

---

## CASE STUDY

### Application Migration & Optimization on AWS

---

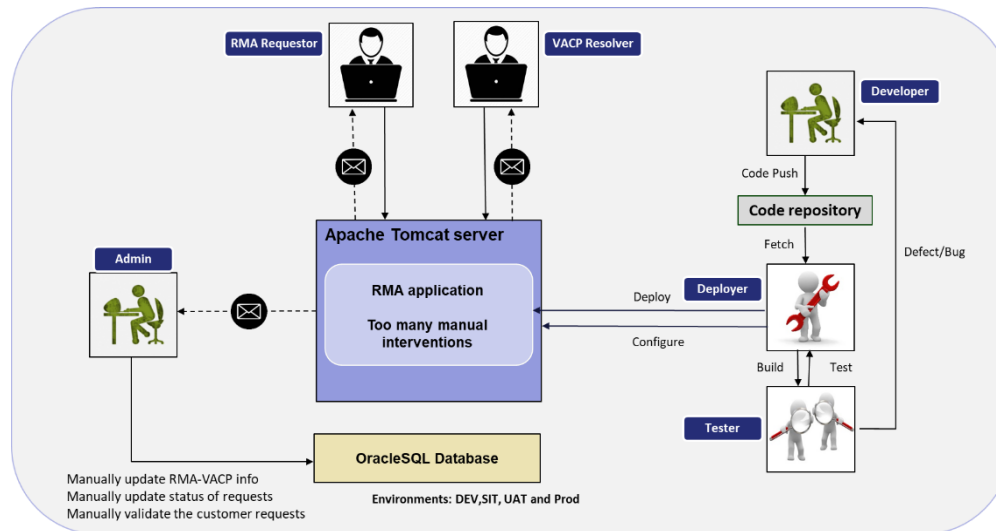
## Introduction

Client is a Fortune 50 company and is one of the largest Telecom service providers in the US and offers multiple products and services to consumers and enterprises. Being a large enterprise, the IT foot-print is huge and there is a plethora of applications both COTS and internally developed. The vision of the Business and IT leadership teams is to migrate applications from Data centers to public cloud infra-structure in a phased manner. Each business portfolio and the associated IT operations have chalked out a plan to migrate application workloads to AWS Cloud in 2 to 3 years' time.

## Current System overview:

In one of the business units, there was a need to modernize an application – to develop applications for better /effective management of returned materials and measure supplier discrepancy & performance for Supply Chain Streamline business processes to bring the materials from different systems across Lines of Business into common data objects.

- A customer having a discrepancy with a product purchased from Verizon raises a Return request with the application. A notification e-mail is received by the Admin regarding the request who manually validates the request to either move forward to vendor or deny the request as non-returnable. The Admin sends the RMA to the Vendor, manually updates the database with RMA information and sends an email to customer.
- Admin tracks the RMA with the vendor every week to check if resolution is provided or not, Vendor provides resolution to the request by providing credit or replacement for the product. Admin receives the credit/replacement information from Vendor and manually updates the shipment information in the database. Customer receives the credit or replacement within the stipulated days
- Developer pushes the code into code repository. Build team fetches the latest source code and manually builds the application into WAR using Maven. Testing team tests the application and raises defects/bugs to developers.



### Key Business/Systems requirements:

The stated goal was to make the development and deployment process more automated and hosted in the cloud. The application load will be seasonal and hence elastic scaling was essential and the data needs to be archived after every launch season of a new product for future analysis.

### Solution Considerations

The Client has enumerated the following requirements for mandatory compliance while designing an approach to migration.

- I. Automated Server Provisioning.
- II. Software Installation & Configuration (Java, spring REST service, Spring Microservices, Tomcat, Oracle)
- III. Automated Code Deployment
- IV. Scheduled Backup implementation

### Choice of Migration approach

Since the application must be made cloud ready, the recommendation was to rewrite the same with a modern architecture and a contemporary tech stack. It was architected & developed in a domain driven Micro service structure to leverage full benefits of DevOps and cloud

- Created DB model for common objects across systems (COEP, CATS, PeopleSoft etc.) and create ETL scripts
- Created DevOps pipeline for auto build, unit test & regression test integration
- Implemented Watson Search to facilitate flexible and faster part search across systems
- Application tools stack - Java, Spring boot, Swagger, Angular JS, Ionic framework, Oracle DB, ELK, Jira, Jenkins, Selenium, TestNG, AWS

The application has an external Interface BI which interacts with 5 external systems. The BI data is replicated into the application Database on a scheduled basis.

