Building Resilient Healthcare-Mobility Platforms: Engineering Epic-Rideshare Integration at Scale

Platform engineering for healthcare demands unprecedented reliability, security, and interoperability. This presentation explores the technical architecture behind integrating Epic EHR systems with ride-sharing platforms, demonstrating modern platform engineering solutions for complex healthcare logistics challenges.

Sai Sharan Reddy Mittapelly

Jawaharlal Nehru Technological University

Agenda

Technical Architecture Overview

Platform scale, integration patterns, and core components

Engineering Challenges

API flows, fault tolerance, and HIPAA-compliant data pipelines

Microservices Implementation

Event-driven architecture, data management, and orchestration

Security & Observability

Zero-trust architecture, monitoring, and healthcare-specific SLAs

MLOps in Healthcare

Feature engineering, model serving, and compliance

Platform Scale: The Numbers

6.5M

Daily Patient Record Exchanges

Processed securely across systems

2,400+

Health Systems

Connected through the platform

<3s

API Latency

For SMART on FHIR integrations

Our platform handles massive scale while maintaining the performance and reliability standards essential for critical healthcare operations.

Epic EHR + Rideshare: The Integration Challenge

Epic SMART on FHIR

Standards-based framework requiring strict compliance with healthcare data protocols and authentication flows

Rideshare APIs

Consumer-oriented transportation platforms with different security models and operational expectations

Integration Layer

Our platform bridges these worlds with bidirectional data flows while maintaining compliance and reliability

Epic's SMART on FHIR framework establishes a standardized way for third-party applications to integrate with EHR data while maintaining security and compliance. Our platform extends this model to the transportation domain.

Core Engineering Challenges

Bidirectional API Flow Management

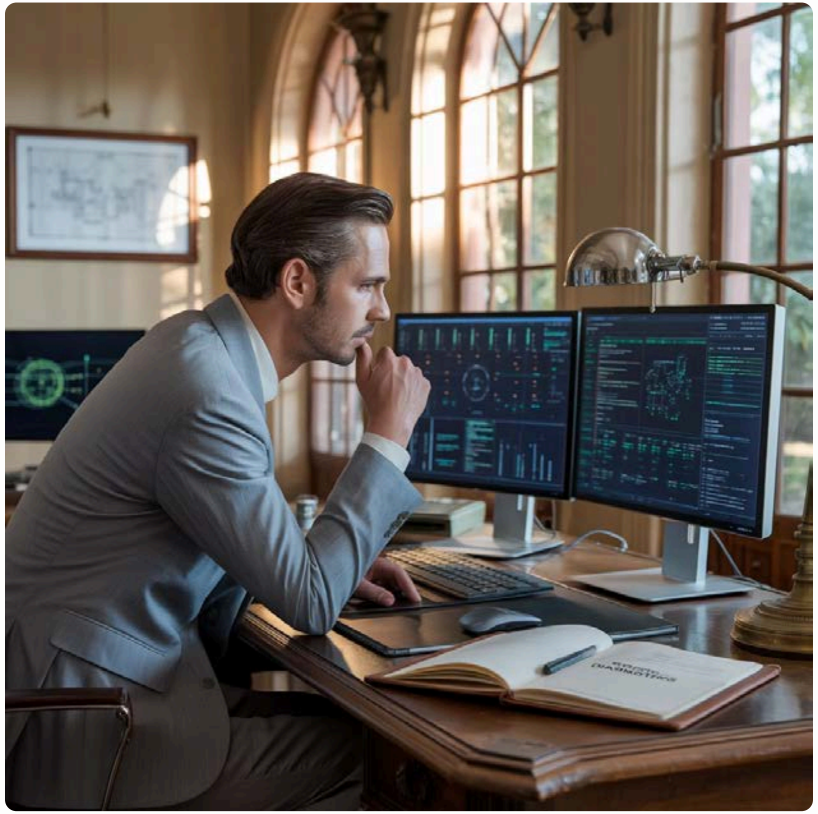
Synchronizing data between Epic's clinical workflows and rideshare platforms with different rate limits, authentication models, and response expectations

