Designing for Zero downtime

The Art of Zero-Downtime Gateway Migration

Author

Kalyan Inturi

Conference

Conference 42 2026

Zero Tolerance for Downtime

99.999%

Required Availability

For SaaS providers processing **1000 requests per sec or more**, "rip and replace" strategies are commercially unviable.

The Risk

Synthetic testing alone misses complex, undocumented edge cases accumulated over years in legacy systems.

The Solution

Automated reliability engineering and continuous validation are the only proven methods to prevent service interruptions.

The Strategy

Phased rollout strategies combined with shadow traffic validation enable seamless infrastructure transformation.

The Legacy Monolith: A Closer Look

Single Runtime Coupling

Request routing, authentication, and session management tightly bound in a single **Ruby/Sinatra** codebase.

The "Blast Radius" Problem

A minor change in routing logic could inadvertently break the entire authentication stack due to shared state.

Resource Constraints

Ruby's Global Interpreter Lock (GIL) prevented effective multi-core utilization, forcing inefficient uniform scaling.

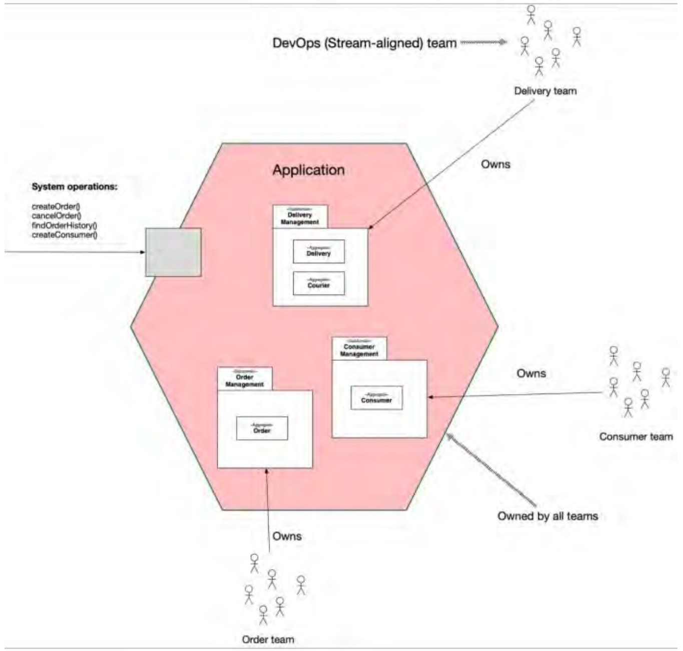


Figure 1: Tightly Coupled Legacy Architecture

The Legacy Gateway Bottleneck

Tight Coupling of Concerns

Routing, authentication, and session management combined in a single codebase. **Result:** Increased "blast radius" of minor updates.

Scalability Inefficiency

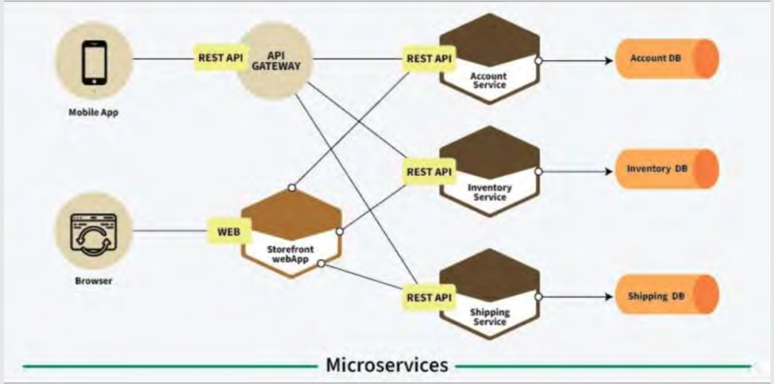
Monolithic design forced uniform scaling. Ruby's Global Interpreter Lock (GIL) constrained multi-core utilization.

Operational Risk

Deprecated language versions with scarce expertise created security patch compliance challenges.

Migration Objective: Transition to Kong Enterprise while maintaining continuous service availability.

