Observability as a Platform Contract Zero-Code Instrumentation and CI/CD-Enforced SLOs

By : Sumit Kaul

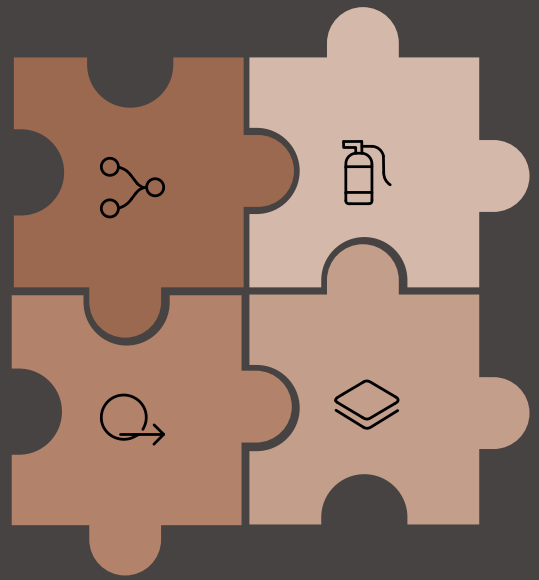
Staff Software Engineer, Payjoy

Conf42 DevOps 2026

The State of Observability Today

Manual Instrumentation

Reactive Debugging



Inconsistent Coverage

Team Silos

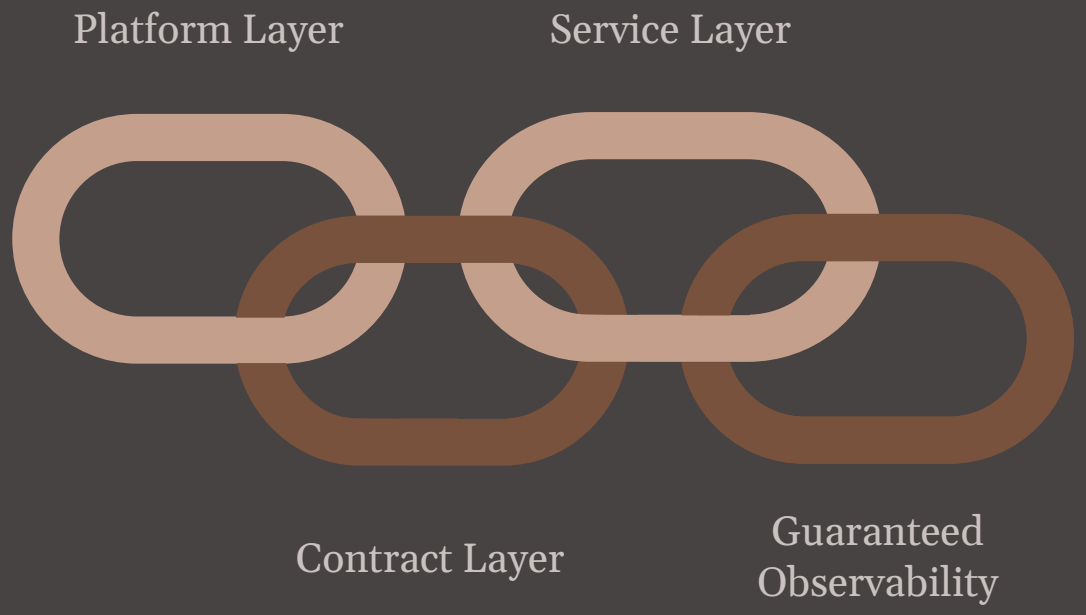
A New Model: Platform Contract

Elevate observability from opt-in heroics to an explicit platform responsibility. Transform it into a set of guarantees provided to every service in exchange for following the paved road.

This approach makes observability a **first-class platform capability** rather than an afterthought bolted on by individual teams.

- Standardised instrumentation patterns
- Consistent signal collection across services
- Enforceable reliability agreements
- Automated compliance verification

The Platform Contract Model



Services get comprehensive observability without writing instrumentation code. The platform handles collection, the contract defines expectations, and CI/CD enforces compliance.

