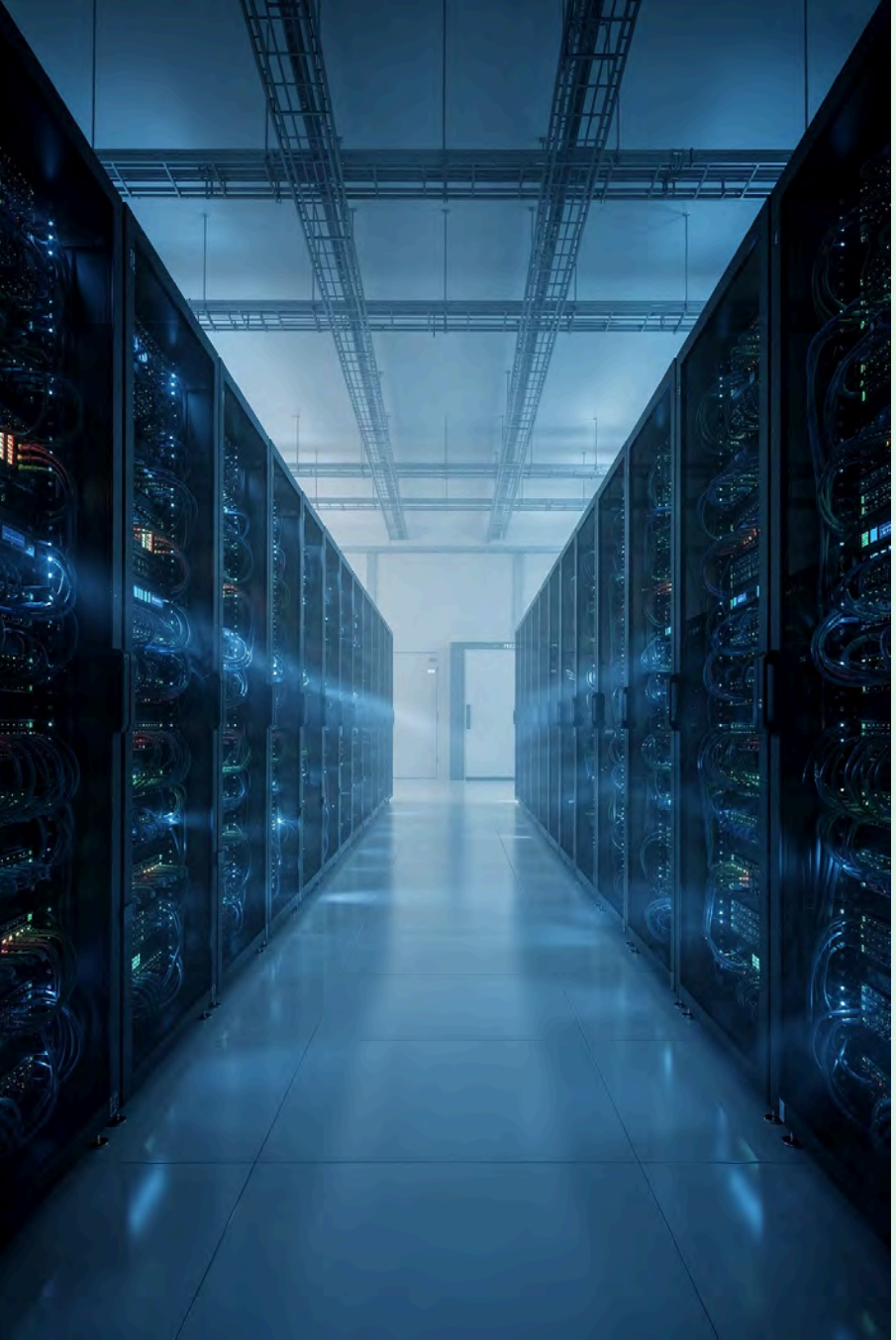


SRE for National-Scale Regulatory Reporting: HA, DR, and Audit-Ready Design

Bhargavaram Potharaju
University of Bridgeport
Conf42 Site Reliability Engineering

The Stakes: Why Regulatory Reporting Infrastructure Is Different

The Pressure Is Intensifying

- Regulatory datasets grow significantly with every reporting cycle
- Submission windows are shrinking, not expanding
- Regulators expect deterministic, auditable outcomes not best-effort delivery
- Legacy enterprise architectures were never designed for these constraints

The Core Problem: Where Legacy Architectures Fall Short

Most enterprise reporting platforms were built before national-scale regulatory demands became the norm. Today, they crack under five critical SRE requirements.