Target Architecture: Decoupled Microservices



Migration Challenges & Validation

Undocumented Legacy Behaviors

Years of "tribal knowledge" and implicit assumptions created edge cases that synthetic test suites cannot reliably detect.

Performance Impact Risks

Standard traffic shadowing creates a **"Double Execution"** risk, where replayed requests might trigger duplicate backend transactions (e.g., double payments).

Coupled Authentication Logic

Extracting auth logic to independent microservices requires careful coordination to prevent authorization gaps during the transition.

Traditional testing methods cannot capture years of accumulated legacy behaviors.

The Required Solution

State-Mocking Shadow Framework

To validate logic using actual production traffic without impacting backend state.

Interceptor-Based Validation

The Challenge

Standard traffic shadowing risks **"double execution"** (e.g., duplicate payments).

The Solution

A custom **State-Mocking Shadow Framework** to validate logic without side effects.

01 Request Capture

Legacy Gateway generates UUID and asynchronously stores backend response in NoSQL (Redis/MongoDB).

02 Shadow Forwarding

Original request tagged with UUID is forwarded to Kong Gateway for parallel processing.

03 Backend Interceptor

Kong processes request normally but routes to specialized Interceptor Service instead of live backend.

04 State Mocking

Interceptor retrieves stored response by UUID and returns to Kong, simulating complete lifecycle.

05 Automated Analysis

Background process compares downstream payloads and client responses to flag regressions.

Eight-Phase Implementation Framework

1-2

Planning & Provisioning

Requirements gathering and parallel infrastructure provisioning using **Infrastructure as Code (IaC)**.

3

Shadow Framework Construction

Deployment of lightweight proxy layer and NoSQL store. Progressive traffic sampling starting with small percentages.

4

Authentication Offloading

Extracting auth logic to independent microservices with **OIDC verification**.

5

Automated Testing

Testing across millions of requests with **behavioral diffing** and edge case validation.

6-7

Incremental Rollout

Traffic shifting using feature flags (**1% → 5% → 100%**) with real-time metrics monitoring.

8

Full Cutover & Retirement

Final SLO compliance verification, full traffic migration, and decommissioning of the legacy system.

Shadow Traffic Findings

Validation Insights

Downstream Request Verification

Compared request payloads sent to backend by Kong vs. Legacy Gateway. Validated transformation plugins and routing logic correctness.

Client Response Verification

Compared final responses sent to clients. Detected missing headers, incorrect status codes, and payload format discrepancies.

Edge Case Discovery

Years of "tribal knowledge" and hidden production behaviors were revealed only through actual traffic patterns, missing from synthetic tests.

Execution Strategy

Progressive Sampling

Started with small traffic percentage to avoid resource overuse, gradually increasing based on validation confidence.

Data-Driven Velocity

Observed system behavior determined traffic shift pace, enabling adaptive migration speed rather than fixed deadlines.

Operational Impact & Benefits

0

Incidents

Customer-facing disruptions

20%

Cost Reduction

Infrastructure efficiency

SOC 2

Compliance

Vendor-supported security

Operational Efficiency

Automated deployment pipelines eliminated manual errors and freed engineering capacity for **high-value feature development**.

System Reliability

Reduced Mean Time to Recovery (MTTR) through **automated rollback capabilities** to known-good configurations.

Future Readiness

Multi-tenant capabilities position the infrastructure to accommodate business growth without requiring additional migration initiatives.

Unlocking 20% Cost Efficiency

Independent Scaling

Authentication workloads scale autonomously. We no longer scale the heavy monolithic stack just for lightweight auth checks.

Resource Optimization

Eliminating the **Ruby GIL bottleneck** allowed for significantly higher concurrency and CPU utilization per instance.

Reduced Operational Overhead

Vendor-supported security patches and automated pipelines replaced expensive manual maintenance and custom compliance efforts.

