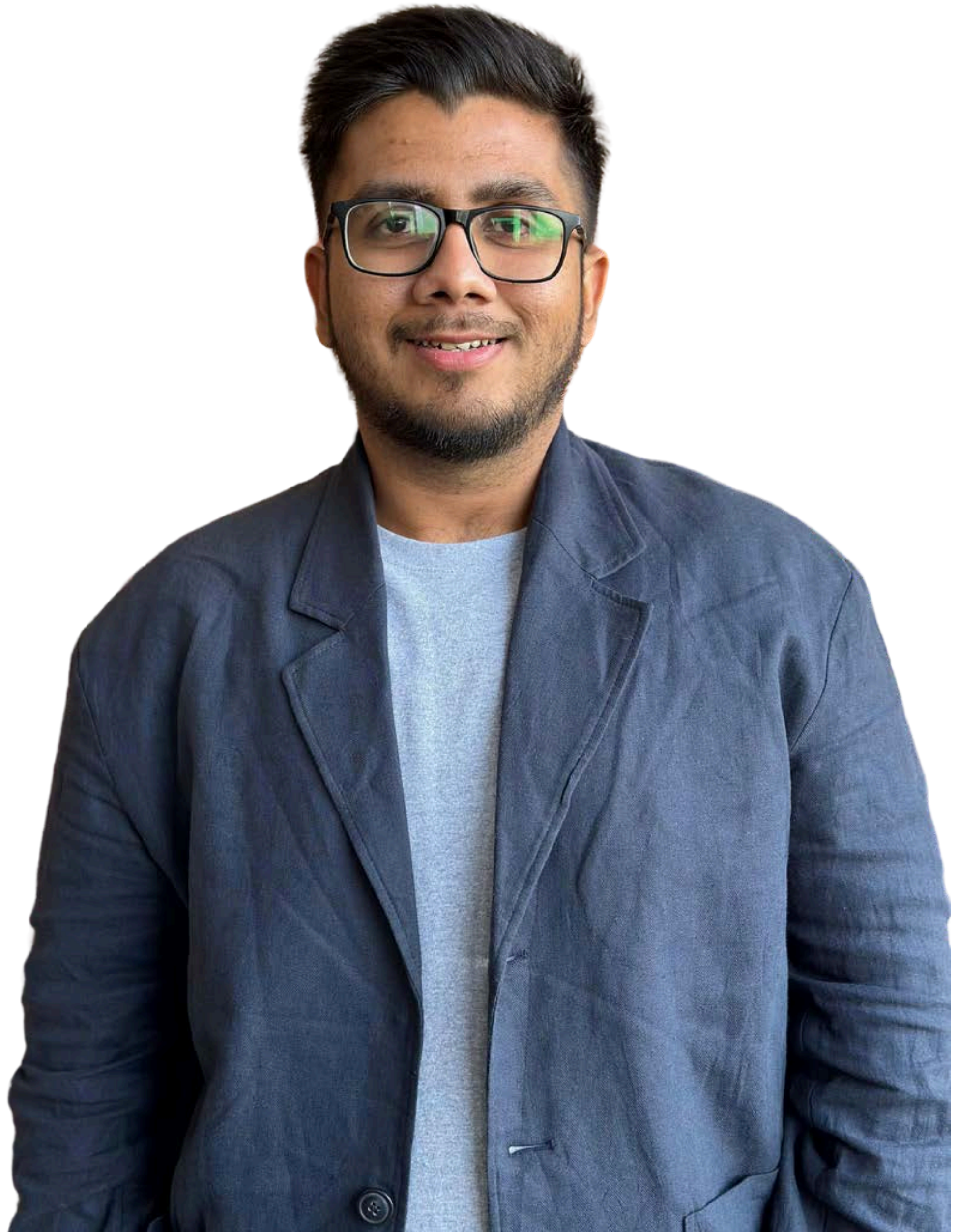
Observability 2.0: Much More Than Just Logs, Metrics, and Traces

By Neel Shah

Neel Shah - DevOps Community Guy

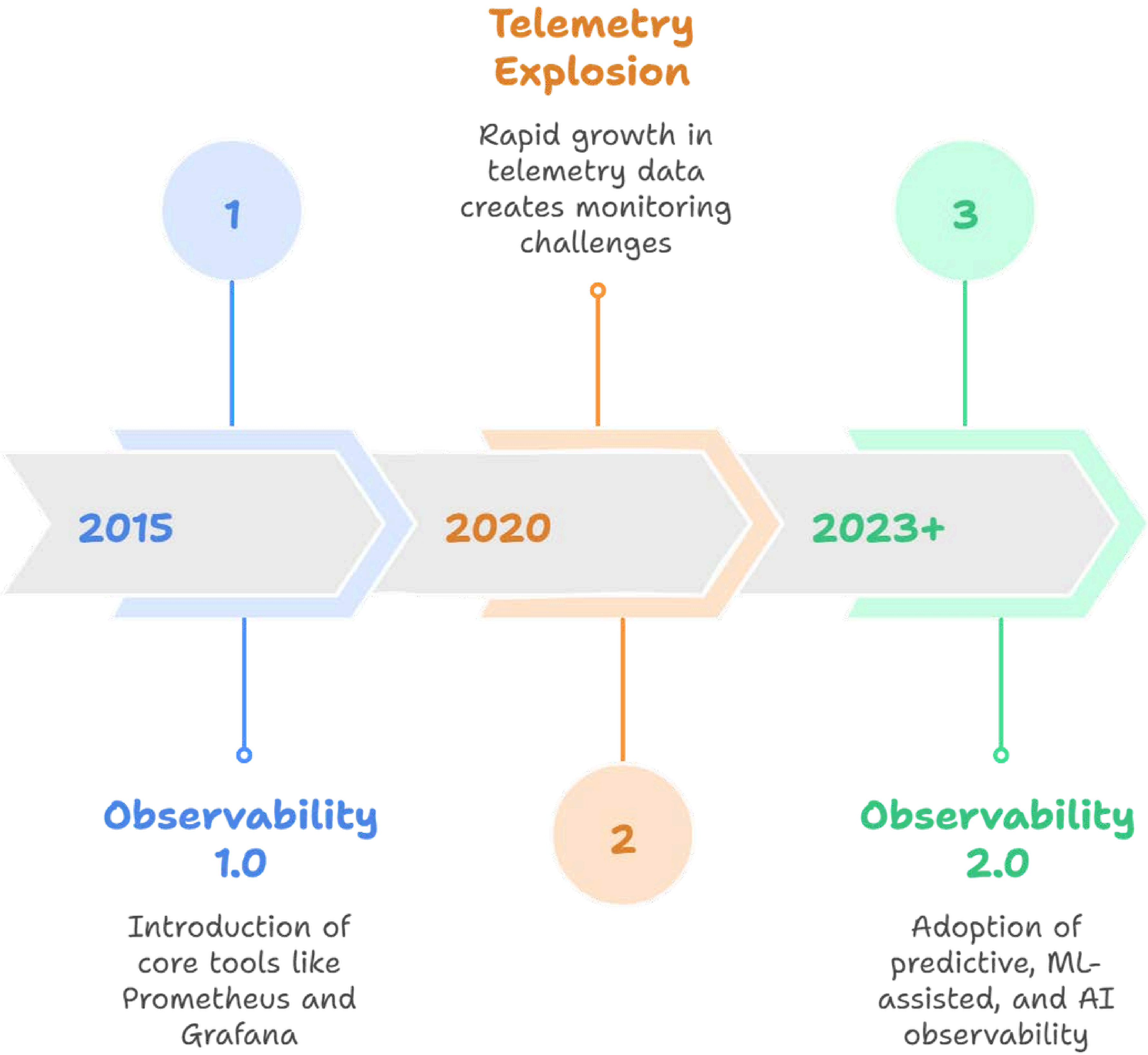
- Developer Advocate at Middleware
- Co-organiser GDG Cloud Gandhinagar, CNCF and Hashicorp Gandhinagar
- Mentored more than 15+ hackathons
- Gave talks in 10+ conferences, including Platform Con, LinuxFest, KCD, etc.