High Availability

No tolerance for unplanned downtime during regulatory windows

Peak Load Performance

Predictable throughput when submission volumes spike

Data Accuracy

Correctness at every transformation and validation stage

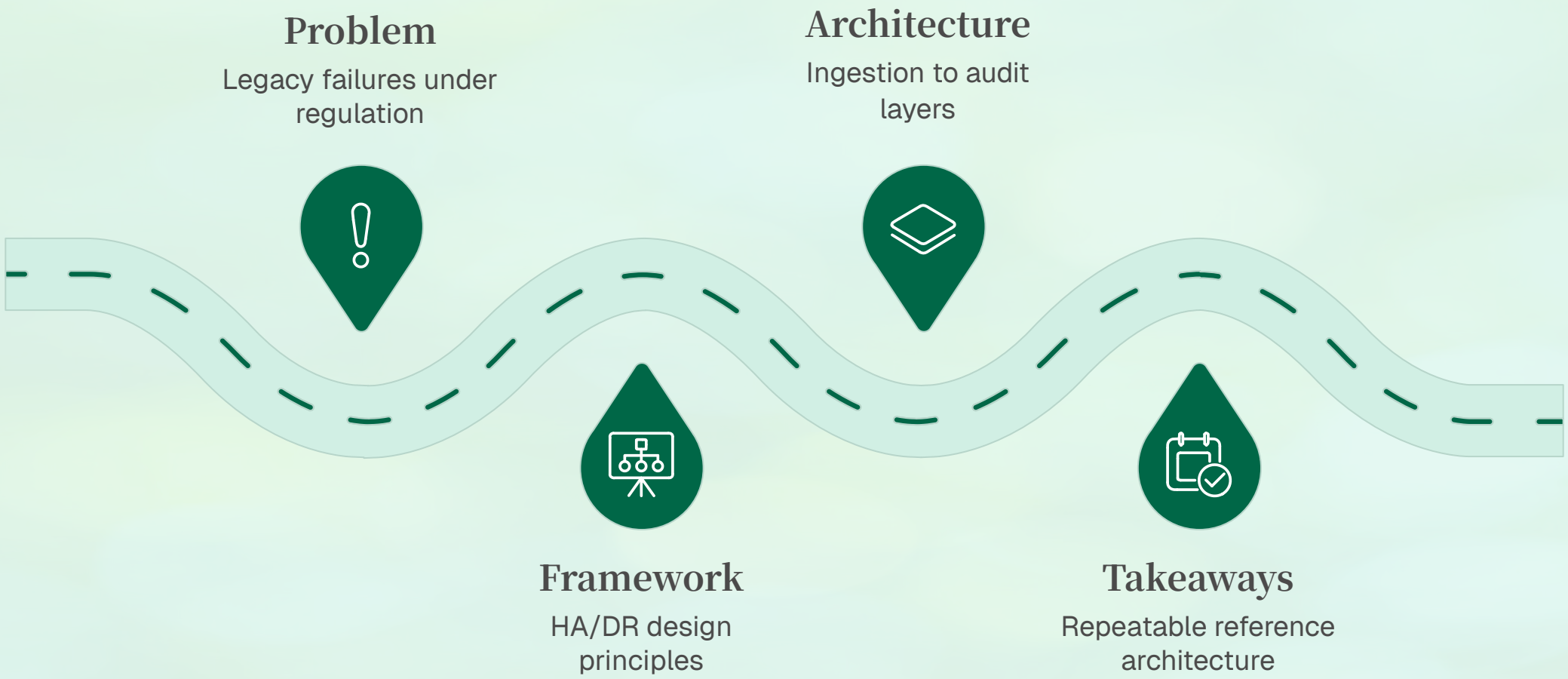
Auditable Outcomes

Full lineage and replay capability for every submission

Disaster Resilience

Recovery objectives that align with regulatory deadlines

Session Narrative: The Framework We'll Build Together



This session traces a deliberate path from understanding why existing platforms fail under regulatory pressure, through the design principles and architecture layers that resolve those failures, to a repeatable compliance-aware reference architecture your team can adopt.

Positioning SRE as a Compliance Driver

Infrastructure Engineering Is Not a Passive Concern

In regulatory ecosystems, SRE practices are directly responsible for compliance outcomes. When a reporting pipeline misses its submission window due to an infrastructure failure, the root cause is an engineering decision—or the absence of one. Reliability engineering must be a first-class design input, not an operational afterthought applied post-launch.