Fault Tolerance Implementation

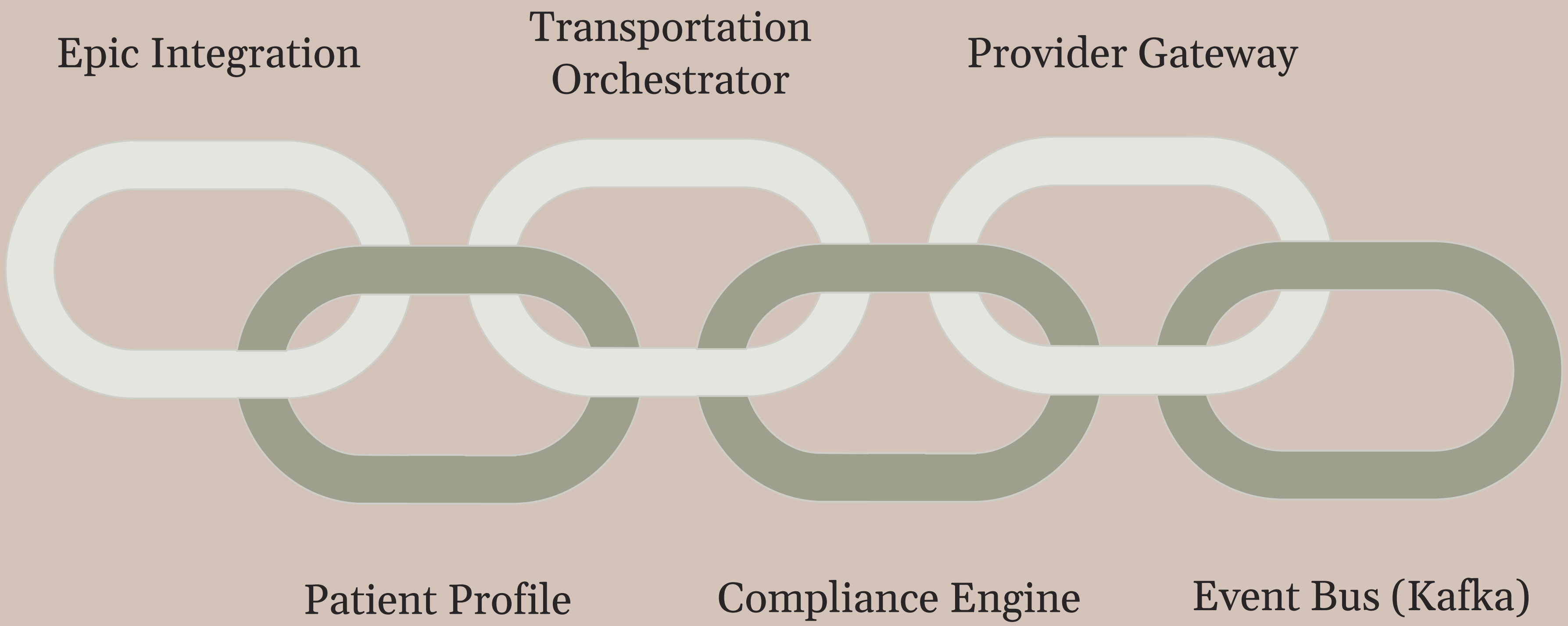
Circuit breakers and fallback mechanisms ensure patient transportation isn't disrupted even when third-party services experience degradation

HIPAA-Compliant Data Pipelines

Processing 47+ variables per transportation request while maintaining patient privacy, appropriate data sharing, and complete audit trails



Microservices Architecture



Epic Integration

Transportation Orchestrator

Provider Gateway

Patient Profile

Compliance Engine

Event Bus (Kafka)

Our event-driven architecture decouples components while maintaining a consistent view of patient transportation needs across the entire platform. This approach enables independent scaling, targeted resilience strategies, and domain-specific security controls.

