Zero Downtime ML Observability

Ensuring Reliability and Insights Without Interruptions

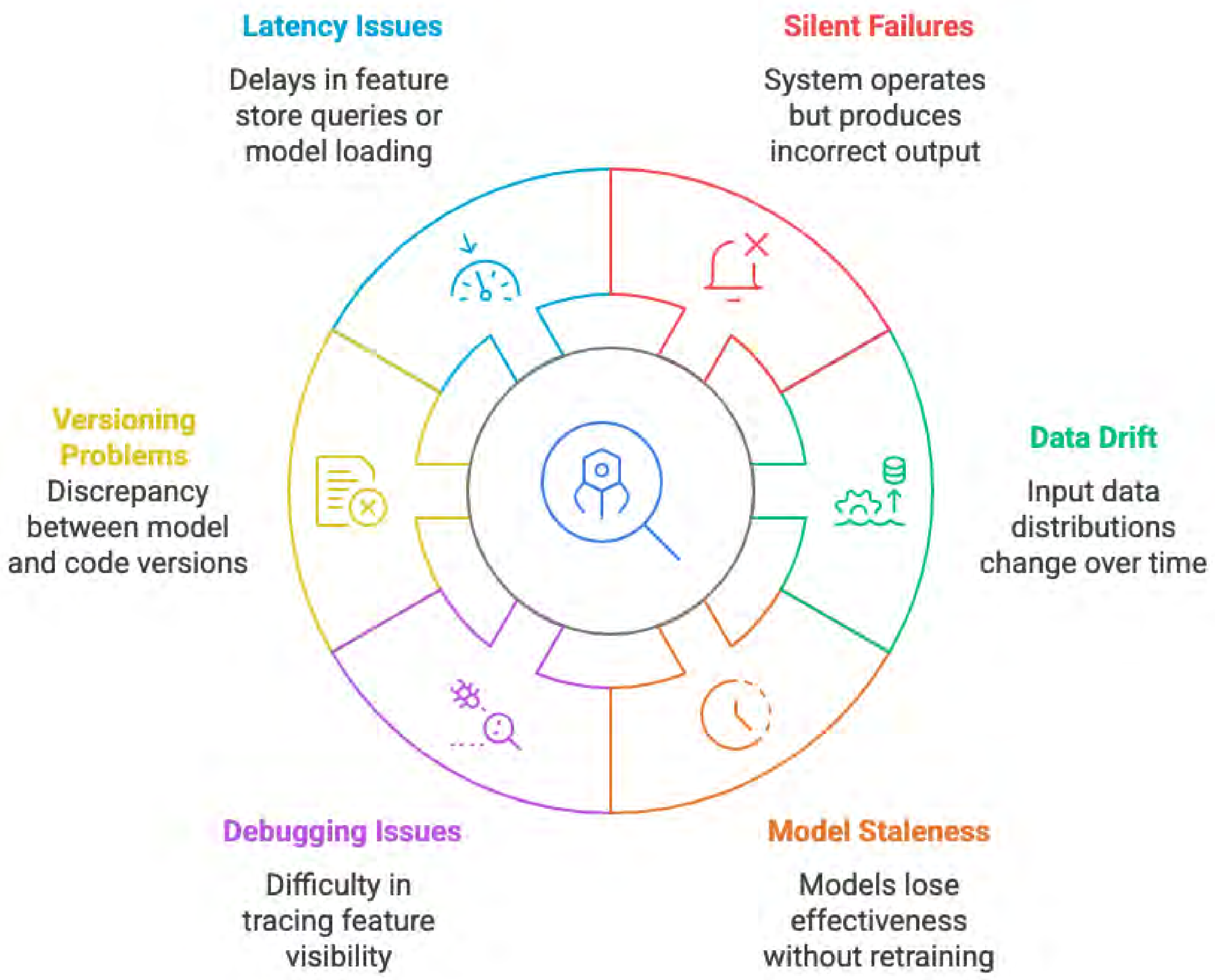
Payal Godhani
Principal Engineer – Oracle Cloud Infrastructure

The Two Worlds: ML vs SRE

ML Engineers	SREs
Build Models	Run Systems
Experimentation - Driven	SLA/SLO-driven
Focus on accuracy	Focus on uptime, latency
Offline evaluation	Real-time production metrics

Why Traditional Observability Isn’t Enough ?