This reframing changes how we design, how we staff, and how we measure success. SLOs become compliance commitments. Error budgets reflect regulatory risk tolerance. On-call escalations map to submission deadlines.

- SLOs = Compliance Commitments
- Error Budgets = Regulatory Risk Tolerance
- Incident Escalation = Deadline Awareness
- Postmortems = Audit Evidence

The Reference Architecture: Design Constraints First

The framework is grounded in modern cloud and hybrid platform engineering. Three constraints are non-negotiable and treated as first-class design requirements from day one.

High Availability

Multi-zone active/active or active/passive topologies with no single points of failure across ingestion, processing, and submission layers

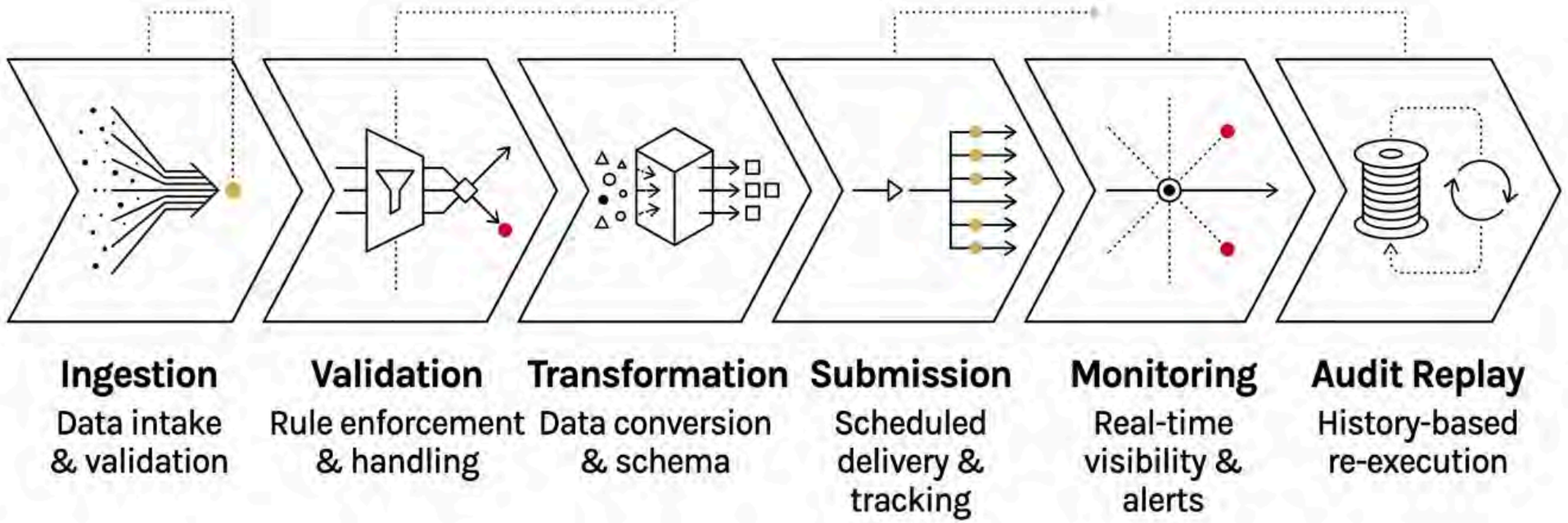
Disaster Recovery

Defined RTO and RPO targets aligned to regulatory submission deadlines, with tested failover paths and runbooks exercised under load