Agenda

- 🛠️ Evolution of Observability
- 🚀 OpenTelemetry's Role
- ⚡ Reducing Downtime with Otel Collector
- 🧠 ML and LLMs in Observability
- 💸 Optimising Observability Costs
- 🤖 AI-Driven Future
- 🎯 Takeaways

Evolution of Observability



Telemetry Explosion
Rapid growth in telemetry data creates monitoring challenges
1
3
2015
2020
2023+
Observability 1.0
Introduction of core tools like Prometheus and Grafana
2
Observability 2.0
Adoption of predictive, ML-assisted, and AI observability

From Firefighting to Foresight: Why Observability 2.0?

Imagine this:

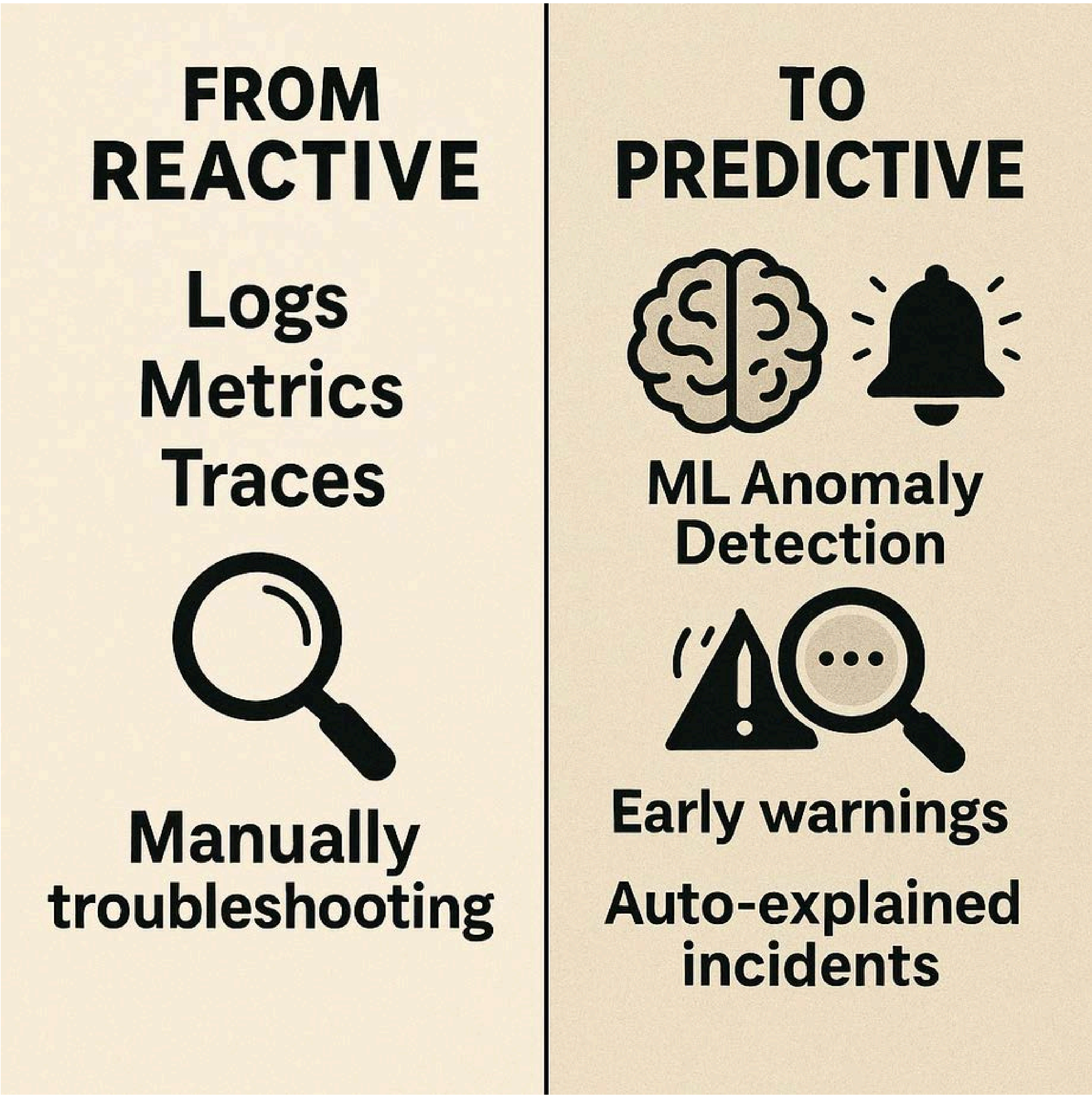
- It's 2 AM. Your production system is down.
- You’re flooded with millions of logs and scattered metrics.
- Hours of manual digging reveal the cause: a silent third-party outage.
- Meanwhile, downtime costs escalate, and customer trust erodes.

From Firefighting to Foresight: Why Observability 2.0?

Now imagine a better world:

- Your observability system detects subtle anomalies early.
- AI flags a growing risk before any major outage.
- You act proactively, avoiding the incident altogether.

From Firefighting to Foresight: Why Observability 2.0?



