

CASE STUDY:

Cloudification & Migration On-Premise App to AWS Cloud

Proctor Finance **reduced TCO** almost **20%** by moving workloads to **AWS Cloud**.

Client Overview

Proctor Financial, Inc. is a leading provider of lender-placed hazard & flood insurance including hazard tracking services to the mortgage servicing community. Since the 1970s, Proctor has blended comprehensive insurance programs with innovative tracking & reporting services to meet the specific servicing requirements of mortgage servicers, community banks, credit unions, sub-servicers, distressed asset investors and the U.S. Government.

Solution Overview

R Systems leveraged its global delivery model to provide the required services for this project. The engagement roadmap was planned with due consideration to client's objectives. Agile Scrum methodology was adopted to expedite the application development, migration, and faster time-to-market. R Systems team focused on providing rich Quality of Service (QoS) & Quality of Experience (QoE) to the client through innovative efforts and by adopting standard industry best practices.

Challenges

The client has migrated to a robust and scalable platform to endorse the brands of various finance provider companies across the globe. The customer had VMware vCenter based infrastructure running on version 6.5 with more than 250 virtual machines and 40TB data. In contrast, the database size was around 500GB running on MS SQL 2017 Standard edition engine with approximate yearly growth of 50GB (~10%). As a result of the legacy system, limited end users handled minimum hardware & software support & costly maintenance of these

applications. As facilitating a highly available & cost-effective solution was of utmost importance to be taken care of:

- Information flows
- Data markets
- Restructuring for the new digital realities with modern technologies
- Opportunities
- Steps for business transformation
- Proctor Finance Inc. for an exceptional experience to the end-users

Why AWS

Amazon web services is a pioneer in providing leading technology services:

- Global Reach and High Availability.
- Easy to get started and schedule migrations.
- Incremental replication for fast migration with limited downtime.
- Cost effective migrations with no additional cost to use the service.
- No dependency on the agent's deployment to migrate.
- Security and Compliance like PCI-DSS, SOC-1,2,3 & ISO
- Well-Architected framework as guidance for resilient architecture
- Service monitoring with CloudWatch and SNS for real-time notifications.
- Systems Manager to regularly patch the Windows EC2 Workloads.
- License options addressing for BYOL and AWS provided OS

Our Solution

R Systems delivered database migration solutions leveraging the potential of various popular AWS services stack, e.g., Migration Hub, Application Discovery Service, Database Migration Service, Server Migration Service, CloudFormation, EC2, ELB, RDS, etc., for planning, migration, and optimization. Completed entire database migration using DMS. Used migration type "Migrate Existing Data." Once the migration is done, it replicates the data changes (CDC).

DMS managed the entire migration process via AWS Console. It was homogenous database migration which identified all the risks during the Pre-migration assistance.

Database Migration Service:

Source and target endpoints were defined with proper connectivity. "t3.large" instance was chosen as replication instance with engine version 4.3.

On-premises Database environment, versions, and size:

On-Premises SQL Engine and Version: **MS SQL 2017 Standard Edition**

Database Size: **500GB**

Port Number: **1433**

Number of Databases: **25+**

Number of Tables: **1500+**

Project Time: **10 Weeks**

Planned Vs Actual RPO & RTO:

For relational Databases, we used AWS PAAS Services. Due to AWS RDS managed services offerings, the RPO & RTO was minimum (> 1 or 2 min) and the switch over was taken care by AWS.

Downtime:

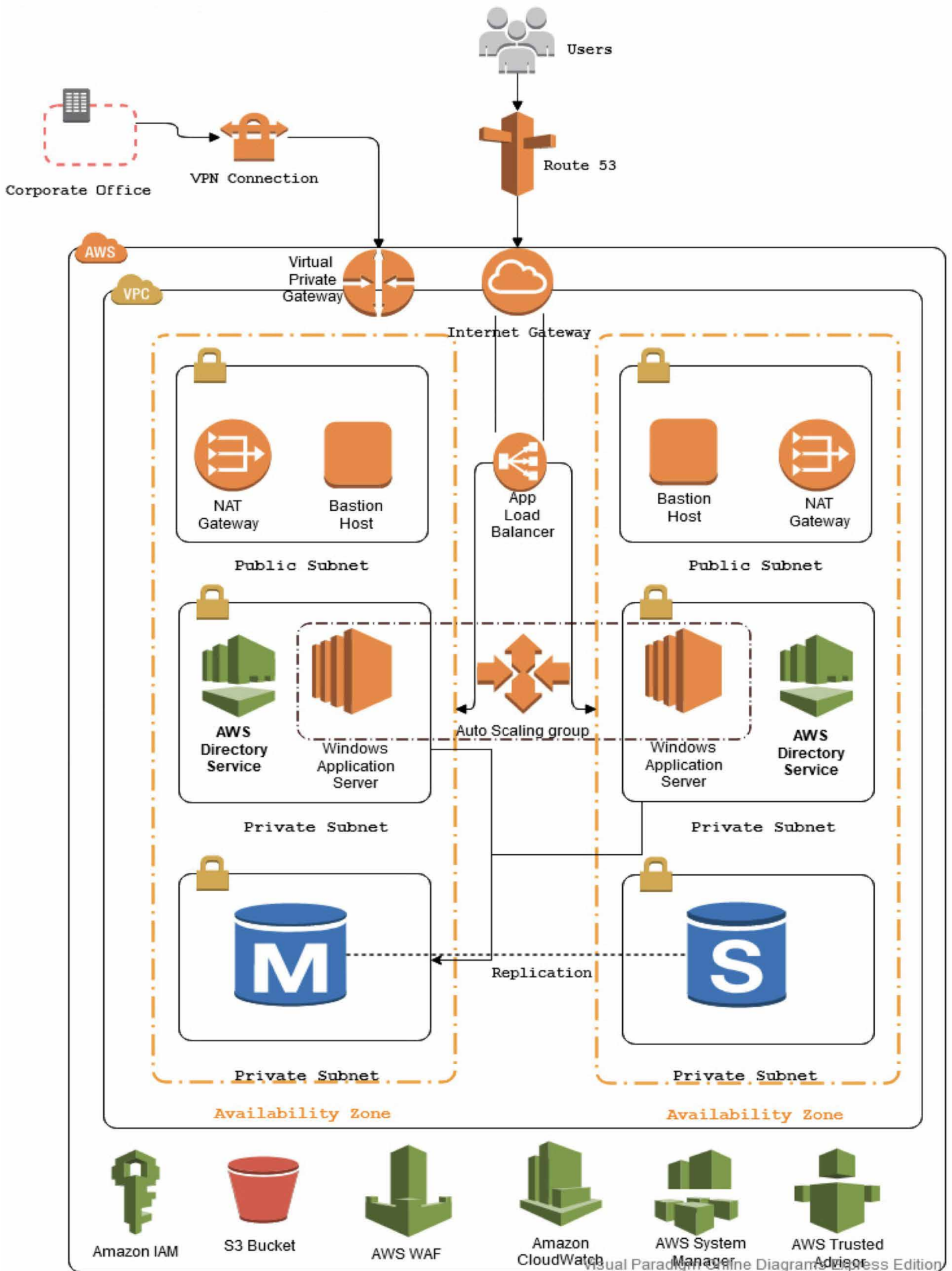
The Downtime was minimum as we performed the complete testing for each application before the Actual Cutover. We tested many point-in-time replicas and scheduled the last replication before the cutover.

Monitoring and Backup:

We used the Migration Hub Console to monitor the replication jobs. For backup, we had enabled automatic snapshots.

Deployment Highlights:

- Migrated on-premises VMware-based infrastructure to AWS.
- Migrated on-premises Databases to AWS RDS (Multi-AZ) using DMS.
- Re-architected and deployed legacy applications to run in AWS.
- Addressed application dependencies, migration strategies, costing, OS licensing using Application discovery service, SMS, TCO, S3 Calculator, etc.
- Boosted migration plan for on-premises servers based on RTO and RPO.
- Configured .NET apps on MS Windows EC2 instances in the multi-AZ environment
- Used RDS MS-SQL server in Multi-AZ config for scalable DB operations.
- Autoscaling groups to reduce the operational complexity for managing applications.
- Used System manager for Microsoft VSS enabled snapshots.
- Configured required services: IAM, Elastic load balancer, Route 53, SNS, etc.
- Notification and alerting configured on AWS SES and SNS
- AWS Migration Hub and SMS services to track and migrate the workload.

