

# Improving IIS HL7 Web Service Performance in New York City

---

**AIRA 2019 National Meeting**

**August 13, 2019**

**Breakout Session 1D**

*Angel Aponte, Michael Suralik, Surabhi Midha,*

*Chasmandeep Bring, Emily Martinez, Melissa Mickle-Hope, and Monica Sull*

# Outline

- **Goal**

- Describe CIR performance and reliability improvements to handle the growing demand for services from the Citywide Immunization Registry (CIR)

- **Background**

- Citywide Immunization Registry
- Technical aspects of the improvements

- **Methods/Implementations**

- Change software and database architecture
- Database migration

- **Results**

- **Lessons learned**

# Citywide Immunization Registry (CIR) Background

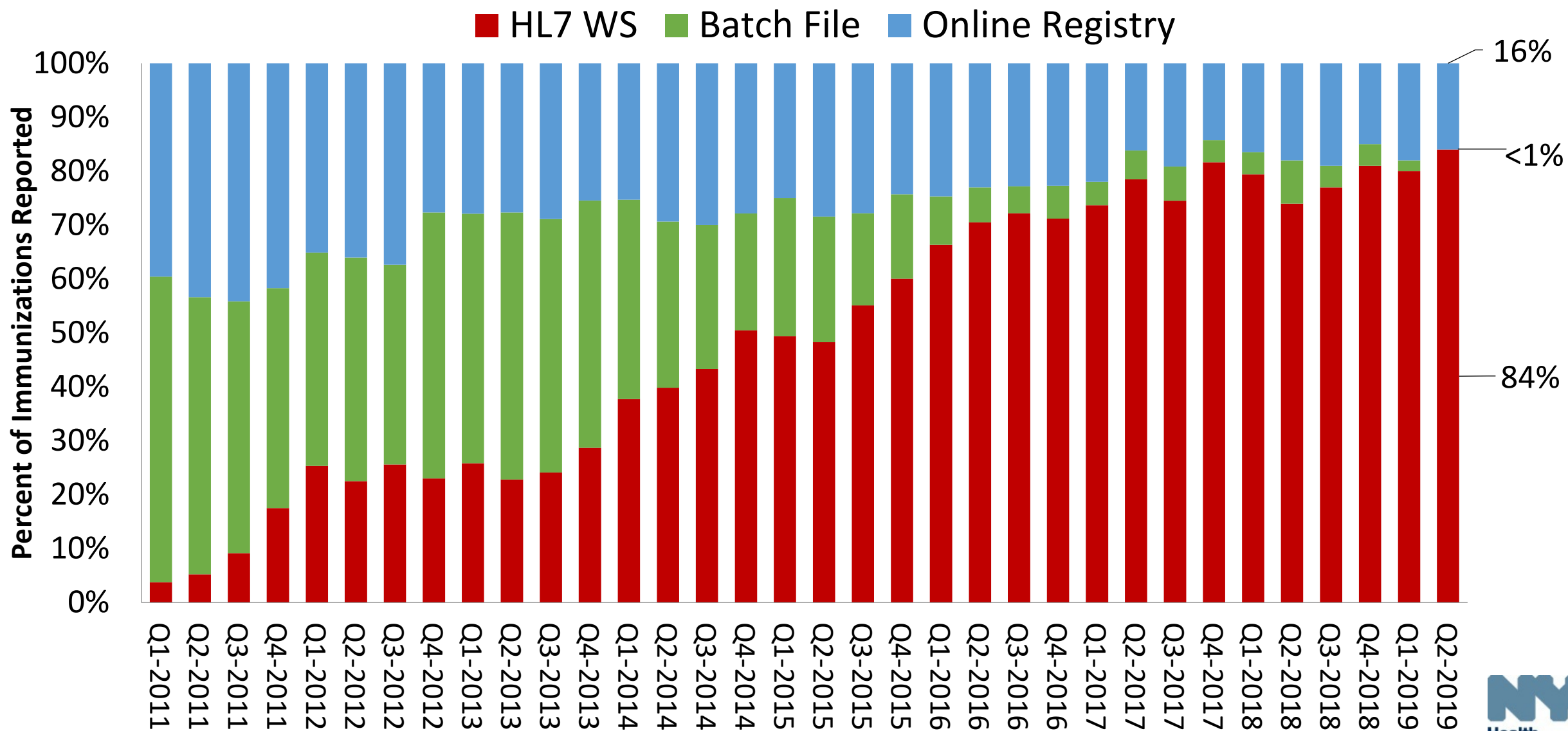
- Started 1996; implemented citywide 1997
- NYC providers mandated to report immunizations for patients age < 19 years
  - Reporting for adults  $\geq 19$  years requires consent
- > 7 million patient records
- > 104 million immunizations
- Population-based
  - Birth certificate data loaded weekly