Zero-Code Instrumentation Architecture

Automated Instrumentation

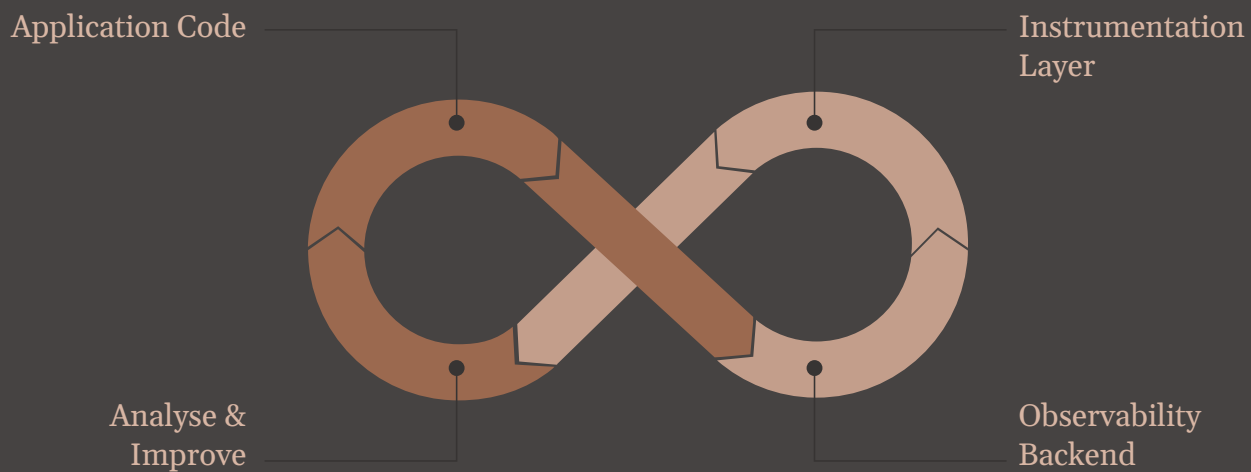
OpenTelemetry agents and framework configurations inject telemetry across languages and protocols without code changes.

Runtime Sidecars

Sidecar patterns automatically capture network traffic, database queries, and message interactions.

Base Container Images

Pre-configured images ship with baked-in observability components, ensuring consistent deployment.



Multi-Language, Multi-Cloud Reality

Implementing zero-code instrumentation across heterogeneous environments requires strategic planning.

1	2
Language Support Automatic instrumentation for Java, Python, Node.js, Go, and .NET via OpenTelemetry agents.	Cloud Agnostic Consistent deployment across AWS, Azure, and GCP using portable containers and standardized collectors.

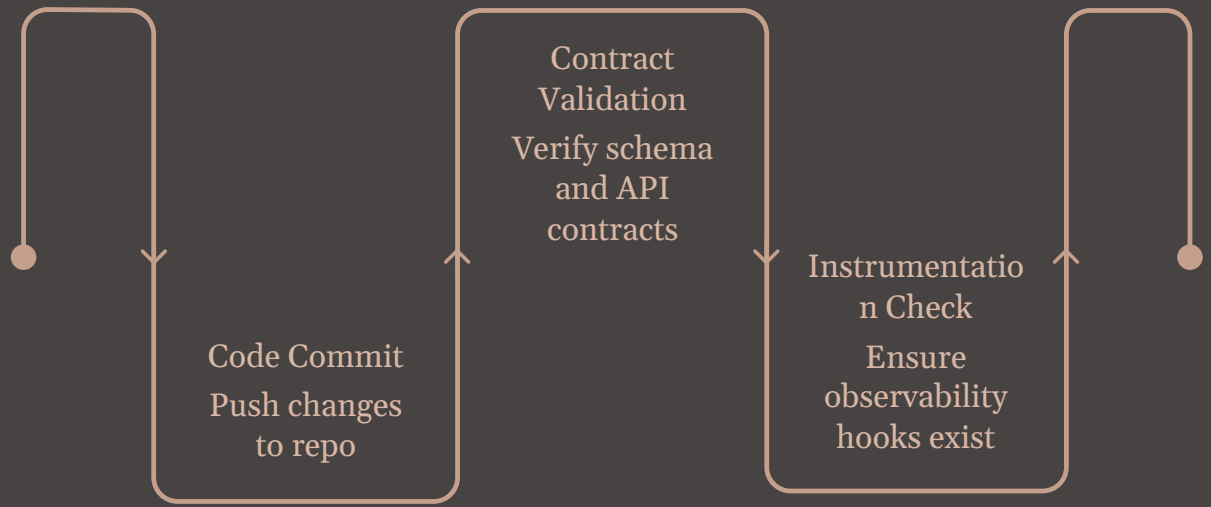
Successfully instrumented 200+ services across 3 clouds with 99.2% coverage.