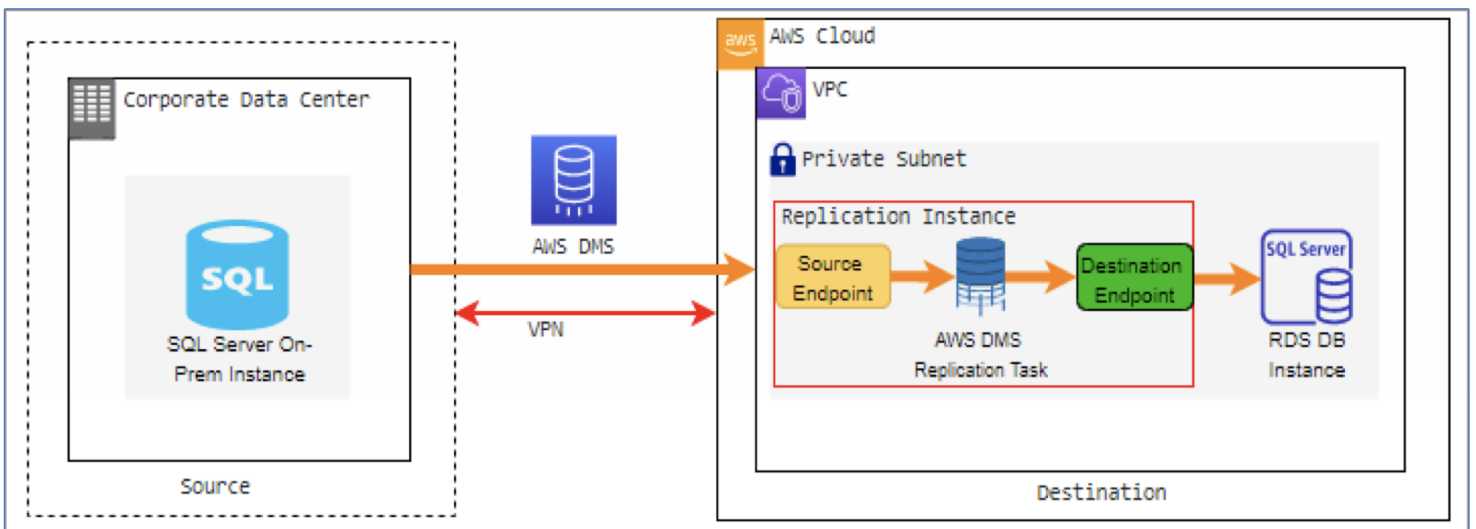


AWS services used as a part of the solution:

- Server Migration Service to replicate on-prem VM's to AWS.
- Database Migration Service to migrate on prem database server to RDS MS-SQL – Managed SQL Database.
- **Migration Hub** – Single place to track the migration progress
- **S3** – Scalable storage in the cloud (To store flat file – like snapshot, images, videos etc.)
- **Systems Manager** – To manage OS patching of Windows based EC2 instances.
- **Route 53** – Scalable DNS and Domain name registration
- **ACM (Certificate Manager)** – provision, manage and deploy SSL certificates.
- **AWS CloudWatch** – Monitor resources and applications (logging)
- **SNS** – managed message topics for pub/sub
- **SES** – Email sending and receiving services
- **Elastic Load Balancer** – Distribute Load to servers in multiple AZs.
- **Auto Scaling** – To increase or decrease server count as per incoming traffic.
- **EC2** – AWS Compute service for Linux and Windows workload
- **CloudFormation** – Service to model and provision the resources
- **IAM** – Manage User Access and Encryption Keys
- **CloudTrail** – To log all the activities in AWS account to a S3 bucket.

OS & Database:

S.N.	Operating System	Database
1	Windows 2019 Datacenter edition	RDS MSSQL Standard edition 2017



Database Migration Service:

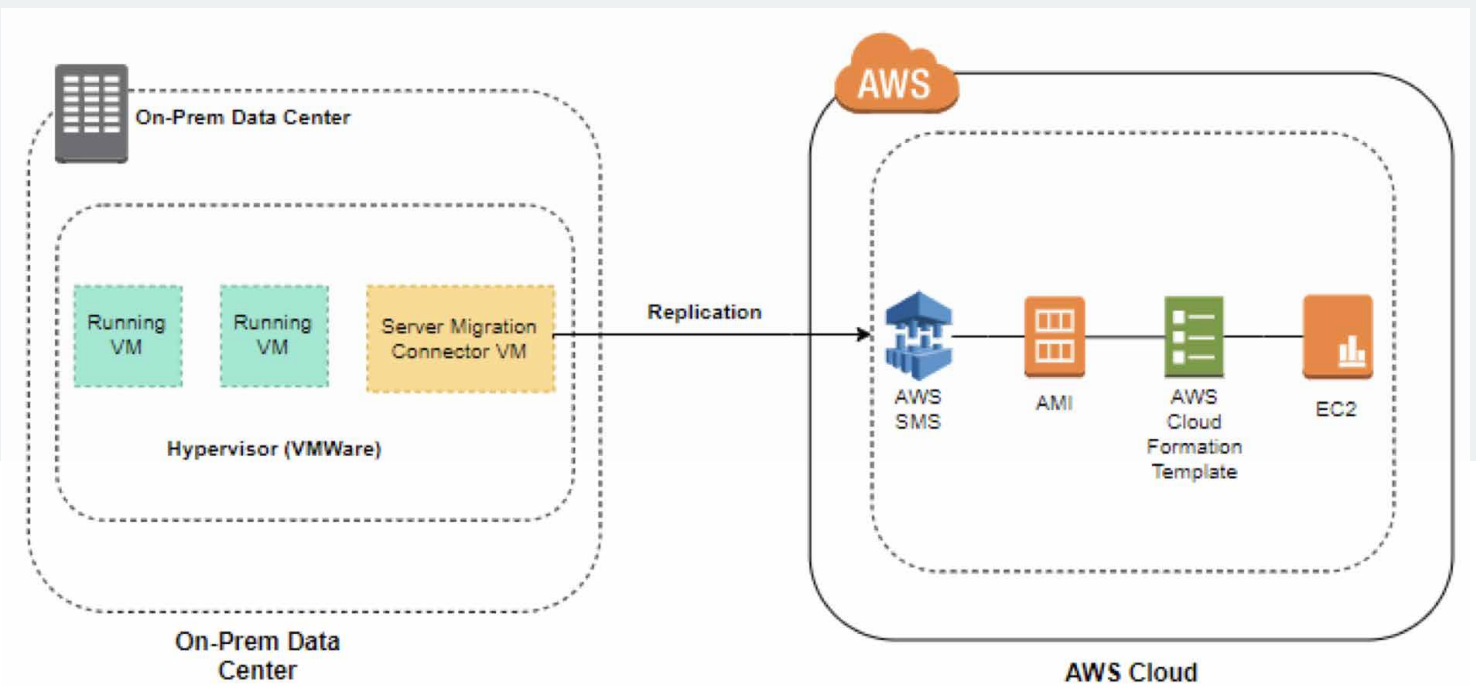
- Used DMS to migrate 50 GB database with more than 1500 tables using AWS DMS with expected yearly growth around 10-20%.
- This was homogenous database migration between two similar database platforms.
- We used the private replication instance with VPN connection.
- We configured the “t3 large” replication instance to connect with both the databases and perform migration.
- Source database was fully operational during the migration process and the application connected to the source database was not impacted.
- We configured the IAM user with required policy to access the DMS resources.

- We enabled the AWS KMS encryption key to secure data at rest, encrypt the storage used by replication instance and endpoint connection.
- Configured the SSL/TLS for communication with other AWS resources.
- Configured the CDC task to capture and replicate changes that are made to data.

Server Migration Service:

- We deployed one Server Migration Connector with thick provisioned option onto the VM environment running vCenter version 6.7
- We migrated 250 VMs to AWS cloud, & more than 150 VM was larger than 100 GB.

- We used the RDS for Database as it is a managed service, so RPO & RTO is minimum (within 1 or 2 min) in case of Database and take care by AWS.
- Downtime was minimum as we did the complete testing for each application before Actual Cutover. we also tested many point-in-time replicas and scheduled last replication before cutover
- Project duration was 12 weeks including assessment and migration phase.
- Replication jobs were monitoring centrally from the Console.
- We used AWS Migration Hub to generate right-sized EC2 instances for running on-premises workloads in AWS.



AWS Systems Manager:

More than 250 servers were running on AWS cloud and using AWS Systems Manager to control infrastructure. Used AWS standard AMI with SSM agent preinstalled by default.

