

Perspectives | 2021-03-24

Achieving escape velocity in the cloud

To meet its growth objectives and customers' rising expectations, Venerable needed to modernize a core application with 1980s roots. To get there, it created a technology ecosystem of partners that pivoted around Amazon Web Services to quickly up its cloud migration game.

Sometimes, a business has to jettison its past in order to embrace the future.

[Venerable](#) was created in 2018 as the result of a divestiture from a large U.S. insurance company. Venerable immediately adopted a cloud-first strategy, which meant zero investment in establishing a mainframe infrastructure. But this created a challenge: a critical business application for managing agent commissions, known as the Agent Management System (AMS), had to be taken off the old mainframe and modernized to run on [Amazon Web Services](#) (AWS).

There was a strategic business element at stake, too. Once the modernization was completed, Venerable planned to use the application to enable an acquisition engine, allowing the company to pursue its business objective of growth through acquisition.

The migration was hardly a minor challenge. As Venerable CIO Tim Billow notes, the AMS system was custom-built in the 1980s and had grown extremely complex, riddled with add-on code, over the decades. That's a situation many traditional enterprises find themselves in; where digital natives are born with a cloud-first mentality, older businesses (or, in



Related industries

[Communications](#)

Related articles



Communications

Closing the Purpose Gap

Venerable’s case, those spun out of older businesses) require a shift in mindset, priorities and spending.

But for Venerable, as for other businesses in similar situations, all this change brought risk. AMS had been customized through the years; it performed its task well, handling large volumes of data with sufficient processing speed. Venerable didn’t want to sacrifice these positives. The company also wanted to make as few user interface changes as possible to minimize any retraining and upskilling required by both users and support staff.

To make the transition, Venerable engaged us along with [Heirloom Computing](#) to re-platform AMS into a cloud-native Java application through a process known as automated refactoring using [Heirloom’s](#) product suite.

The modernization challenge

At Venerable, AMS provides core business-critical functions for commission calculations, payment generation, license validation, supplemental compensation, accounting and statements. Featuring more than 650 online screens, it is accessed by approximately 100 Venerable users spread across multiple groups, each with different roles and security access. The modern application had to provide exactly the same business functionality when running on AWS — this was non-negotiable.

As revealed in the figure below, the task would be formidable due to the application’s vintage technical footprint. AMS was a CICS mainframe application with a small percentage of its base system supported by batch processing. Written in COBOL, it used Basic Mapping Support (BMS) for screen definitions. Data for the online application was hosted in Virtual Storage Access Method (VSAM) files and IBM DB2 database.

When it comes to an professional life, your expectations that businesses have trouble understanding, let alone meeting. We explain how employers can cater...

[Back to top](#) 

[Read more](#) 



Communications

5G + IoT = Opportunity

Ultra-fast wireless connectivity has the power to handle burgeoning data volumes from connected sensors. Communication service providers should seize the moment to go to market wit...

[Read more](#) 



Communications

Getting Fit for New-Economy Realities

Leaders, laggards and everything in-between: Here’s a look at how telecom, tech and media/entertainment businesses are progressing on the digital maturity curve.

[Read more](#) 



Mainframe Makeover

To learn more, visit the Cloud Enablement section of our website or contact us.

Cognizant

Users accessed the application through terminals via IBM Customer Information Control System (CICS) command-level transactions. Background CICS transactions were also used heavily within the CICS environment. With code being batch-processed in small increments, online performance was a problem, to say the least.

From a business perspective, key goals included enabling and building an acquisition engine to support expansion plans — the company may very well acquire mainframe workloads in the future, so the AMS transition serves as a model of sorts. Venerable is determined to grow its market, diversify its customer base, and serve new market segments. Naturally, users will demand the self-service options, cross-platform transparency, and overall ease of use found in today's applications.

Moreover, senior management was determined to strike out on a new, modern technology path — divesting Venerable's tech platform, in other words, to parallel the business's divestiture from its prior owner.

On the technical side, while AMS was transformed into a cloud-native Java application and onto AWS Cloud, Venerable wanted improved performance (especially at peak periods) and disaster recovery (Recovery Point Objective and Recovery Time Objective) capability.

These varied goals had to be accomplished within a 24-month timeline agreed on by Venerable and the prior owner for transitional services — while minimizing disruption of business and inconvenience for users. Venerable had agreed to move the divested workload to AWS in multiple lots for better management. The mainframe application and data were moved in the final lot, which included two phases.

Communications

Keep It Simple and Ensure It Works, Especially for Generatio...

When it comes to experiences, young consumers know what they want. Here's how organizations can put their preferences to work.

[Read more](#) ▶



Communications

How COVID Changes the Digital Divide Debate

By exposing the worst elements of our digital have and have-not world, COVID-19 has shone light on how the divide can and should be bridged.

[Read more](#) ▶

Elevating to the cloud

Working with a Venerable subject-matter expert, a Cognizant team with active support from Heirloom staff refactored and re-platformed AMS to [Amazon EC2](#). The engagement consisted of multiple phases:

1 Discovery.

During this month-long phase, we completed a detailed analysis of AMS, including inventory baselining. Cognizant's in-house proprietary inventory analysis tool and Heirloom's [Probe](#)TM were used to produce an application map; collect application assets; identify complexities and dependencies; and extract metadata required for code refactoring and data migration.