Audit Readiness

Immutable event logs, full data lineage, and deterministic replay capability for every submission available to regulators on demand

End-to-End Workload Stages: From Ingestion to Audit Replay



Each stage introduces distinct failure modes and must be designed with independent fault isolation, so that a failure in transformation does not cascade into a missed submission or corrupt audit trail.

High Availability Architecture: No Single Points of Failure

Topology Design Principles

- Active/active deployment across availability zones for ingestion and validation tiers
- Stateless processing workers with shared durable state in replicated storage
- Load-balanced submission endpoints with circuit breakers and back-pressure controls
- Health checks and automated instance replacement integrated into orchestration layer

During Peak Regulatory Windows

Peak conditions are the adversarial test of any HA design. Horizontal auto-scaling policies must be pre-warmed before scheduled submission windows open reactive scaling alone introduces latency that violates SLOs.

Disaster Recovery: Aligning RTO and RPO to Regulatory Deadlines

Generic DR targets are insufficient for regulatory reporting. RTO and RPO must be derived from the specific submission deadline constraints of each regulatory program.

1

Define Deadline-Aware Recovery Targets

Map each regulatory submission window to a maximum tolerable downtime. RTO must be shorter than the remaining window at any point of failure.

2

Architect for Tested Failover, Not Theoretical Failover

DR runbooks are only valid if exercised under realistic load. Chaos engineering and scheduled failover drills are non-negotiable.

3

Ensure Data Consistency at Failover Boundary

Replication lag at the moment of failover determines RPO realization. Synchronous replication or consensus protocols must be applied to submission-critical state.

4

Validate Recovery in Staging Under Peak Load

Failover procedures must be validated under simulated peak conditions not just baseline traffic to ensure they hold when it matters most.