Observability Contracts: First-Class Artifacts

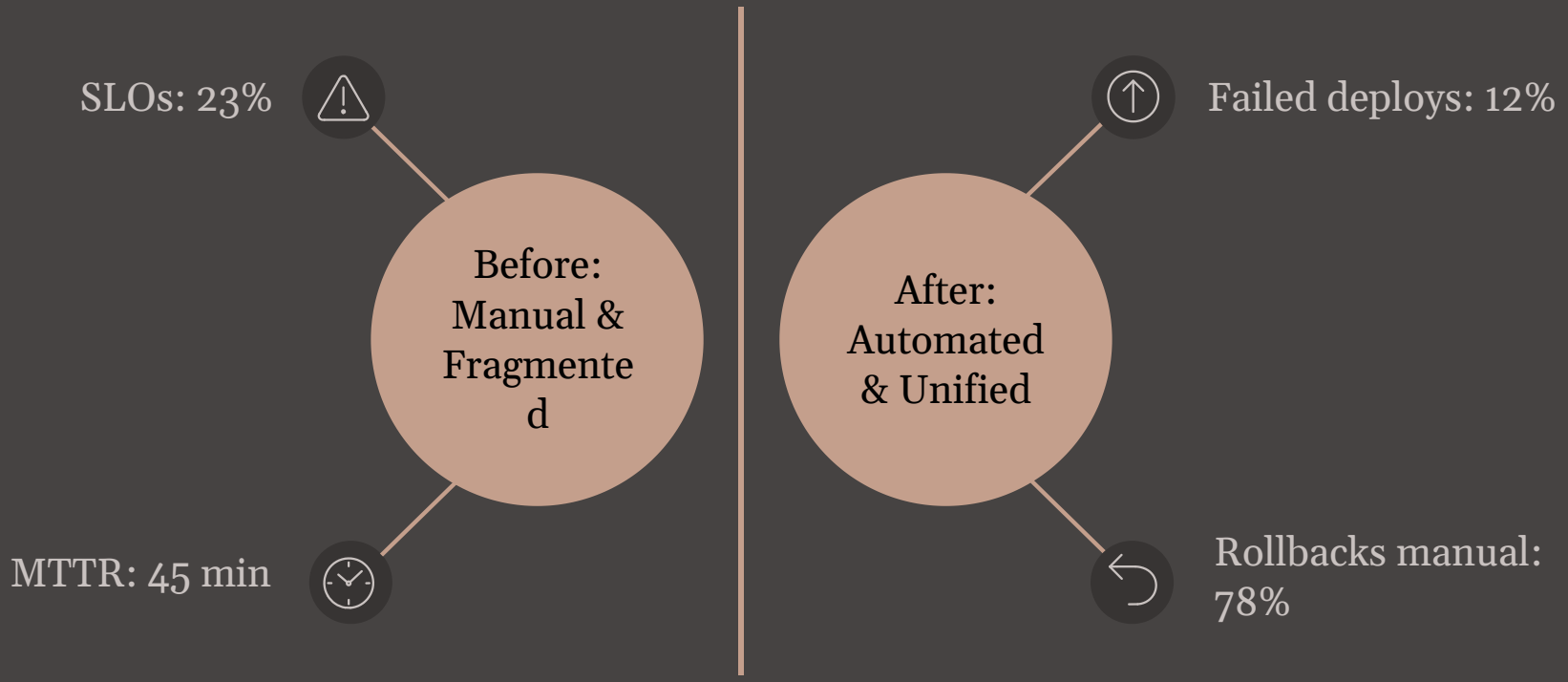
Observability contracts sit alongside manifests and pipelines as explicit, version-controlled agreements. They define the signals every service must emit and the reliability standards it must meet.