# CIR's HL7 Web Service

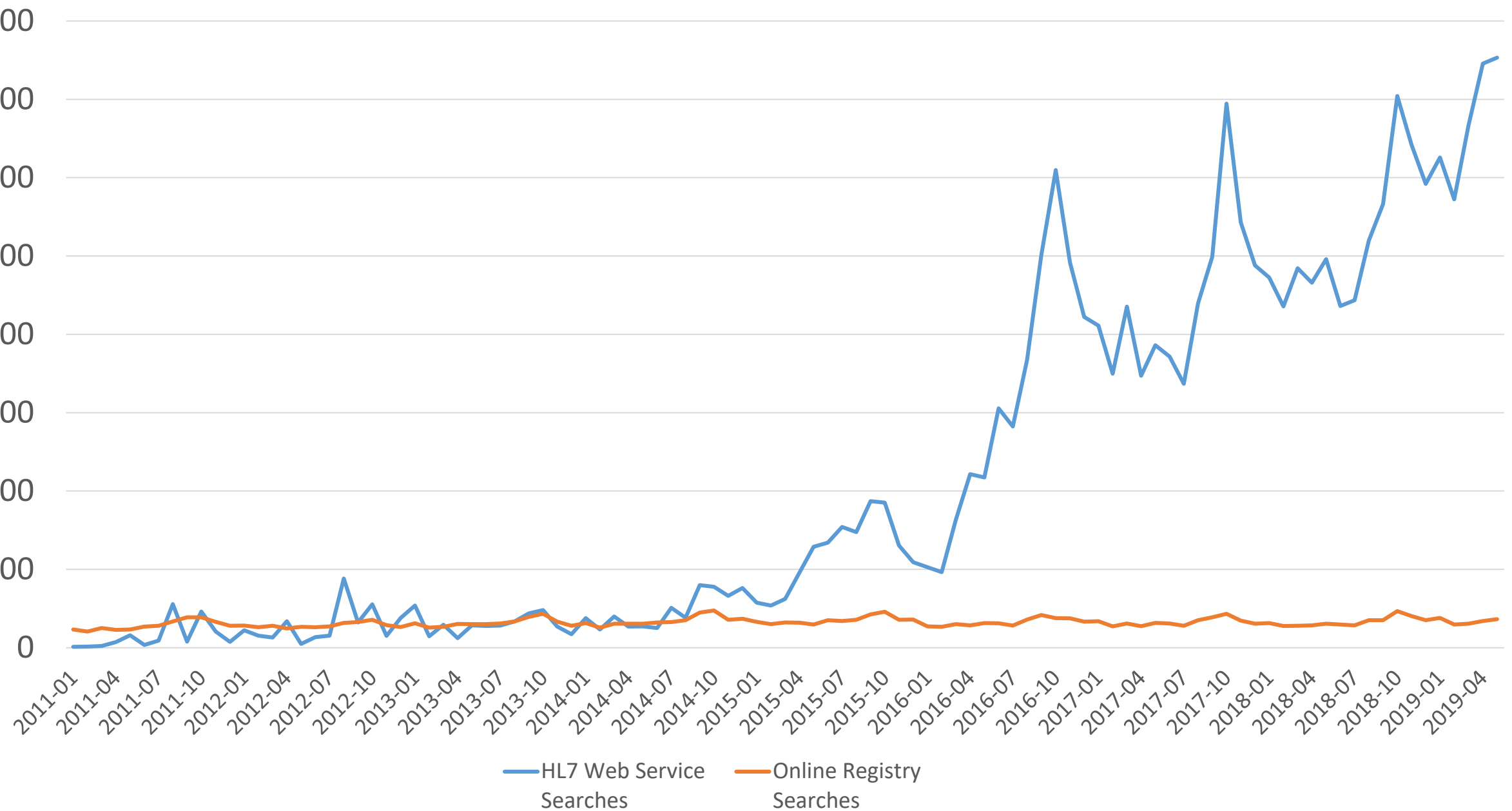
- CIR's HL7 Web Service (WS) was established in 2009 to facilitate real-time, electronic data exchange between EHRs and the CIR
  - First facilities connected in 2011
- 3,156 facilities ever connected\*
- ~70 EHR vendor products connected\*
- 59% of connected facilities are bidirectional