Audit Readiness: Immutability, Lineage, and Replay

Three Technical Pillars

→ Immutable Event Logs

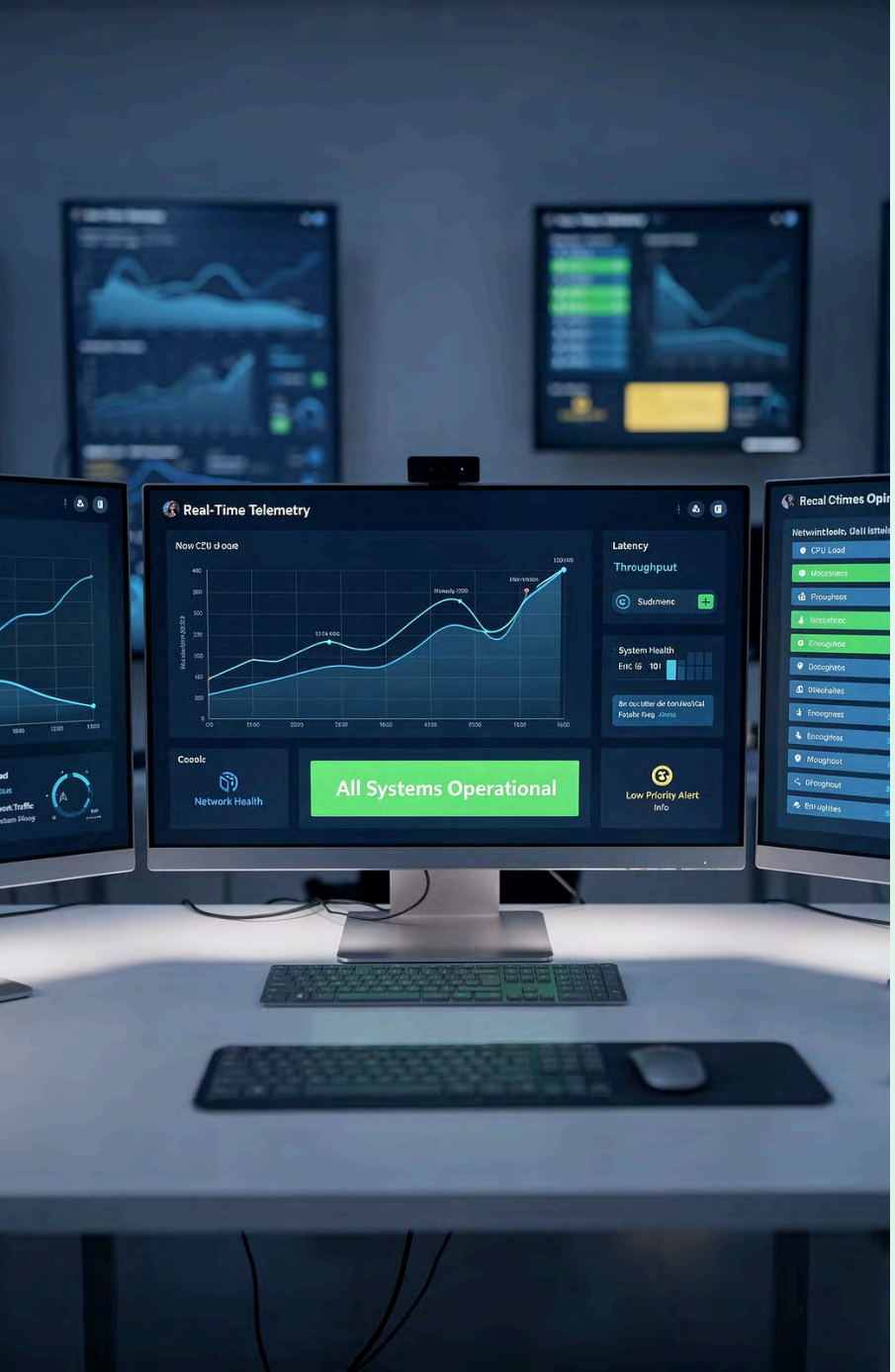
Append-only, tamper-evident records of every state transition across the pipeline stored separately from operational databases

→ Full Data Lineage

Traceable provenance for every data element from source ingestion to final submitted value, with transformation metadata preserved

→ Deterministic Replay

Ability to re-execute any historical submission from its original input state, producing bit-for-bit identical output for regulator review



Observability: Seeing the Pipeline as Regulators See the Outcome

Effective observability for regulatory pipelines goes beyond infrastructure metrics. You must instrument the business logic layer tracking record counts, validation pass rates, transformation durations, and submission acknowledgment latency as first-class signals.

Pipeline Telemetry

Per-stage throughput, latency percentiles, and queue depth metrics surfaced in real time not aggregated after the fact

Deadline-Aware Alerting

Alerts calibrated to submission window countdowns, not static thresholds escalating as remaining time shrinks relative to pipeline progress

Structured Logging for Audit

Machine-queryable, correlated log records that serve dual purpose: operational debugging and regulatory evidence retrieval

Safeguards: Preventing Silent Failures in Regulatory Pipelines

Silent failures—records dropped, validations skipped, acknowledgments missed—are uniquely dangerous in regulatory contexts because they may not surface until an audit or regulatory inquiry.