FROM REACTIVE

Logs
Metrics
Traces

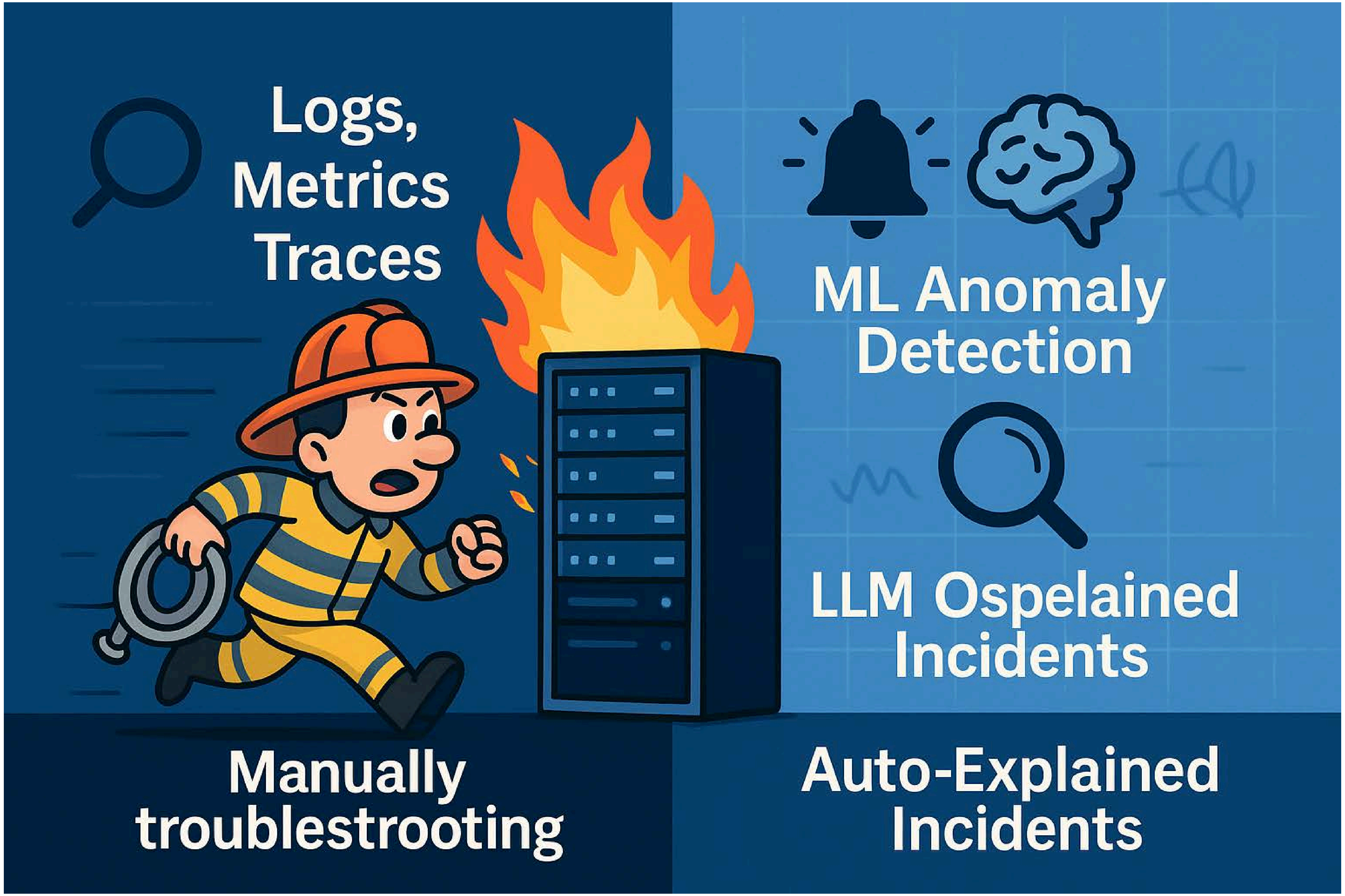
Manually troubleshooting

TO PREDICTIVE

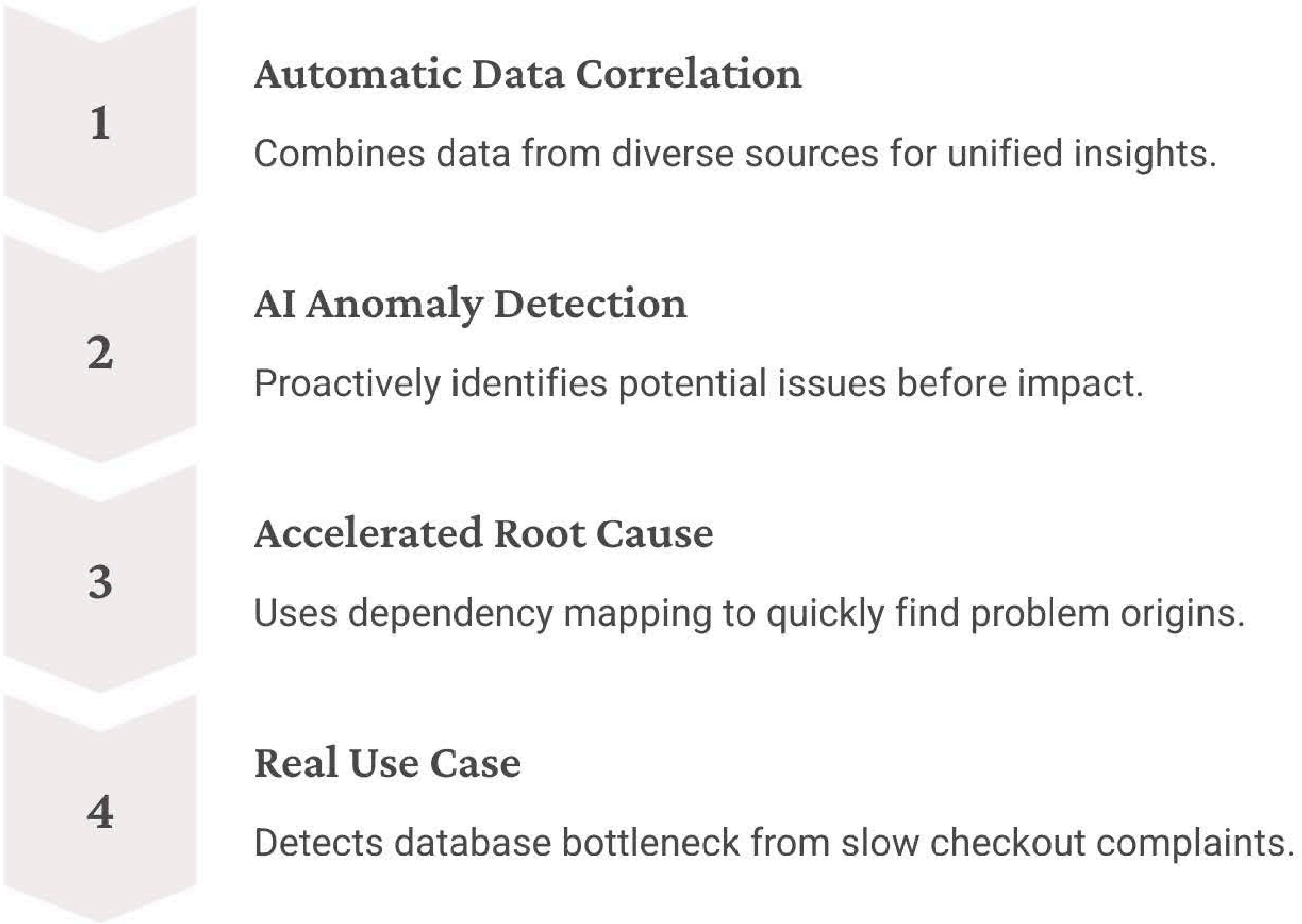
ML Anomaly Detection

Early warnings

Auto-explained incidents


Logs,
Metrics
Traces
ML Anomaly Detection
LLM Ospelained Incidents
Manually troublestrooting
Auto-Explained Incidents

Enhanced Capabilities: Correlation and Contextualization



1

Automatic Data Correlation

Combines data from diverse sources for unified insights.

2

AI Anomaly Detection

Proactively identifies potential issues before impact.

3

Accelerated Root Cause

Uses dependency mapping to quickly find problem origins.

4

Real Use Case

Detects database bottleneck from slow checkout complaints.

Today's Journey into Observability 2.0

- **Evolution of Observability:**

How traditional monitoring is giving way to smarter, predictive systems.

- **Rise of OpenTelemetry:**

Why Otel is becoming the new foundation for unified observability.