- Logs, metrics, and traces tell us if a service is *up*, but not if a model is *right*.
- Errors may be silent—i.e., prediction is returned, but it's wrong or biased.
- Drift in data can degrade performance over time, even if infrastructure is stable.
- No standard SLOs for ML quality (accuracy, fairness, etc.)

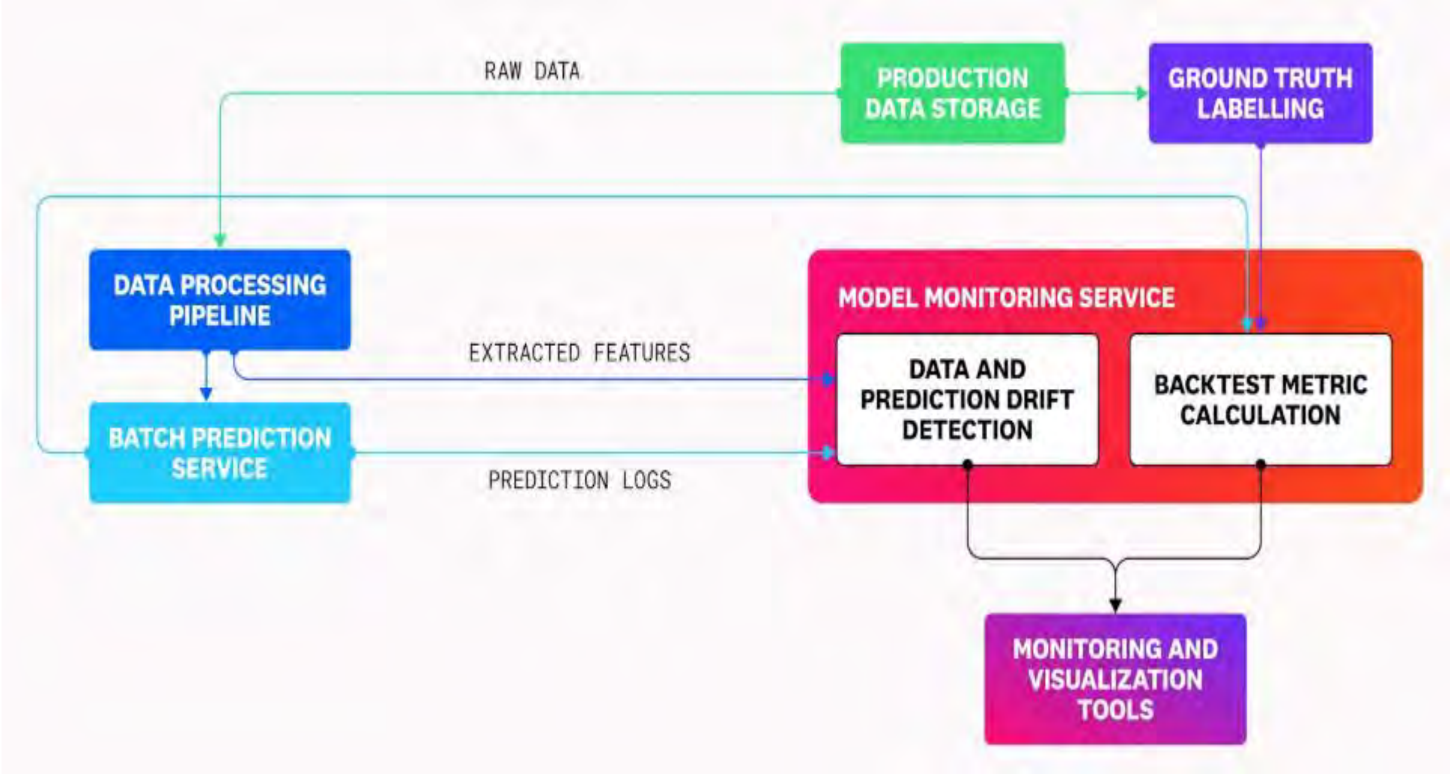


SRE Challenges in ML Systems

What is ML Observability ?