End-to-End Record Accounting

Every record ingested must be traceable to either a successful submission or an explicit rejection with documented reason no unaccounted drops permitted

Idempotent Processing

All transformation and submission operations must be safe to retry without producing duplicate submissions or corrupted state under failure recovery

Validation Gate Enforcement

Business rule validation must be a blocking gate, not an advisory check—records that fail validation must be quarantined, not passed downstream with flags

Submission Acknowledgment Verification

Regulatory endpoint acknowledgments must be explicitly confirmed and logged HTTP 200 is not sufficient without payload-level confirmation from the regulator

Cloud and Hybrid Platform Engineering Considerations

Cloud-Native Patterns That Apply

- Managed Kubernetes for stateless processing workloads with pod disruption budgets enforced
- Object storage with versioning and cross-region replication for audit log retention
- Managed message queues with dead-letter channels for ingestion fault isolation
- Infrastructure-as-code for reproducible, auditable environment provisioning

Hybrid Reality: On-Premises Constraints

Many regulatory platforms operate in hybrid topologies due to data residency requirements, network isolation mandates, or existing infrastructure investments. The reference architecture accommodates this through a consistent control plane abstraction that spans on-premises and cloud tiers without compromising HA or audit semantics.

Implementation Guidance: Where to Start

01

Baseline Your Current Failure Modes

Run a structured failure mode analysis across each pipeline stage before designing anything new. Know exactly where your current architecture would break under peak regulatory load.

02

Define Compliance-Aligned SLOs

Translate submission window constraints into concrete availability, latency, and accuracy SLOs. These become your engineering contracts, not aspirational targets.

03

Instrument Before You Optimize

Deploy end-to-end pipeline telemetry before making architectural changes. You cannot prioritize improvements without data on where time and failures are actually occurring.

04

Build and Test DR Runbooks Under Load

Write DR runbooks and validate them under simulated peak conditions in a staging environment that mirrors production topology and data volumes.

05

Harden Audit Trails Incrementally

Add immutable event logging at each pipeline stage incrementally, starting with the submission layer where regulatory scrutiny is highest, then working upstream.

Measurable Outcomes: What This Framework Delivers

When HA, DR, and audit readiness are treated as first-class design constraints rather than features added after initial launch regulatory reporting platforms demonstrate consistent operational improvements across four dimensions.

Processing Throughput

Measurable improvement in records processed per unit time during peak regulatory windows, driven by stateless horizontal scaling and queue-based decoupling

Cycle Latency

Reduced end-to-end reporting cycle time from data ingestion close to submission acknowledgment, enabling earlier confirmation and margin for exception handling

Data Accuracy

Consistently high accuracy at submission, achieved through blocking validation gates and idempotent processing that eliminates transformation-induced corruption

Unplanned Downtime

Minimal unplanned downtime during peak windows, supported by pre-warmed capacity, automated health-based recovery, and tested failover procedures

Key Takeaways for SREs and Engineering Leads

SRE is compliance infrastructure

In regulatory systems, reliability engineering decisions directly determine compliance outcomes. Treat them with corresponding urgency and rigor.

Design constraints before architecture choices

HA, DR, and audit readiness must be specified as hard constraints before selecting technologies or topology not fit in afterward.