## Planning Phase:

### Timelines

The client engagement was an end to end application modernization to run on AWS infra-structure with a scalable architecture. The system study took about 2 weeks and the development using Agile scrum methodology took 8 weeks (4 sprints). A team of DevOps engineers worked in parallel to design and implement a CI/CD pipeline using Jenkins and deploy automation tools for testing.

Phase	Outcome	Elapsed time	Responsibility
System Study	Sign-off PRD	2 weeks	Newt/Client
Design	Sprint plan and deliverables	1 week	Newt
Development	Code base in AWS environment	6 weeks	Newt
UAT	Deployment pipeline	1 weeks	Newt/Client
Production cut-over	AWS configuration tweaking	1 week	Client

Start of the project: August 2017

End of the project: December 2017.

### Team structure

The project team had 2 core Java developers, 1 Micro-services architect 2 testers and 1 AWS certified solution architect.

## Implementation Phase:

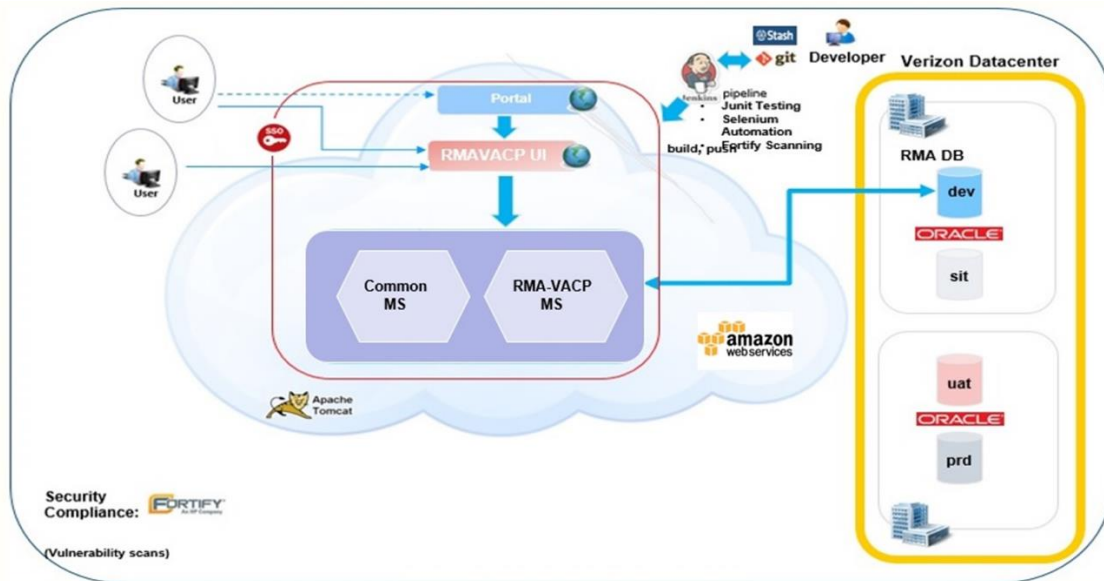
### Use of DevOps tools

The development process was accelerated with the use of DevOps tools for build and test management, Infra-structure automation and code deployment.

JIRA, Maven, Selenium and Stash were used along with the development tools.

### Deployment Architecture

The newly developed software was deployed in the AWS environment on 20 VMs for different components of the solution. The systems architecture is a under:



## Monitoring

Infra-structure monitoring was done using CloudWatch and Nagios and remediation was based on triggers and alarms configured for service health or capacity thresholds.

## AWS services used:

The AWS services used were: CloudWatch, EBS, EC2, S3 and RDS

## Third Party services used:

The application used Java, Spring REST service, Spring Microservices, Tomcat, Oracle, Swagger, Maven, sl4j, Junit, Bootstrap, JIRA, Stash

## Security considerations and implementation

In line with the expectations of the application owner within the enterprise and meeting the corporate security guidelines, the following best practices were implemented.

- ✓ VPC was adopted with NAT for enterprise access to EC2 instances
- ✓ All programming / API access to AWS was encrypted with TLS and user access was thru SSH.
- ✓ AMIs and deployment scripts were hardened as per business requirements
- ✓ Regular vulnerability scans were done
- ✓ IAM policies were baked with enterprise LDAP credentials.
- ✓ Data at rest was protected by using EBS and its native encryption

## Business benefits of the migration:

- Automated workflow cutting down on deployment time and providing consistency
- Respective Stakeholders periodically notified of each stage of the procurement process

- Made procurement process more efficient, user friendly, and accessible through Mobile devices