"ML Observability is the ability to monitor, debug, and understand ML systems in production."

System View: ML in Production

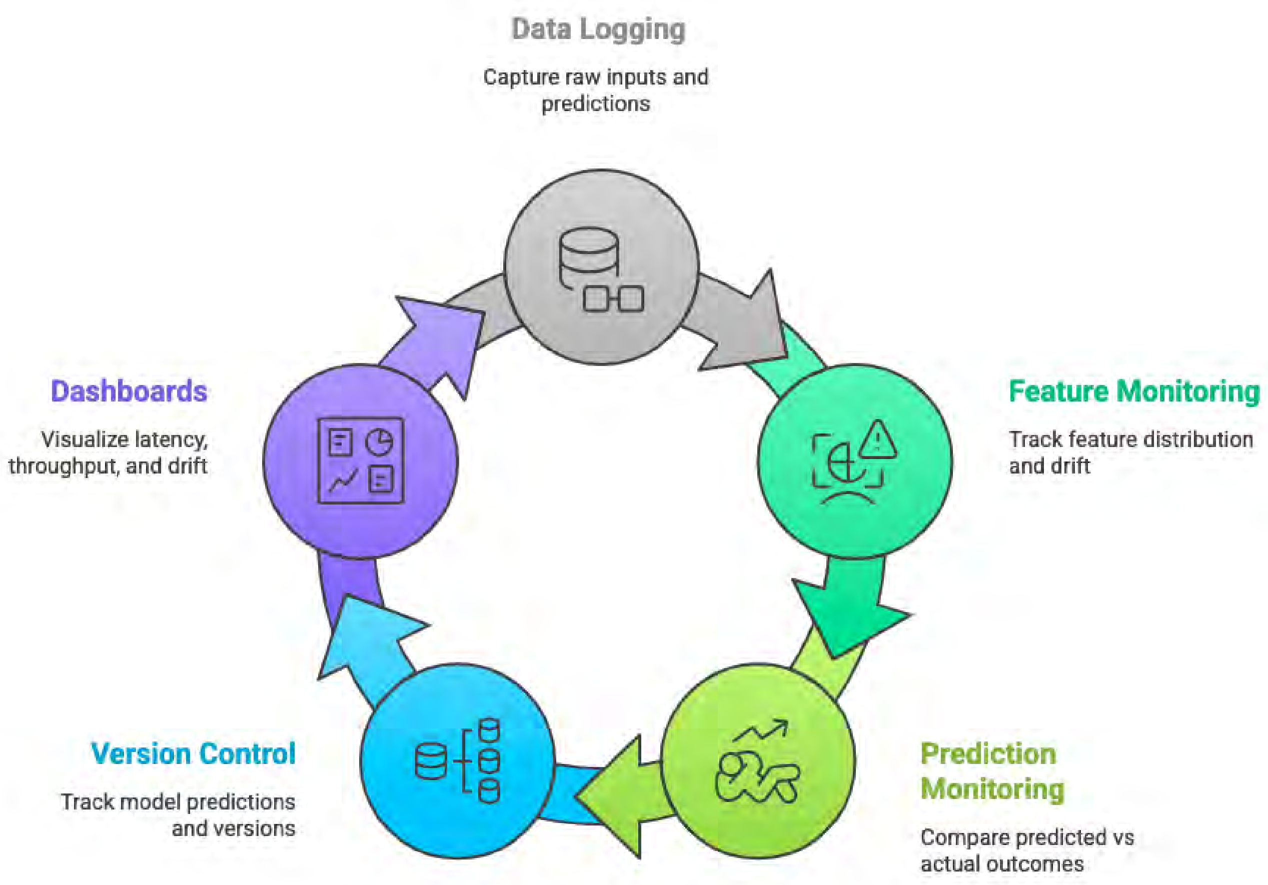


Model Evaluation
Average latency (by region)
RMSE (by version)
RUM
Page views
Checkouts
Funnel FROM Sofas VIEW ...TO Checkout ACTION
0% End-to-end Conversion
/department/sofas
433 sessions of 5.43K total
click on ADD TO CART
78.0 sessions
/cart
21.0 sessions
/checkout
6.00 sessions
click on CHECKOUT
0 sessions
18%
82% dropoff
27%
73% dropoff
29%
71% dropoff
0%
100% dropoff

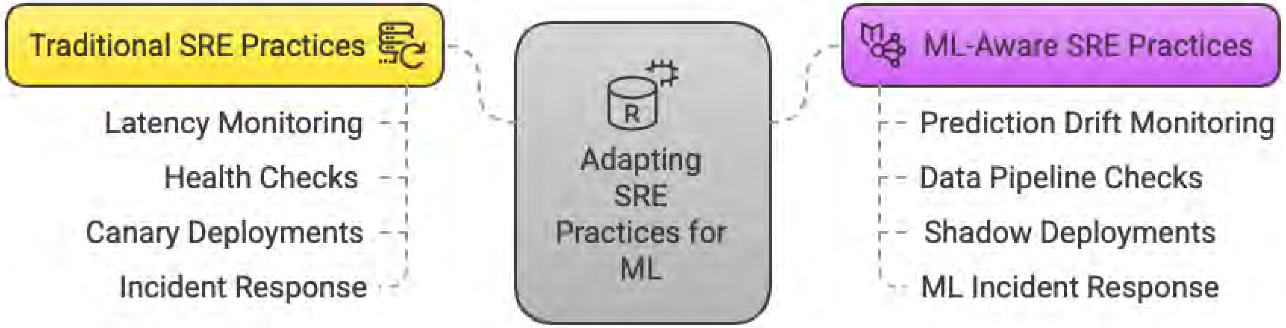
Monitoring Prediction and Data Drift

Drift Type	What It is	Example	Why it Happens
Prediction Drift	Change in the distribution of model outputs over time	Recommendation model starts suggesting only premium products	Changes in input data patterns, user behavior, or seasonal trends
Data Drift	Input feature distribution differs from training data	Surge in traffic from a new user demographic not seen during training	Business expansion, new product lines, marketing campaigns
Feature Drift	Shift in values of individual input features	“User location” feature shifts from mostly urban to suburban users	Change in customer base, regional promotion, data collection bias
Feature Attribution Drift	Change in feature importance across retrains	“Brand” becomes more important than “price” in product rankings	Frequent retraining, changing model weights, label noise in retrains

What Good ML Observability Looks Like



Adapting SRE Practices in ML



Tools for Detecting Drift

Evidently

WhyLabs

Alibi Detect

River

Vertex AI

Amazon SageMaker

Why Zero Downtime Matters ?

High availability for critical ML services

Continuous monitoring without halting predictions

Ensures confidence in deployments and real-time systems

Zero Downtime ML Observability

1

Use **mirrored traffic** for passive drift detection

2

Keep **observability async** to avoid inference path latency

3

Deploy **shadow models** to compare old vs new in prod

4

Design for **graceful degradation** when model fails

5

Make **observability part of your CI/CD** pipelines

Final Thoughts

“ML systems fail differently. They rot silently. They don’t always crash, but they decay. As SREs, we must evolve our observability mindset to include these nuances—and build systems that don’t just stay up, but stay smart.”