\*CIR data, as of 6/30/19

# Methods for Reporting Immunizations to the CIR



Patient immunization record searches in CIR, monthly, by real-time search method  
January 2011 to May 2019



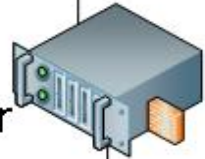
# Performance Challenges Necessitated CIR Database Migration

- Increased use of the CIR, especially HL7 Web Service
- Hardware upgrades, including additional processor cores, needed to improve performance
- Proprietary database platforms are licensed per processor core
  - Don't want to spend more money on software licenses
  - Don't want to spend more money on annual database support and maintenance contract
- CIR chose to migrate to open-source database – PostgreSQL
- Added potential benefit: NYC can share more of its software with the IIS community

Healthcare Providers Use  
Citywide Immunization Registry  
Online Registry, Web File Repository,  
Or Electronic Health Record System  
Integrated with real-time  
HL7 Web Service



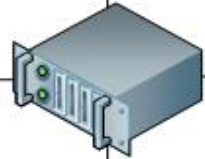
Apache Web  
Reverse Proxy Server



Tomcat and  
Apache CXF Web Service  
External Application Server



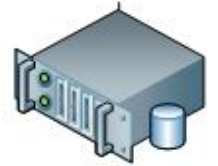
Tomcat, JBoss, and Payara  
Internal Application Server