Event-Driven Data Management

Kafka for Real-Time Events

- Transportation request state changes
- Appointment updates from Epic
- Driver location and ETA updates
- Compliance events and audit trails

Redis for Performance

- Patient transportation profiles
- Frequently accessed clinical context
- Temporary session data

PostgreSQL for Transactions

- Complete ride history
- Billing and reconciliation data
- Compliance documentation
- Configuration management

⊗ Data residency and regional compliance requirements often necessitate multi-region deployments with appropriate data segregation and replication strategies.

Kubernetes Orchestration & SRE Practices

1

Multi-Region Deployment

Distributed across 6 AWS regions to ensure proximity to health systems and regulatory compliance with data residency requirements

2

Auto-Scaling

Horizontal pod autoscaling based on appointment volumes, with predictive scaling using historical patterns for clinic schedules

3

Automated Failover

Active-active configuration with automated regional failover ensuring 99.9% uptime even during regional outages

4

Canary Deployments

Healthcare-specific deployment strategies that minimize risk by routing non-urgent transportation requests to new versions first

Security Engineering: Zero-Trust Architecture

Data Protection

- AES-256 encryption for all data in transit
- Field-level encryption for PHI at rest
- Tokenized patient identifiers to minimize PII exposure

Authentication & Authorization

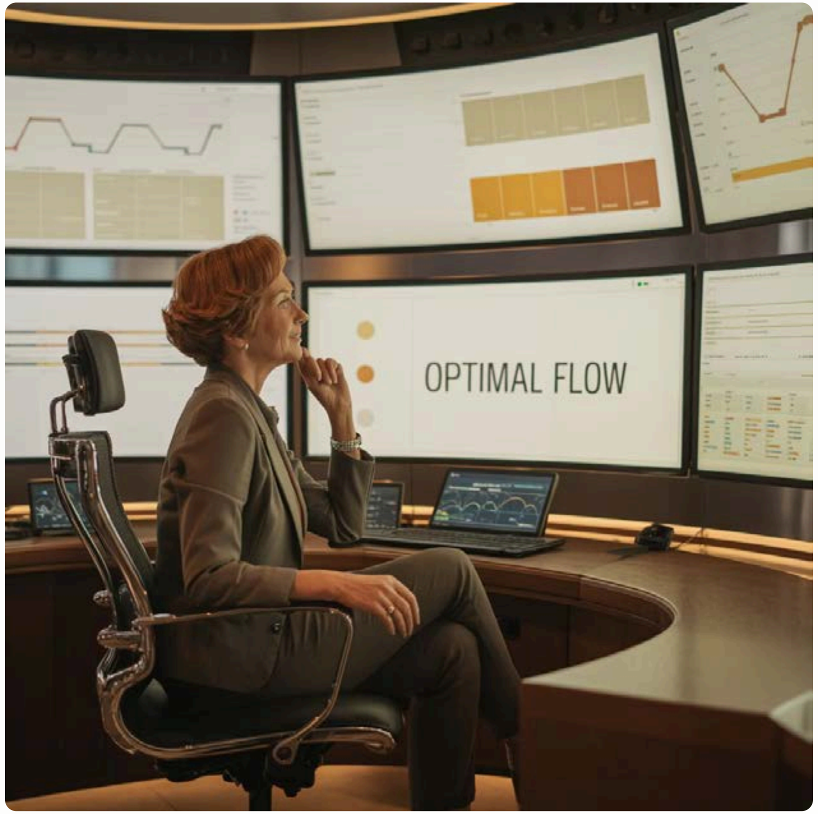
- OAuth 2.0 with PKCE for Epic integration
- SMART on FHIR scopes limiting data access
- Short-lived service tokens with automatic rotation

Network Security

- mTLS between all services
- Private VPC connectivity to Epic instances
- WAF and API gateway protection layers

ⓘ Healthcare security requires defense in depth. Our platform implements verification at every level, assuming no component is inherently trusted. This zero-trust approach minimizes the blast radius of any potential breach.