Contracts transform vague reliability goals into measurable, enforceable agreements that integrate directly into your deployment pipeline.

Contract Enforcement Through CI/CD



The Results Speak for Themselves



SLOs: From Vague Targets to Measurable Agreements

99.9%

Availability

Target encoded directly in contract

20-50ms

Latency SLO

P99 response time enforced

0.1%

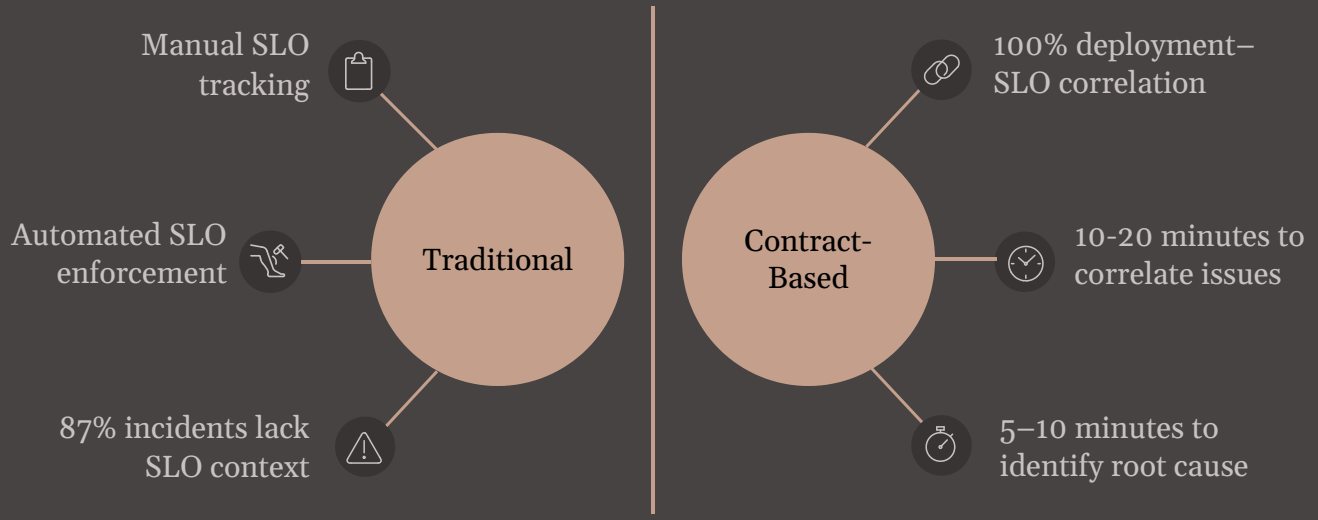
Error Budget

Remaining budget visible in real-time

73%

Reduced MTTR

After contract implementation



Release Automation Integration

Deployment Strategies Driven by Contracted SLOs

- **Baseline Capture**

  Record current SLI performance before deployment begins

- **Canary Deploy**

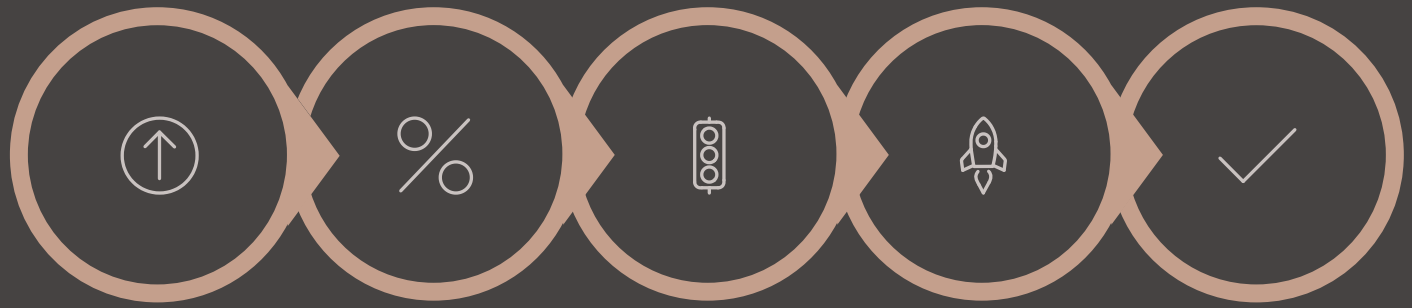
  Route small percentage of traffic to new version