CIR Database Server  
PostgreSQL



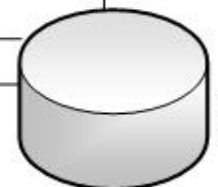
MCI Database Server  
PostgreSQL



Lead Quest  
SQL Server  
Database

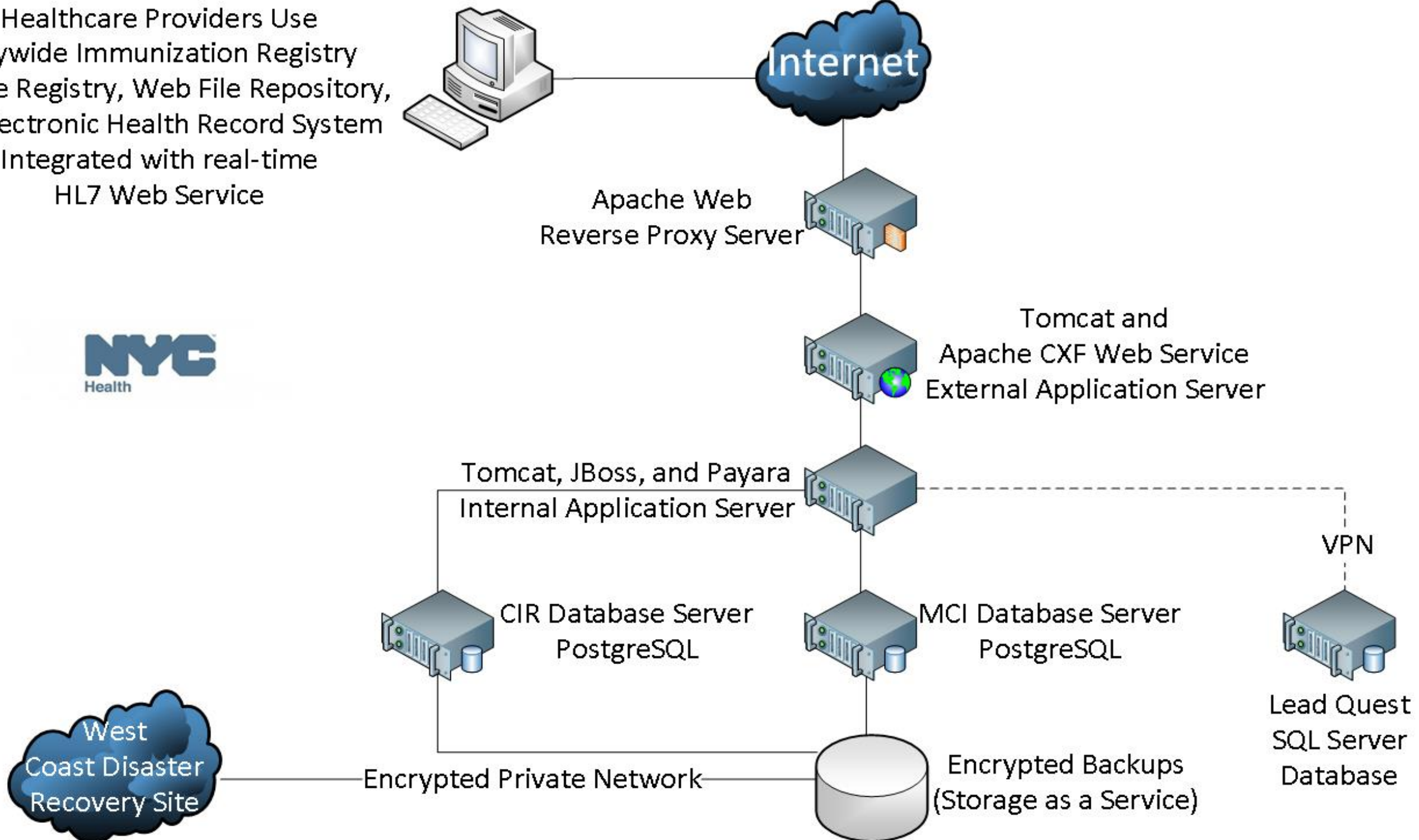


Encrypted Private Network



Encrypted Backups  
(Storage as a Service)

VPN





# CIR software architecture changes

- The old stored procedures were relatively fast and tightly coupled to the database platform
- Wanted database agnostic implementation in a middle tier
- Move business logic from Oracle stored procedures to a new set of Middle Tier Services (MTS)
- Makes it easier it to run CIR on any database platform

# HL7 Web Service Performance Challenges, Summer 2017

- CIR didn't have the capacity to keep up with reporting and query demand from external data exchange partners
  - Heavy use of other CIR services and applications, which rely on MTS, led to performance bottlenecks
- Pattern of degradation
  - Web Service was fine in the morning
  - Response times slowed as business hours progressed
  - Message failures occurred at times of peak demand
  - Web Service crashes (not responding to requests; required re-start)