- **ML and LLMs in Observability:**

Applying AI to find anomalies, explain issues, and predict failures.

Today's Journey into Observability 2.0

- **Reducing Downtime with Otel Collector:**

Streamlining telemetry data to troubleshoot faster.

- **Cutting Costs with Smart Ingestion:**

Ingesting only what matters to control costs and reduce noise.

- **The AI-Driven Future of Observability:**

Self-healing systems, predictive insights, and autonomous operations.

Evolution of Observability

Challenges:

- Siloed data sources.
- Manual root cause analysis.
- Growing telemetry volume leading to higher costs and complexity.

Evolution of Observability

Observability 2.0 (Today):

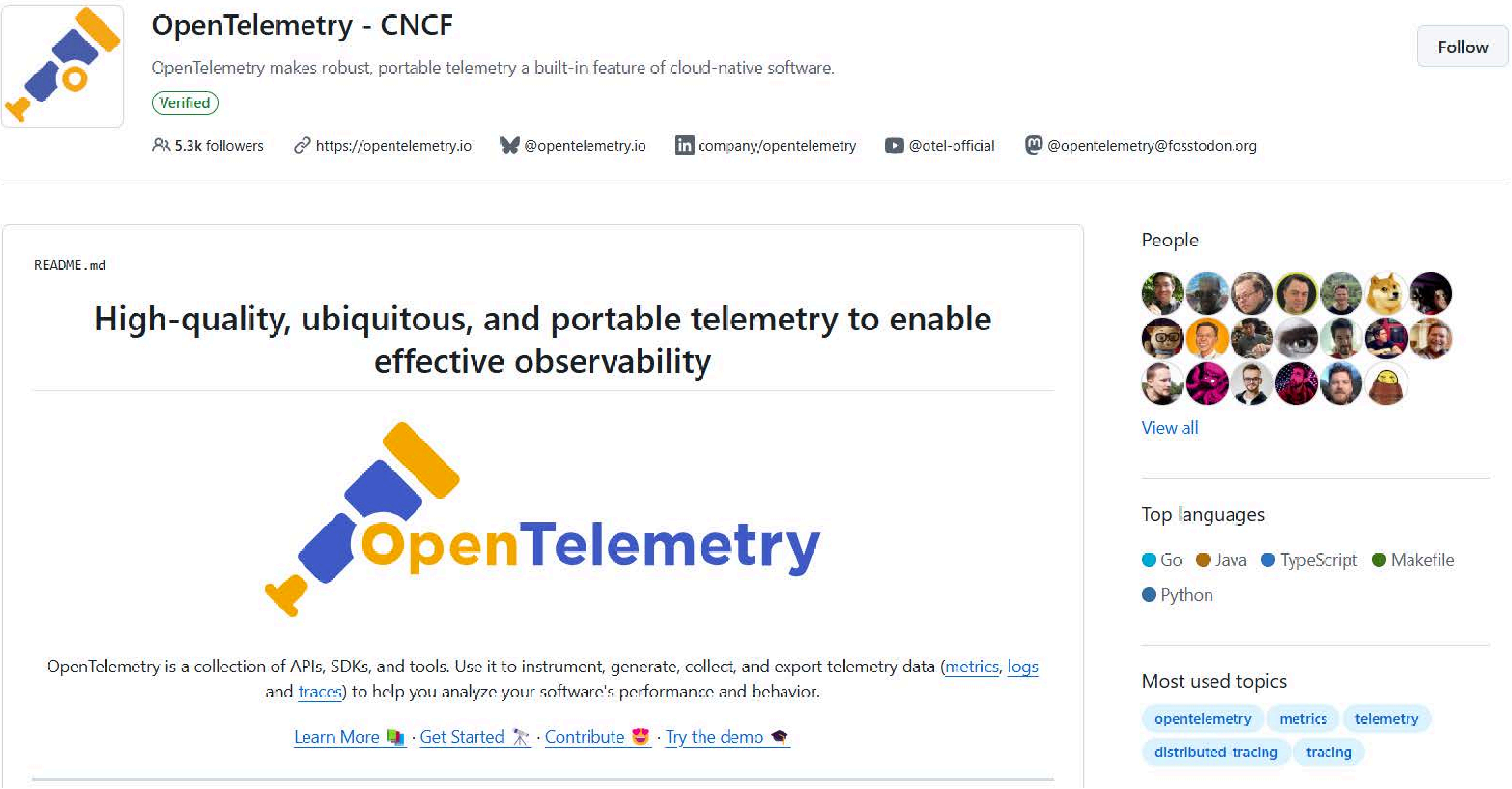
- Unified telemetry with OpenTelemetry (Otel).
- Proactive anomaly detection using ML models.
- Predictive insights, not just dashboards.
- Cost-optimized data pipelines

Rise of OpenTelemetry (Otel)

What is OpenTelemetry?