- **SLO Evaluation**

  Compare canary metrics against baseline and burn rate policies

- **Automatic Decision**

  Promote on success, rollback on SLO violations



Accelerated Incident Response

Streamlined Resolution: Before & After

From Symptoms to Root Cause

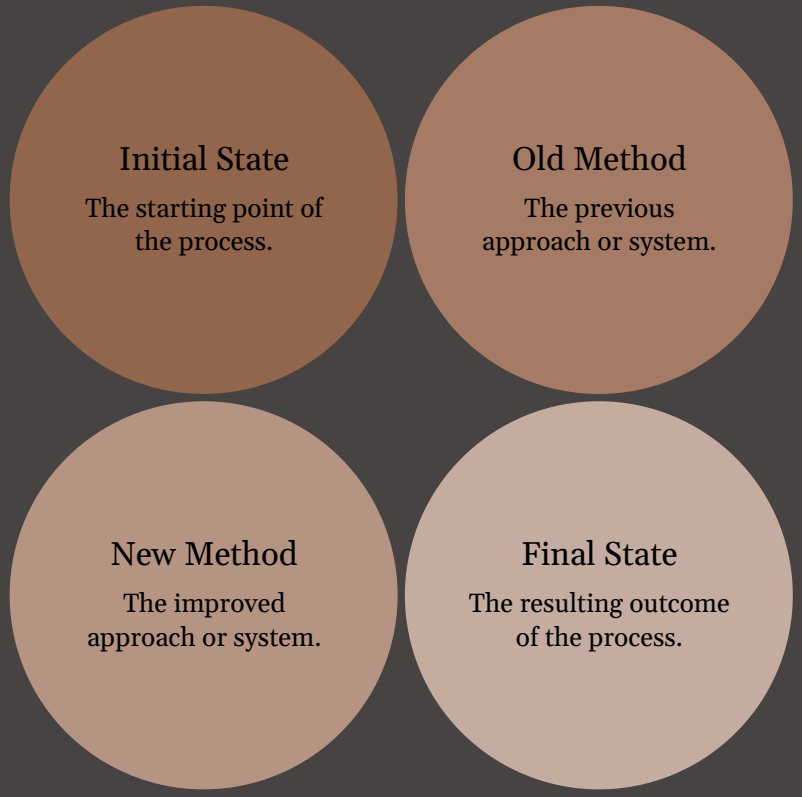
Unified workflows use existing traces and metrics, allowing rapid identification of problematic changes.