## Methods/Implementation

---

# Method 1 -- Redundancy

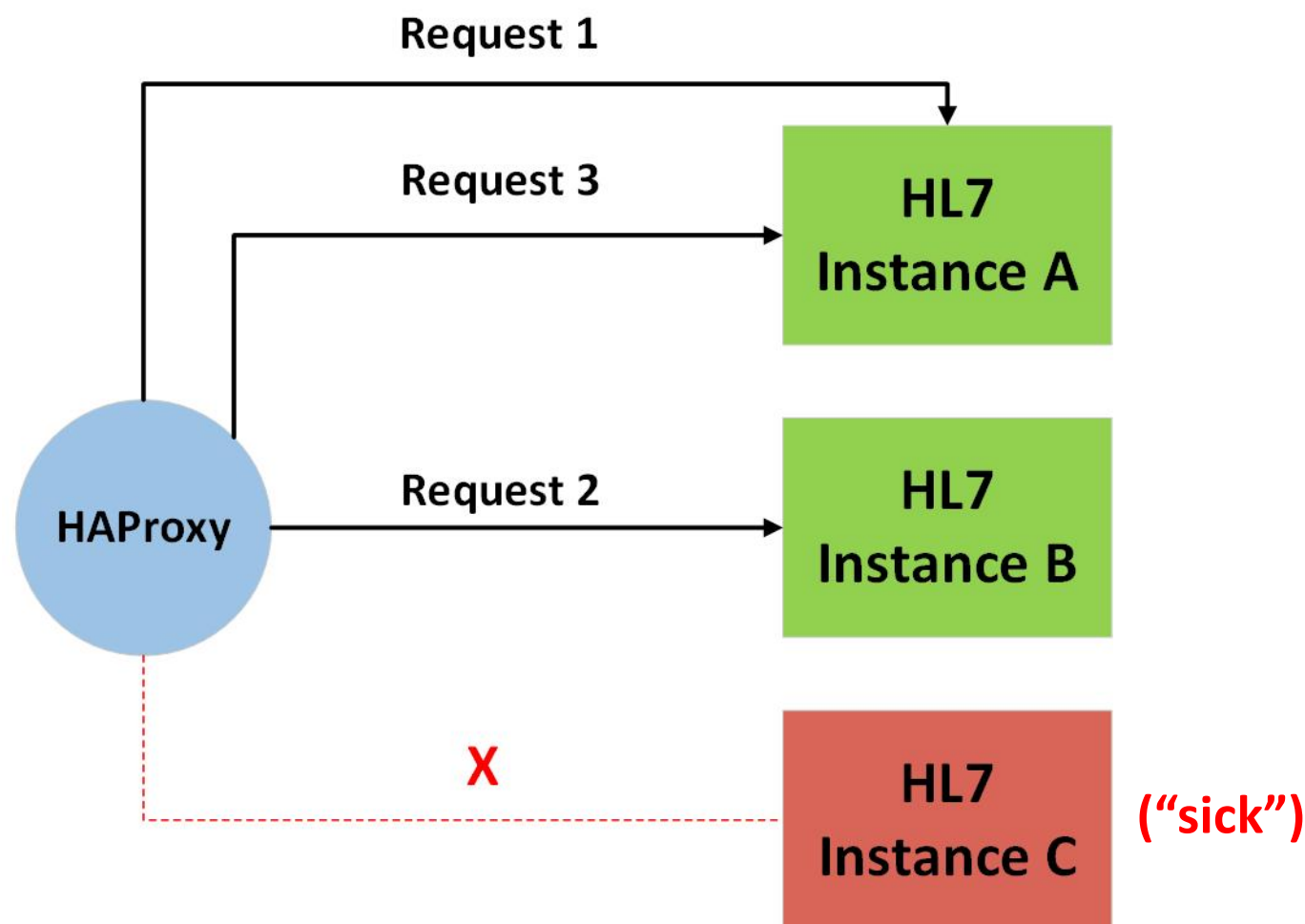
- Added software instances to increase reliability
  - ChoiceMaker probabilistic record matching services
    - July 2017 – 1 to 2 instances
  - MCI Services
    - July 2017– 1 to 2 instances
  - Middle Tier Services
    - September 2017 – 1 to 7 instances
  - HL7 Web Service
    - September 2017 – 1 to 3 instances

# Method 2 -- Load Balancing

- Load balancing allows requests to be evenly distributed between multiple instances and enables high availability
  - Implemented for HL7 Web Service and Middle Tier Services
- August 2018 – Replaced Apache reverse proxy with free, open-source load balancer, HAProxy
  - Apache proxy lacks the ability to monitor the health of the instances
  - HAProxy monitors if an instance is not functioning correctly and re-routes the request to other instances instead of resulting in continuous errors because of the “sick” instance

# HAProxy (multiple HL7 Web Service instances)

- Incorporates health check monitoring into load balancing
- Does not assign requests to “sick” HL7 instances
- Easy to create additional instances (i.e., adjust the pool size) to handle WS traffic needs



# Method 3 -- Database Indexing

- A database index is a data structure used to quickly locate and access data in a table
- Improves speed of database queries
- September 2017 – Added over 100 new indexes
  - Most frequently searched CIR data elements were indexed to optimize search

# Method 4 -- Increased Threads for Conveying HL7 Messages between CIR Servers

- Threads handle incoming requests in parallel
- HL7 Web Service spawns a thread to handle each request
- September 2017:
  - Increased the number of threads on the web server
  - Increased the number of tunnel connections between the web server and the application server
  - Increased the number of threads on the application server to service HL7 WS requests