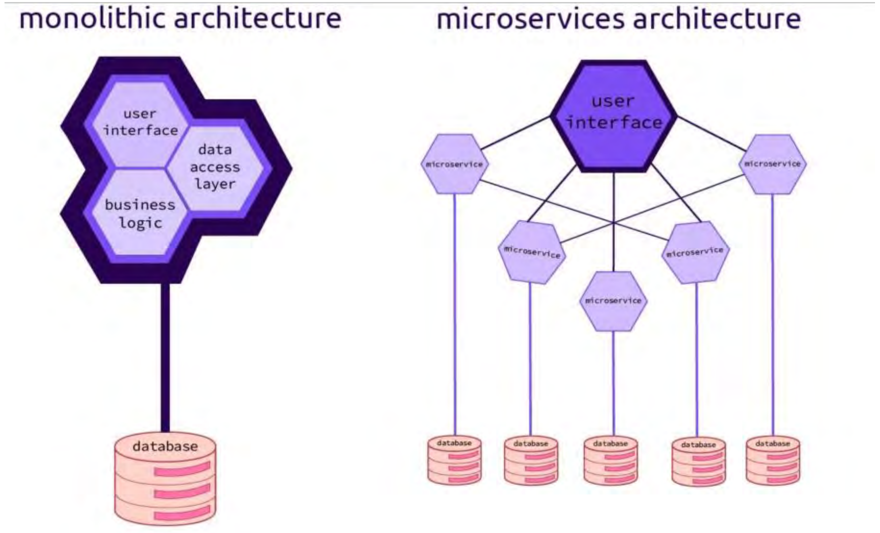


Figure 4: Efficiency Gains from Decoupling

Architectural Benefits

Authentication Independence

Extracted auth logic scales autonomously without constraining overall system capacity or routing performance.

Multi-Tenant Capabilities

Modern gateway accommodates future business growth and acquisition integrations without additional migration initiatives.

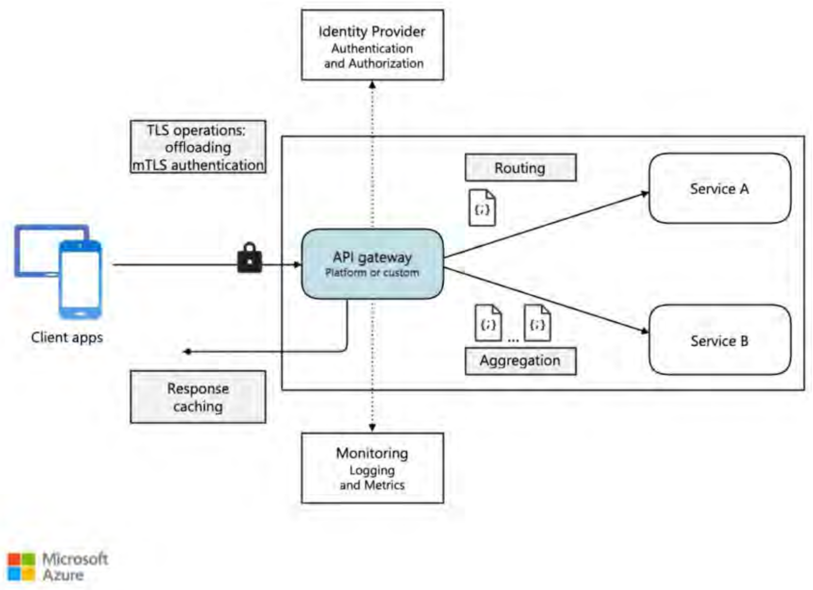
Compliance Simplification

Vendor-provided documentation replaces custom attestation efforts, streamlining SOC 2 and security audits.

Engineering Velocity

Automated testing gates enable confident deployments, freeing capacity previously consumed by manual coordination.

Figure 3: Decoupled Gateway Architecture



Key Success Factors

01 **Shadow Traffic Validation**

State-mocking framework validated production behaviors without "double execution" risks.

02 **Incremental Rollout**

Feature flags and progressive traffic shifting (1% → 5% → 100%) enabled adaptive migration velocity.