- Automatic correlation

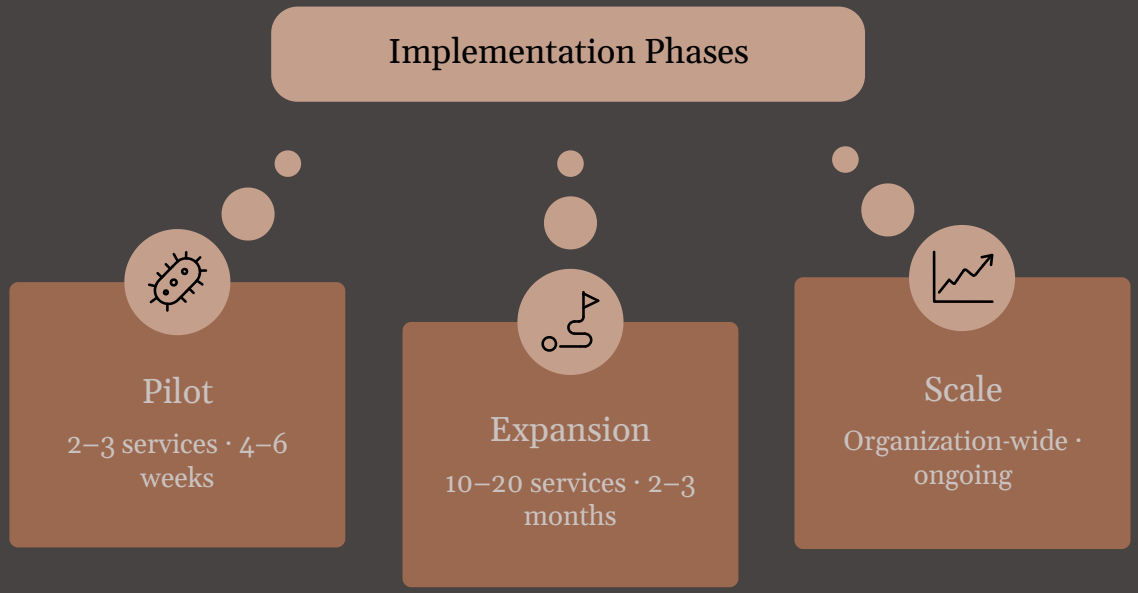
  Link degraded SLIs to specific deployments through version tags

- Trace analysis

  Drill into spans showing exactly where latency increased



Adoption Patterns That Work



- **Gradual Introduction**

  Start with pilot services and expand incrementally, avoiding delivery blocks.

- **Engineer Training**

  Educate teams on SLOs through workshops, documentation, and embedded platform engineers.

Successful adoption requires both technical integration and cultural support from platform teams.

Implementation Across Industries

Financial Services

Achieved 99.99% traceability across 2.3M daily transactions. Strict compliance requirements met through mandatory signal collection and audit trails. Contracts ensure every transaction is fully traceable.

Retail

Maintained 99.95% uptime during Black Friday with 12x normal traffic using error budget policies. E-commerce platforms use error budgets to balance feature velocity with customer experience during peak shopping periods.

Logistics

Instrumented 450+ microservices with zero code changes, reduced deployment time by 40%. Real-time tracking services benefit from automatic instrumentation across distributed microservices and mobile applications.

Your Blueprint for Success

Transform observability from fragmented tooling into a core platform capability that enables safe, fast, and accountable change.

01

Deploy Zero-Code Instrumentation

OpenTelemetry agents, base images

02

Define Contracts

SLO definitions, signal requirements

03

CI/CD Integration

Validation gates, enforcement

04

Production Monitoring

SLO tracking, error budgets

05

Continuous Improvement

Feedback loop

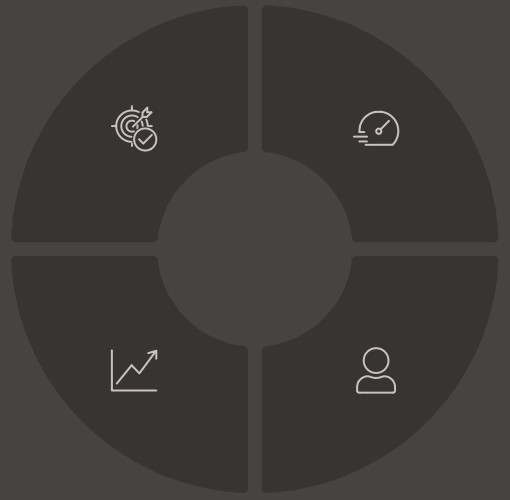
Enable Organisational Scale

Measurable Reliability

SLOs transform from aspirational targets into enforceable agreements

Safe Velocity

Deploy confidently with automated rollback based on objective metrics



Continuous Improvement

Error budgets drive data-informed reliability investments

Team Autonomy

Platform provides guarantees while teams maintain ownership

99.2%

Service Coverage

Automated instrumentation

73%

Faster MTTR

From 45 min to 12 min average

85%

Deployment Confidence

SLO-driven releases

40%

Faster Delivery

Reduced manual validation

Questions? Let's discuss how to implement observability contracts in your organisation.

Thank You!

Questions and Discussion..?

Sumit Kaul
https://www.linkedin.com/in/sumit-kaul-2a8b7237/