Observability: Healthcare-Specific Requirements



Distributed Tracing

End-to-end visibility across Epic, our platform, and rideshare services to quickly identify bottlenecks and failures

Clinical vs. Technical Alerting

Sophisticated alerting that differentiates between platform issues and clinical urgencies, with appropriate escalation paths

Healthcare SLA Metrics

Custom metrics for tracking ride timeliness relative to appointment times, patient wait times, and completion rates

MLOps Challenges in Healthcare Transportation

Data Collection

Feature Engineering

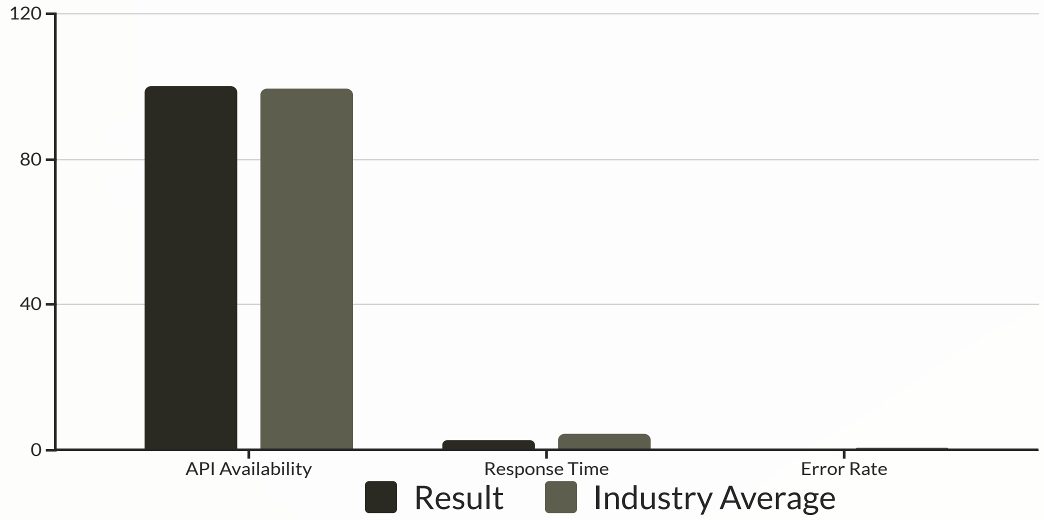
Model Development

Compliant Deployment

Continuous Monitoring

Our ML pipeline processes real-time patient transportation requests while handling the unique challenges of healthcare data: incomplete records, regulatory constraints on algorithm transparency, and the critical nature of accuracy when patient care is at stake.

Platform Reliability Metrics: Engineering Impact



Real-World Performance

- Successfully handled 10x traffic spikes during weather emergencies
- Processing over 15 million transportation events monthly
- Maintaining comprehensive audit trails for healthcare compliance
- Reduced patient no-shows by 42% through reliable transportation

Our platform's reliability directly impacts patient care outcomes. Engineering excellence translates to better healthcare access and improved clinical results.

Key Takeaways for Platform Engineers

Design APIs for Clinical Workflows

Healthcare APIs must accommodate clinical decision processes, not just technical requirements. Understanding the domain is critical.

Implement Healthcare-Compliant CI/CD

Validation, approval gates, and audit trails need to be baked into the deployment pipeline, not added as afterthoughts.

Manage Sensitive Data Across Systems

Create data governance frameworks that track sensitive information flows throughout the entire platform lifecycle.

Align Monitoring with Clinical Requirements

Build observability that speaks both technical and clinical languages, with appropriate context for different stakeholders.

Platform engineering principles can adapt to highly regulated industries while maintaining velocity and reliability. The key is designing with compliance as a first-class concern, not an afterthought.