03 **Infrastructure as Code**

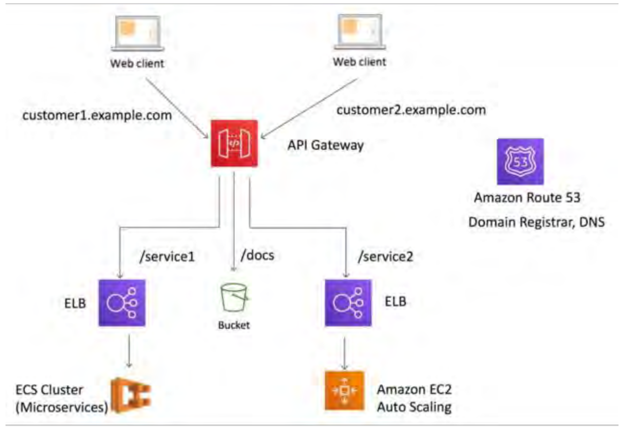
IaC practices ensured environment consistency and enabled rapid rollback capabilities.

04 **Automated Analysis**

Continuous comparison of millions of requests detected regressions before customer impact.

05 **Phased Implementation**

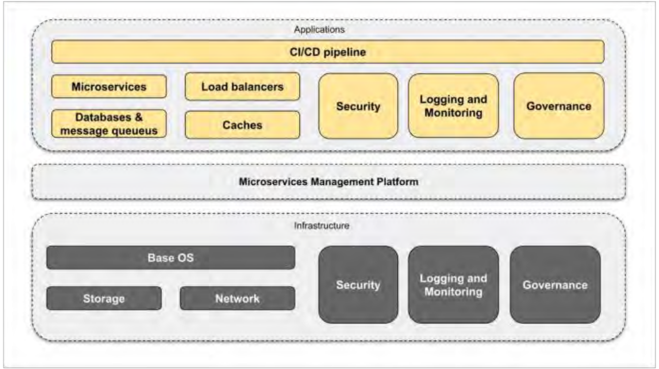
Sequential stages built validation confidence while maintaining production stability throughout transition.



Traffic Routing & Control Plane

Lessons Learned

Reference Blueprint: Microservices Management Platform



Universal Applicability

These architectural patterns serve as a reference blueprint for any organization transitioning from **monolithic architectures** to distributed microservices.

Validation Over Testing

For complex legacy systems, **production traffic validation** proves significantly more effective than synthetic testing at detecting edge cases.

Risk Management

The **"Store-and-Forward"** pattern is essential to decouple execution, preventing double-transaction risks during live validation.

Conclusion

Validated Methodology

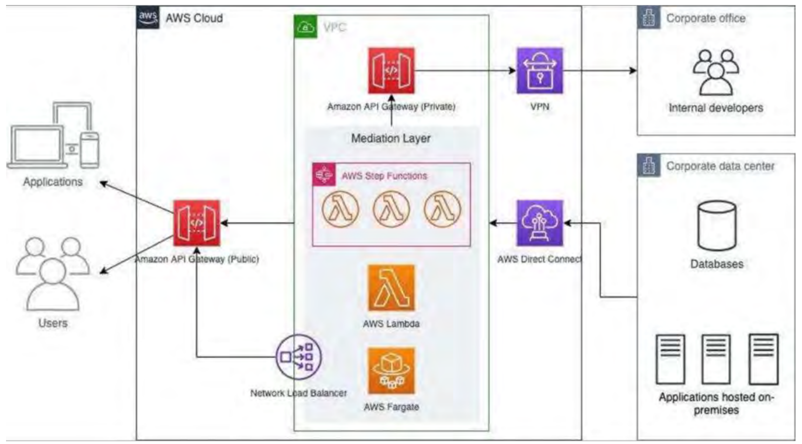
Phased rollout strategies combined with shadow traffic validation enable seamless infrastructure transformation.

Business Impact

Achieved zero downtime, >20% cost reduction, and enhanced security posture through modernization.

Future Directions

Further investigation into automated validation frameworks for distributed system migrations.



Modernized Infrastructure Blueprint

"Investing in **automated reliability engineering** and incremental deployment strategies is the proven path to successful infrastructure modernization."

Thank You

Presented by

Kalyan Inturi

Conference

Conference 42