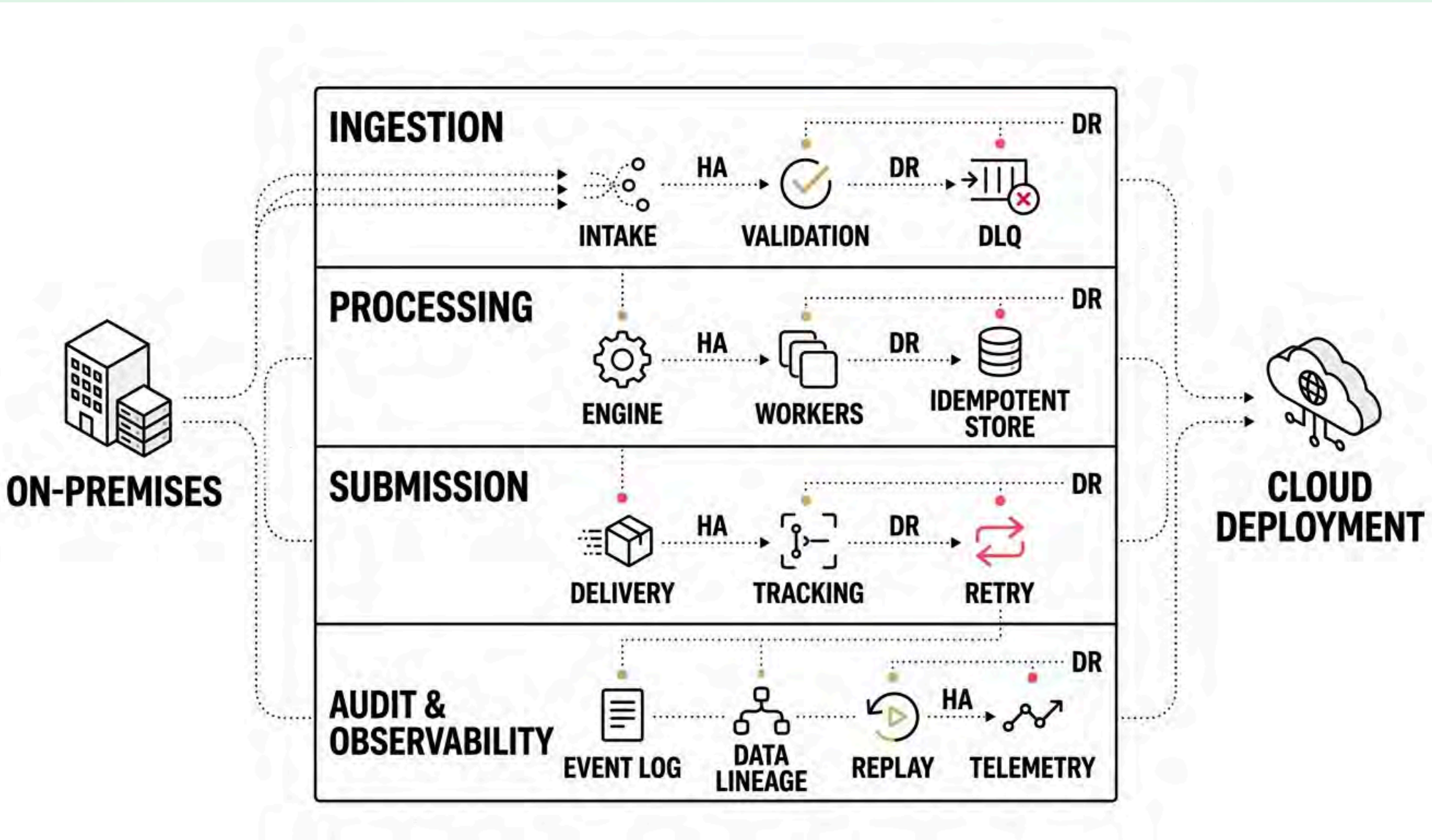
Silent failures are the highest-risk failure mode

Instrument for completeness and correctness, not just availability. A pipeline that silently drops records is more dangerous than one that fails loudly.

Test DR under realistic conditions or it doesn't count

Runbooks that have never been exercised under peak load are assumptions, not safeguards. Validate continuously.

The Repeatable Reference Architecture



This reference architecture is designed to be adopted incrementally by federal and enterprise regulatory teams. Each layer can be hardened independently, with the audit and observability tier deployable in parallel with existing infrastructure during a migration phase.

Thank You!—Let's Build Reliable Regulatory Infrastructure

SRE for National-Scale Regulatory Reporting: HA, DR, and Audit-Ready Design

Reliability engineering is the foundation on which regulatory compliance is built. When infrastructure is designed with submission deadlines, audit requirements, and disaster scenarios as primary constraints, reporting platforms become an institutional asset not a liability waiting to surface during a regulatory cycle.

Questions & Discussion

Architecture trade-offs, DR testing strategies, audit trail design, SLO definition for compliance systems

Connect After the Session

Continue the conversation on HA design patterns, observability tooling, and hybrid platform considerations for your specific regulatory context