- Integrated patch manager with IAM and CloudWatch events to provide a secure patching experience that includes event notifications.
- Configured Least – Privilege permissions through IAM using fine grained access controls.

- We used AWS EC2 tags features to install patches to large group of instances.
- We maintained Windows to setup recurring schedules for managed instances to run admin task like installing patches and get updates without interrupting operations.
- Tasks like Agent Updates & applying Drivers were done using AWS Systems Manager - Automation Service.
- AWS SSM – Inventory Manager helped to collect the software inventory and metadata about application, files, components.

- Resource group helped to manage, monitor, and automate task on large number of resources at one time.

Deployment Patterns:

Patch Manager: AWS Patch Manager used to automate the process of patching for both operating systems and applications. Patch Manager feature patch baselines rules helped for auto-approving patches within days of their releases with list of approved and rejected patches. Configured a maintenance windows to install patched on regular basis.

Workload: Running combination of Windows and Linux servers with long running instances

Operational Excellence:

Monitoring tools AWS CloudWatch was used. Actionable alerts were generated from monitoring tools which were responded by IT team. CloudWatch dashboard was created for IT team to monitor the real-time status of AWS resources. AWS Systems Manager ensured that all Windows based workloads were AD joined and regularly patched in the defined maintenance window.

Security:

Utmost care was taken to provide security inside/outside the AWS perimeter & at multiple layers

- All the users were having access to AWS console and resources through their individual IAM users. Multi factor authentication (MFA) was made compulsory.
- Used AD trusts to connect AWS Managed Microsoft AD to on premises AD. This means users can access AD-aware and AWS applications with their on-premises AD credentials, without needing to synchronize users, groups, or passwords.
- Secured SMS IAM user was created with appropriate granular permissions which disabled post migration.
- MFA was enabled for root account. Programmatic access was disabled for root account.

- Separate subnets were configured for each layer (Web, Application, Database) for network-level isolation
- Applications on EC2 instances were configured to use IAM roles.
- Application Load Balancer was used to route the traffic to private subnet's servers over multi-availability zones of AWS for higher availability, and achieve near DR deployment
- Security groups & NACLs were configured with least permissive rules.
- Antivirus protection using Trend Micro endpoint protection to prevent cyber-attacks.
- EBS volumes were encrypted using KMS to protect data in transit.
- VPC flow logs were configured to monitor the incoming/outgoing traffic at ENI level and logs were stored in S3 bucket for audit purpose. Logs were stored in S3 bucket for 90 days and then archived to Glacier using Life cycle policies.
- CloudTrail was configured to record the API activity of AWS services and logs stored in S3 bucket for audit purpose.
- Enabled Amazon S3 server-side encryption (AES-256) to encrypt data at rest.
- Hardened bastion hosts were configured to access Ec2 servers in private subnets.

Reliability:

R Systems deployed solution in single AWS region leveraging AWS Well-Architected Framework which includes using two availability zones (AZs) to ensure all applications are always available.

AWS Life Cycle was configured to take periodic backups of the tagged EBS volumes. Thus, ensuring the availability of snapshots of EBS volumes for recovery.

AWS managed database service was used to deploy MS-SQL 2017 server in Multi-AZ architecture to ensure high availability of Database. Both Primary & Standby database remained in sync & automated backups were configured to minimize RPO & RTO.

Cost Optimization:

- Autoscaling was configured to automatically reduce the server count across all layers (Web, Application) based on incoming traffic.
- Using managed services e.g. RDS, AWS directory service helped to reduce operational costs.
- M5 family instances were selected and 1-year RI (reserved instance) was purchased to reduce bills.

Result

- Successfully migrated on-prem legacy system and more than 250 VMs to Amazon cloud
- Improved user experience and increased customer satisfaction
- Services are adaptable for use in multiple contexts
- Increased flexibility, scalability, and cost-optimization
- Easier to build and maintain application
- Enabled quick identification and resolution of the root cause of poor application performance

About R Systems

R Systems is a global technology and analytics services company. We help our clients achieve speed-to-market, overcome digital barriers, and create business value with our specialized service offerings and consultative business approach. We speak the language of business as fluently as we do the language of technology. In other words, we speak digital. Our goal: accelerate our clients' digital leadership.