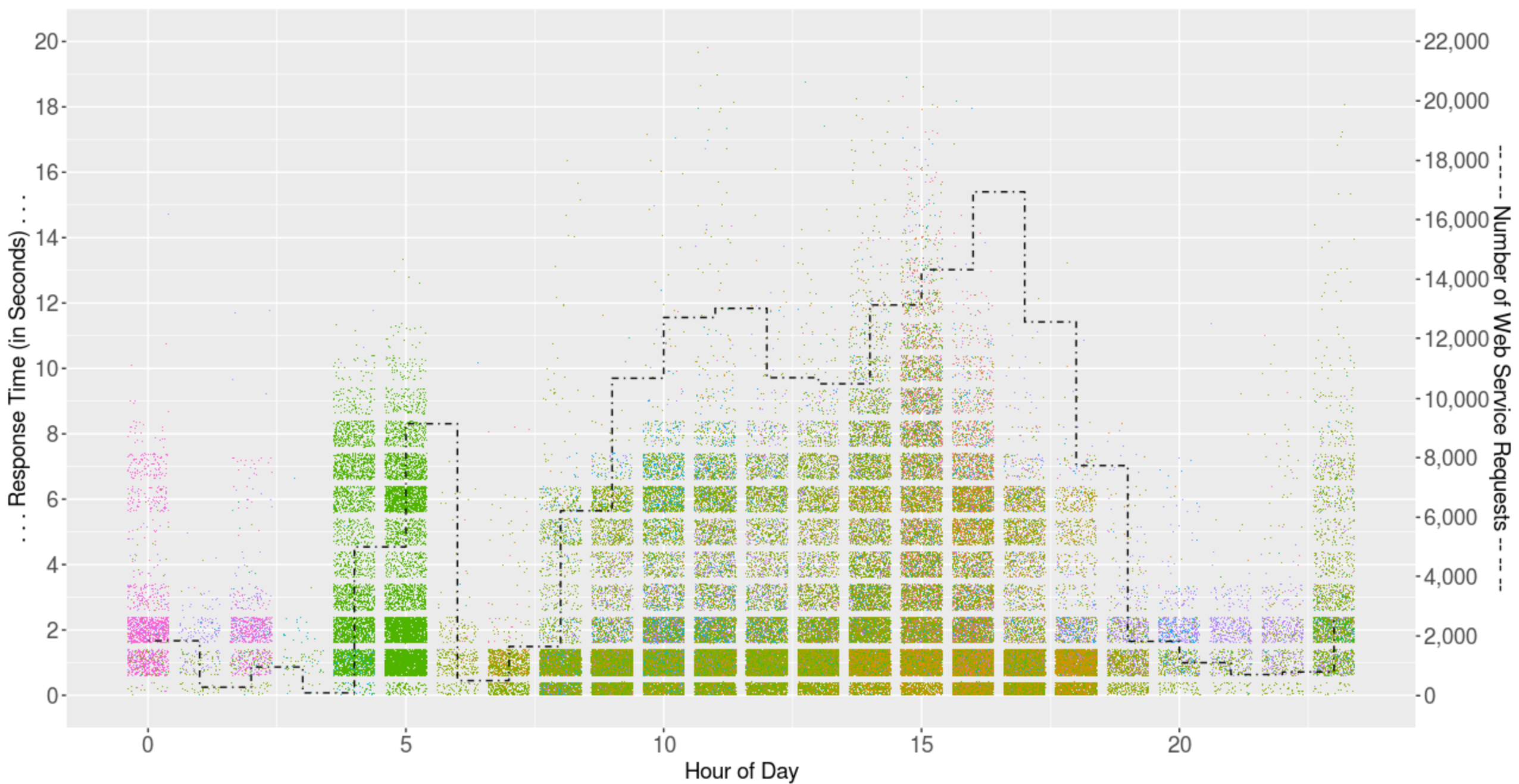
## Results

---

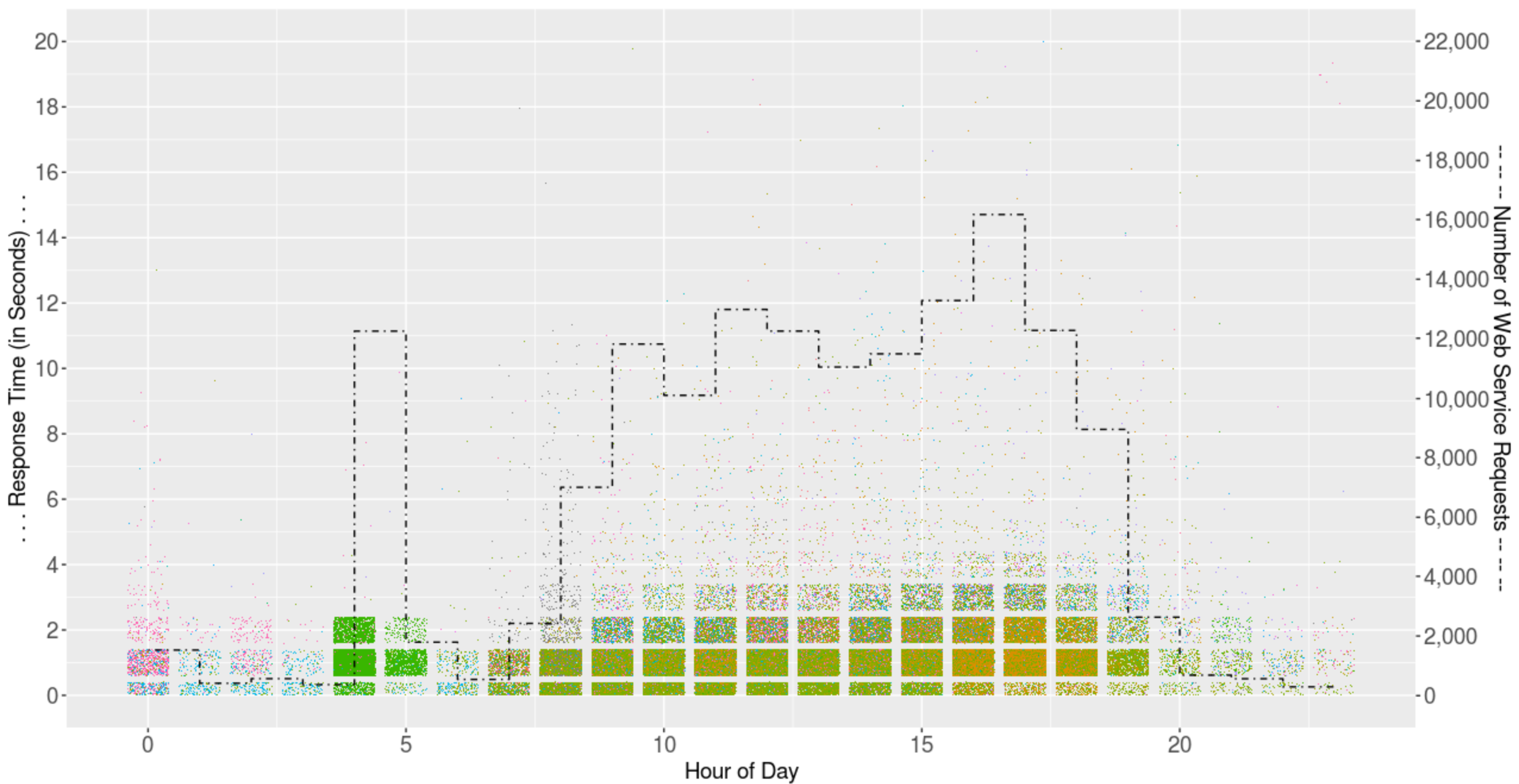
# Decrease in HL7 Web Service Response Times Following September 2017 Enhancements

- Average Web Service response time during Sept 2017:
  - 5.71 seconds
  - Number of HL7 messages processed = 3,428,372
- Average Web Service response time during Oct 2017:
  - 2.62 seconds
  - Number of reporting messages processed = 4,062,033
- Decreased response time by 54%

Response Time For All HL7 Web Service Requests, Colored by Sending Application  
2017-9-13