- A vendor-neutral open-source framework for collecting telemetry data — logs, metrics, and traces.

Rise of OpenTelemetry (Otel)



OpenTelemetry - CNCF

OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software.

Verified

5.3k followers · https://opentelemetry.io · @opentelemetry.io · company/opentelemetry · @otel-official · @opentelemetry@fosstodon.org

README.md

High-quality, ubiquitous, and portable telemetry to enable effective observability

OpenTelemetry is a collection of APIs, SDKs, and tools. Use it to instrument, generate, collect, and export telemetry data (metrics, logs and traces) to help you analyze your software's performance and behavior.

Learn More 📚 · Get Started 🔭 · Contribute 😍 · Try the demo 🎓

People

View all

Top languages

● Go ● Java ● TypeScript ● Makefile ● Python

Most used topics

opentelemetry · metrics · telemetry · distributed-tracing · tracing

Rise of OpenTelemetry (Otel)

Why OpenTelemetry Matters:

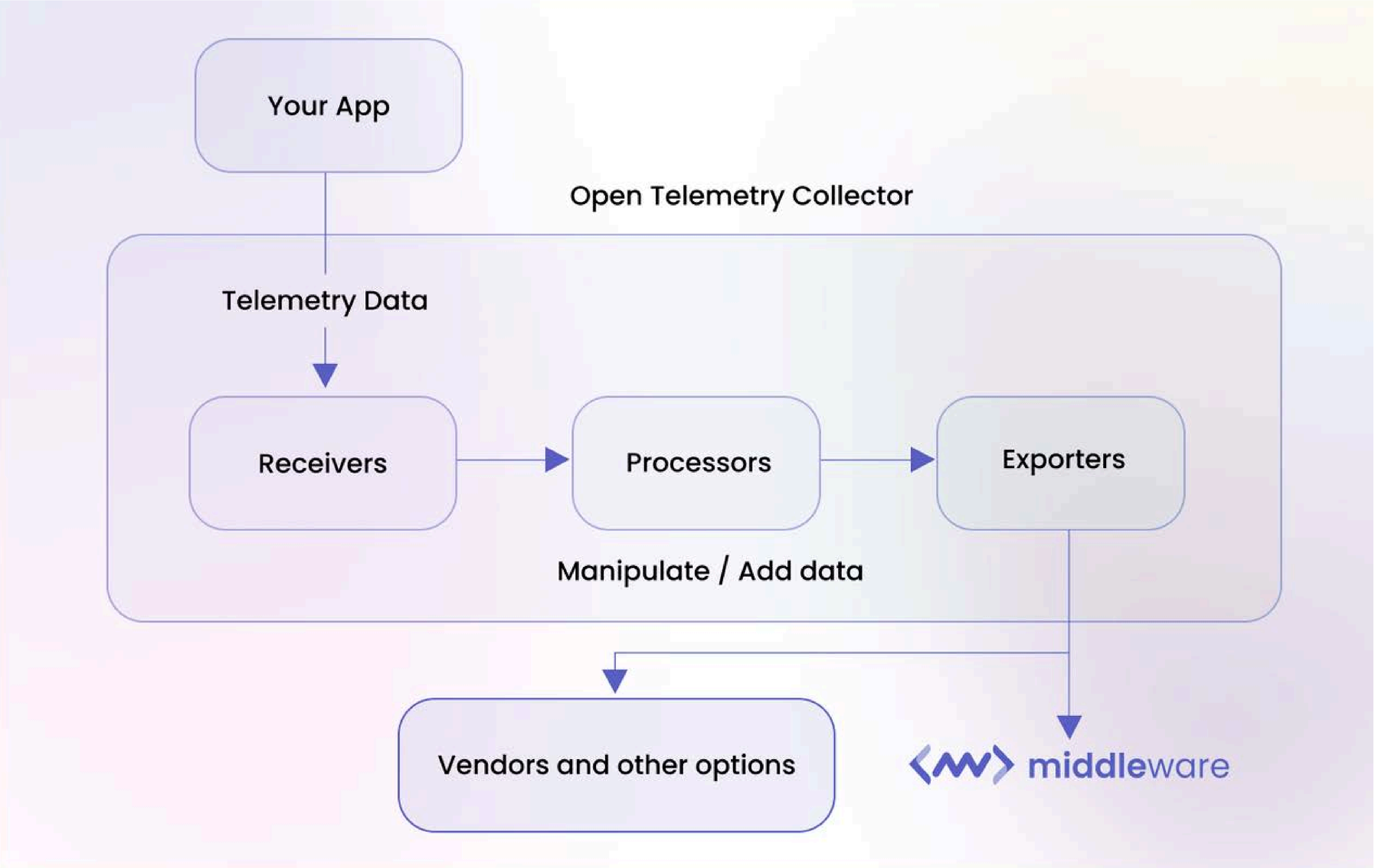
- **Unified and standardized instrumentation.**
- **Easier correlation between different signals (logs, metrics, traces).**
- **Freedom to route data to multiple backends and observability platforms.**
- **Extensible with processors, filters, and exporters.**

Rise of OpenTelemetry (Otel)

Impact on Observability:

- Reduces engineering effort for monitoring setup.
- Enables smarter pipelines for data processing and cost control.
- Forms the foundation for AI-driven observability systems.

OTel Collector



SCAN ME

ML and LLMs in Observability

Machine Learning in Observability:

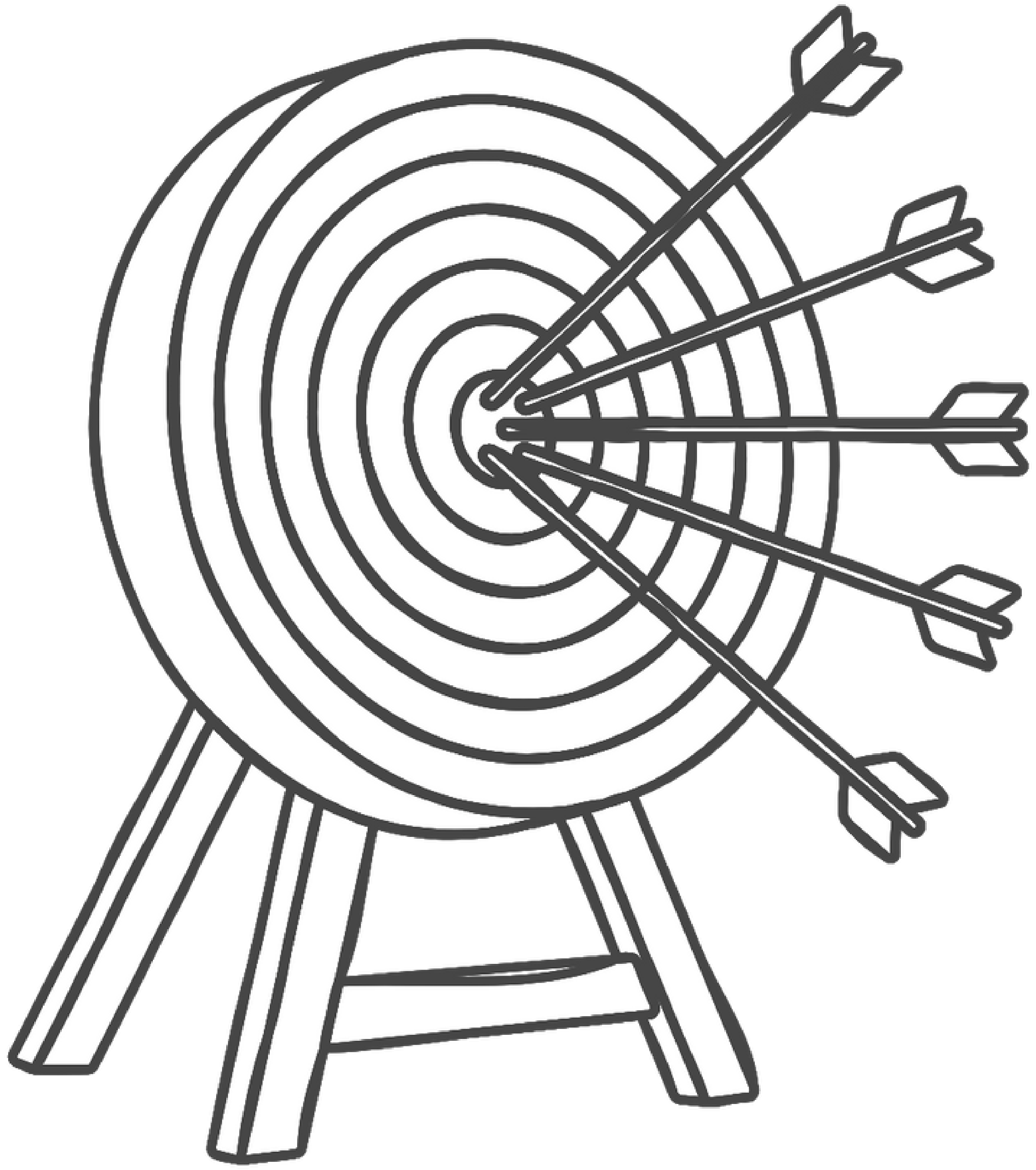
- **Detects anomalies across metrics, logs, and traces automatically.**
- **Identifies patterns that traditional monitoring rules miss.**
- **Prioritizes incidents based on severity and impact prediction.**

ML and LLMs in Observability

Large Language Models (LLMs) in Observability:

- **Summarize incidents in human-readable language.**
- **Assist in faster root cause analysis by correlating multiple signals.**
- **Predict potential future issues based on historical patterns.**

LLM Observability



LLM Observability

Core practice for model understanding

Monitoring Metrics

Tracking performance indicators

Performance Optimization

Enhancing model efficiency

Issue Identification

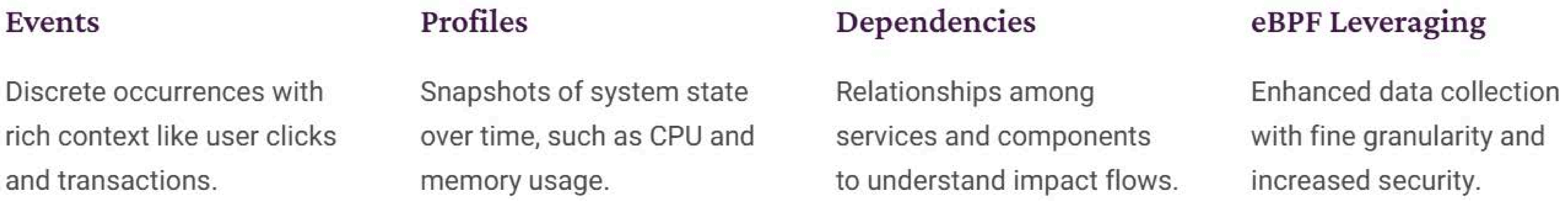
Spotting and addressing problems

Reliability Enhancement

Ensuring consistent model quality

ML and LLMs are transforming observability from passive monitoring to intelligent problem-solving.

Key Data Types in Observability 2.0



Events

Discrete occurrences with rich context like user clicks and transactions.

Profiles

Snapshots of system state over time, such as CPU and memory usage.