2 Data migration.

Data consisted of 110 VSAM files and 25 DB2 tables, which were migrated to relational tables in a Microsoft SQL Server on [Amazon RDS](#) using a mix of Heirloom's Data Migration Toolkit and custom-built utilities, without any changes to the original application source. For a few VSAM files with complex file layouts (variable length, non-sequential, identifier-based segments), data migration was handled by the Cognizant team with custom-built utilities. Data backup and restoration was accomplished using [Amazon S3](#).

3 Application and ecosystem conversion.

This phase required several months, as our team converted a wide range of programs and migrated associated data. CICS BMS screens were automatically refactored into modern web pages. We were careful to retain the look and feel of the user interface so that the end-user experience wasn't changed during the migration. The Easytrieve and Assembler programs were converted to intermediate COBOL programs using our proprietary tools.

The online transaction processing and batch processing comprising COBOL programs were also refactored to Java using Heirloom's Elastic framework. For the RACF Security to Active Directory conversion, access rules for mapping users, groups and resources were unloaded from the mainframe database. Custom scripts were then written to convert the RACF access rules and subsequently loaded into the Active Directory schema. [AWS Secrets manager](#) was leveraged for securely storing the service account and database credentials and encryption keys. A Linux version of Tivoli Workload Scheduler was used to mimic the batch mainframe workload scheduling process. Custom scripts were built to establish smart communication between scheduler and Heirloom's Elastic framework.

4 Testing.



This was critical to validate the functional equivalence between the mainframe application and the refactored Java application. Overall, five months were devoted to this phase. Unit testing ensured that each class and method in the refactored programs was tested independently by mocking dependencies. System integration testing verified the behavior of the complete system (including interactions between components and with other systems). Finally, user acceptance testing was performed to validate that the application met business objectives.

5 Production.

In this final phase, a CI-CD Pipeline was created to automate the building, testing and deployment of applications on the AWS Cloud. Jenkins was used for build and deployment. [AWS CloudFormation](#) templates were used to provision the different environments in a fast and consistent manner. After all pre-defined launch criteria were met, the application was promoted to production and deployed. [Amazon Application Load Balancer](#) was used in tandem with [AWS Cloudwatch](#) to achieve high availability. A DNS failover mechanism was implemented to make the application fault-tolerant by routing the traffic to a disaster recovery (DR) instance in another availability zone.

We delivered the engagement by overcoming several technical challenges including array-based dynamic memory allocation, CICS housekeeping processes replication in Linux, and pointer-variable handling differences between COBOL and converted Java.

A modern digital business emerges

Working together with [AWS](#), the team created a modern cloud-based environment from a complex array of legacy applications. This achieved the dual business objectives of supporting the building of an acquisition engine to underpin business expansion while divesting the technology stack during the transitional services timeframe with the prior company. The project achieved Venerable's goal to align with its cloud-first strategy, among other imperatives, which included:

- Avoidance of more than a million dollars a month in operating payments to the prior company by eliminating mainframe dependency within a given timeframe.
- Matching its prior mainframe transaction/batch processing service-level agreements.
- Dual code-maintenance capability (i.e., in COBOL and Java) as enabled by Heirloom.
- A functionally equivalent and stable cloud application that was delivered on schedule (16 months) and budget.
- Development of reusable utilities for FTP, email, Scheduler integration, and SharePoint connections.

- A Python utility for sort, which is delivering savings of at least \$50,000 per year in licensing fees.

Complex, inadequately documented mainframe systems can be efficiently and effectively modernized to the AWS cloud infrastructure. To succeed, we believe a deep-dive discovery using automated tools and performing a complex proof of concept at the beginning is essential. Once that’s complete, an agile approach is required to migrate the application quickly to the cloud. Active engagement from application SMEs is extremely important to the modernization process, too.

To learn more, visit our [AWS page](#) or [contact us](#).



Industries

Automotive	Healthcare
Banking	Information Services
Capital Markets	Insurance
Communications, Media & Technology	Life Sciences
Consumer Goods	Manufacturing
Education	

Services

Application Services & Modernization	Digital Experience
Artificial Intelligence	Digital Strategy
Business Process Services	Enterprise Application Services
Cloud Solutions	Infrastructure Services
Core Modernization	Intelligent Process Automation

Consulting

Insights

Themes	Research
Modern Business	Deep Green
Sustainability and Resilience	Future-Ready Cities
Tech to Watch	The Future of Us
	The Future-Ready Business Benchmark
	The Metaverse
	The Work Ahead

About Cognizant

Annual Report
Be.Cognizant
Board of Directors
Client Briefing Program
Contact Us
Corporate Governance
Culture & Values
Diversity & Inclusion
ESG
Leadership Team
News & Press Releases
Partnerships
Public Policy
Sponsorships
Talent Worldwide

Glossary

Information for Suppliers

Oil & Gas
Retail
Transportation & Logistics
Travel & Hospitality
Utilities
Internet of Things
Quality Engineering & Assurance
Security
Software Product Engineering
Sustainability Services