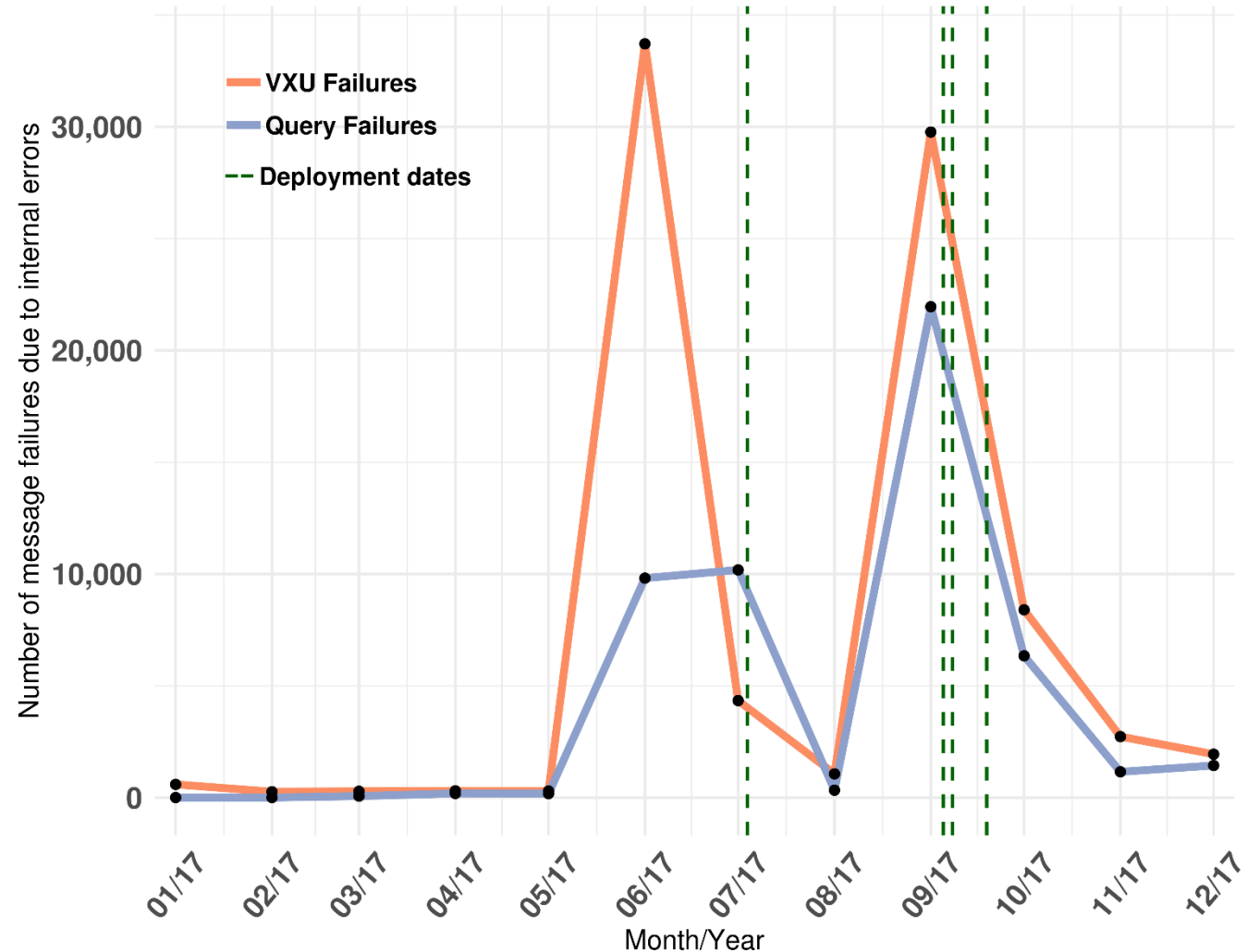


Response Time For All HL7 Web Service Requests, Colored by Sending Application  
2018-9-13



# Decrease in Failures Due to Internal Errors Following September 2017 Enhancements

- Average number of monthly failures due to internal errors from July – Sept 2017:
  - 22,543
- Average number of monthly failures due to internal errors from Oct – Dec 2017:
  - 7,335
- Decreased errors by 67%



# Lessons learned

- Many approaches to improving performance and reliability
  - Redundancy
  - Load balancing
  - Database indexes
  - More threads
- Carefully phase in major changes
- Constantly monitor and respond to emerging challenges
- Need ongoing quality measurement and improvement to sustain an Immunization Information System