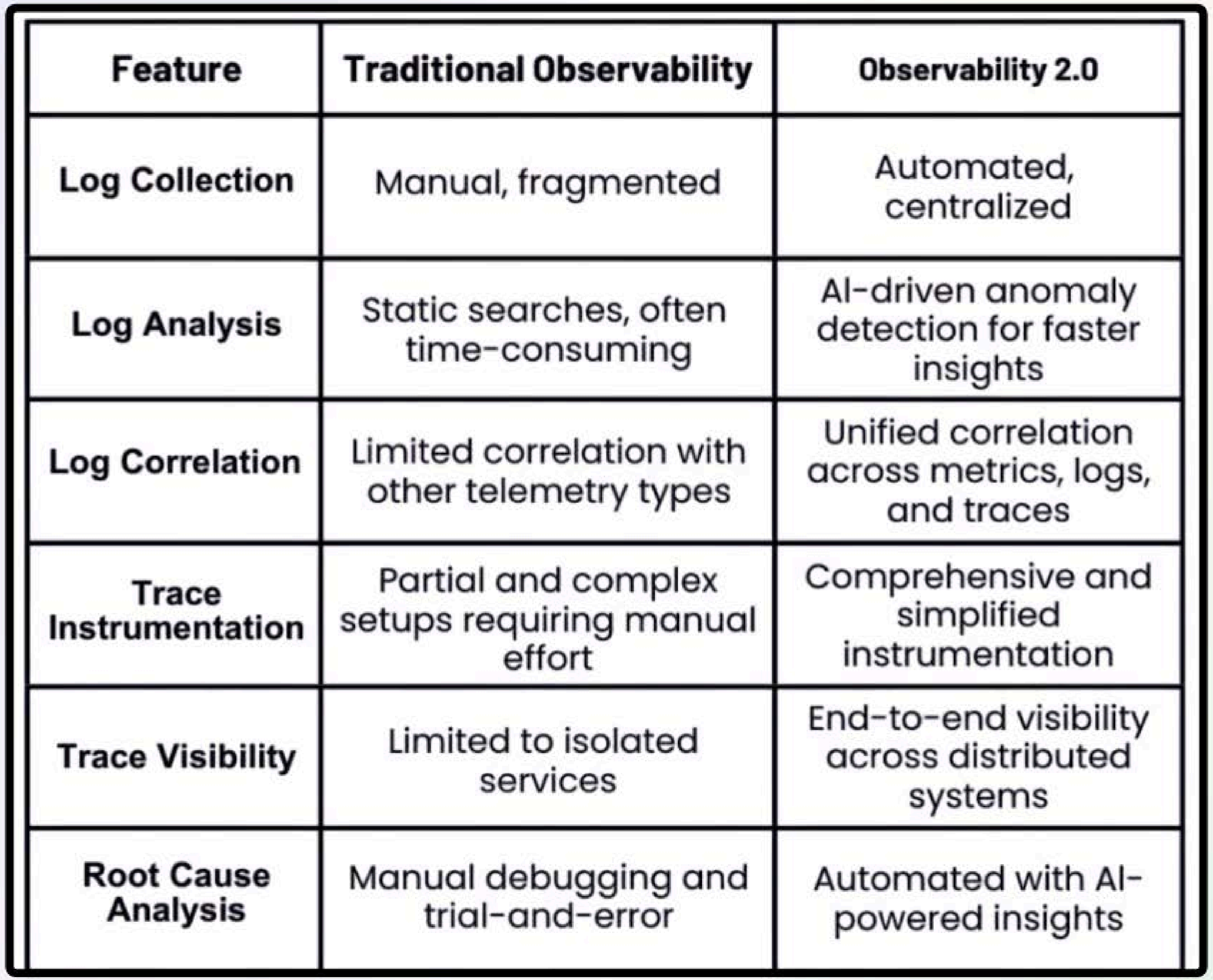
Dependencies

Relationships among services and components to understand impact flows.

eBPF Leveraging

Enhanced data collection with fine granularity and increased security.

Features of Observability 2.0



Feature	Traditional Observability	Observability 2.0
Log Collection	Manual, fragmented	Automated, centralized
Log Analysis	Static searches, often time-consuming	AI-driven anomaly detection for faster insights
Log Correlation	Limited correlation with other telemetry types	Unified correlation across metrics, logs, and traces
Trace Instrumentation	Partial and complex setups requiring manual effort	Comprehensive and simplified instrumentation
Trace Visibility	Limited to isolated services	End-to-end visibility across distributed systems
Root Cause Analysis	Manual debugging and trial-and-error	Automated with AI-powered insights

Observability Cost Explosion

The Problem:

- Massive telemetry growth leads to huge storage and processing costs.
- Storing every log, metric, and trace — even when much of it is irrelevant — wastes resources.
- Traditional observability models were not designed for cloud-native scale.

Observability Cost Explosion

Key Cost Drivers:

- High cardinality metrics.
- Verbose and repetitive logs.
- Over-sampling of traces.

Observability Cost Explosion

Result:

- Increased infrastructure spend.
- Slower query and analysis performance.
- Unmanageable noise in monitoring systems.

Without cost-optimized strategies, observability becomes financially unsustainable.

Smart Data Pipelines for Observability

- **What are Smart Data Pipelines?**

Systems that selectively process, filter, and enrich telemetry

For Example:

Middleware provides an Ingestion Control pipeline that helps to exclude excessive logs from being stored in your S3 bucket or storage.

Smart Data Pipelines for Observability

Key Strategies:

- Dynamic Sampling: Capture only a representative subset of data during normal operations.
- Filtering and Deduplication: Remove irrelevant or duplicate logs and metrics at the source.
- Aggregation: Summarize data points over time to retain trends without keeping every detail.
- Edge Processing: Perform initial data reduction close to the source.

Smart Data Pipelines for Observability

Benefits:

- Major reduction in storage and processing costs.
- Higher signal-to-noise ratio, enabling faster insights.
- Scalability to handle growing application ecosystems.

AI and the Future of Observability

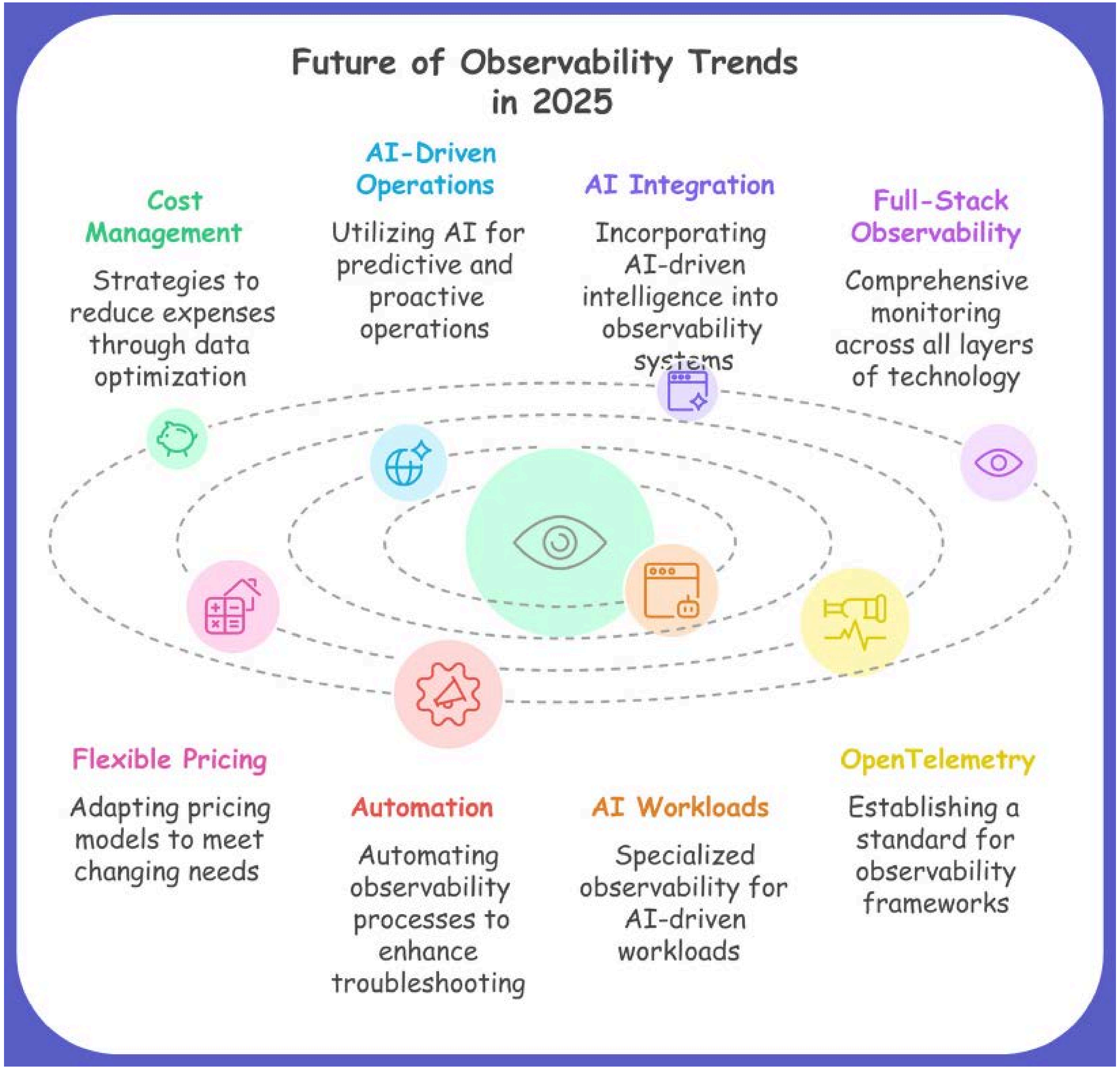
AI-Driven Observability Trends:

- **Predictive alerts based on trend analysis, not just thresholds.**
- **Self-healing systems that automatically trigger recovery actions.**
- **AI-assisted root cause analysis that highlights probable failure point**

AI and the Future of Observability

Emerging Innovations:

- Dynamic telemetry pipelines that adjust data flow based on system health.
- LLMs automatically generating incident reports and remediation suggestions.
- Autonomous observability — systems that monitor, diagnose, and fix themselves with minimal human intervention.



Future of Observability Trends in 2025

Cost Management

Strategies to reduce expenses through data optimization

AI-Driven Operations

Utilizing AI for predictive and proactive operations

AI Integration

Incorporating AI-driven intelligence into observability systems

Full-Stack Observability

Comprehensive monitoring across all layers of technology

Flexible Pricing

Adapting pricing models to meet changing needs

Automation

Automating observability processes to enhance troubleshooting

AI Workloads

Specialized observability for AI-driven workloads

OpenTelemetry

Establishing a standard for observability frameworks

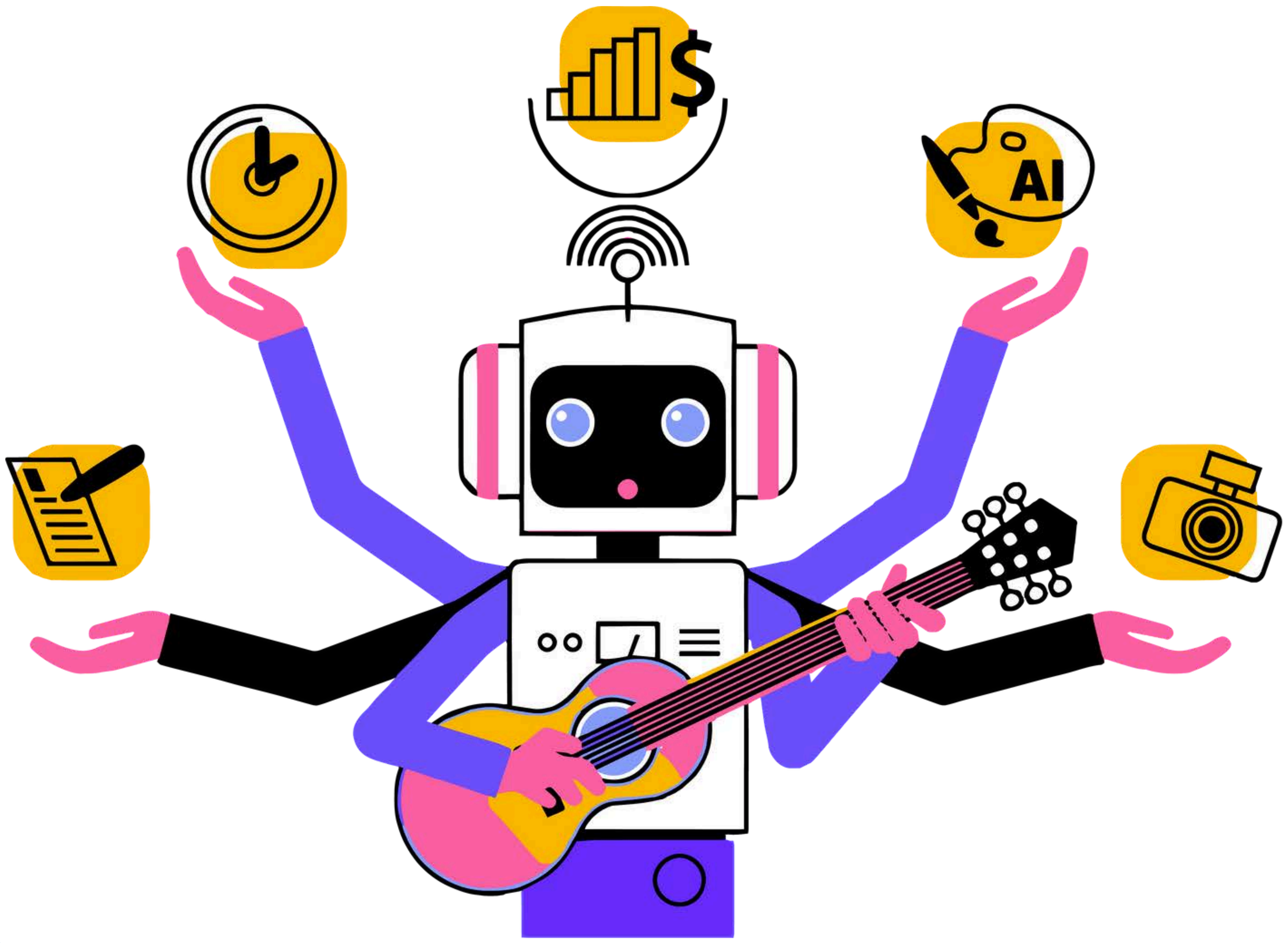
AI and the Future of Observability

Outcome:

- Faster recovery from incidents.
- Greater reliability and resilience at scale.
- Engineers focus on innovation instead of manual monitoring.

The future of observability is proactive, intelligent, and increasingly autonomous.

Detailed Blog on Observability 2.0



SCAN ME

Connect with me for any queries !

SCAN ME

Thank You