Legacy z/OS Modernization with AWS - A Hybrid Approach for a Payment Routing System

Suddhasvatta Das[*], Abhijit Chakraborty[†]

Abstract: Legacy mainframe-based payment routing systems face increasing challenges, including high operational costs, slow processing speeds, and difficulties adapting to modern digital payment ecosystems. To ensure scalability, security, and regulatory compliance, financial institutions must modernize these systems without disrupting real-time transaction flows. This paper explores how AWS enables the seamless transformation of payment routing infrastructure through a hybrid architecture, i.e., using AWS and in-house mainframes. A real-world case study illustrates how a financial institution successfully transitioned to AWS, improving efficiency while maintaining business continuity. The paper concludes with a strategic roadmap for banks and payment providers to modernize payment systems while mitigating risk and ensuring regulatory alignment..

1 Introduction

Legacy mainframe systems have underpinned banking and finance for decades, valued for their reliability, scalability, and security in core banking, transaction processing, fraud detection, and compliance management [1]. Despite modern computing advancements, mainframes remain indispensable for handling high transaction volumes and ensuring stability [2]. However, modernization is now essential due to evolving technology and regulations. This paper explores how AWS complements mainframes, enhancing agility, cost efficiency, and innovation while retaining legacy strengths.

A critical mainframe component is z/OS, IBM's high-performance enterprise OS, designed for mission-critical workloads with exceptional reliability and security [3]. Widely adopted across industries requiring secure, real-time processing [4], z/OS became integral to banking due to its transaction-handling capabilities and regulatory compliance [5]. Its architecture supports multitasking, uptime, and data integrity, reinforcing its status as the backbone of financial infrastructure.

Mainframes were a strategic necessity in banking, providing transactional reliability for core banking, payments, and fraud management [3]. Proven through decades of evolving demands, they offer resilience against downtime and breaches [4]. Even today, mainframes process millions of secure transactions daily in regulated environments [6], ensuring their continued relevance. Despite this, mainframes face adaptability challenges in a digital-first economy. High maintenance costs, reliance on outdated skills like COBOL, and integration difficulties with cloud-based tools limit innovation [7]. Additionally, they lack elastic scalability for peak demand and native support for AI and ML technologies [2]. Addressing these limitations while preserving legacy strengths requires a strategic modernization approach [1].

AWS provides solutions that extend mainframe capabilities through hybrid cloud integration [3]. Tools like AWS batch, fargate, step, etc [6]. Hybrid adoption enables financial institutions to scale efficiently, meet regulatory demands, and leverage AWS for AI, analytics, and customer experience improvements [5]. This reduces costs while ensuring operational continuity and future-ready digital transformation [7].

This paper examines the opportunities and strategies for banking and finance organizations to modernize their legacy mainframe systems using Amazon Web Services by exploring a real-world example. The paper outlines how institutions can achieve a balanced hybrid approach that leverages both technologies' strengths to drive innovation, efficiency, and resilience in an increasingly competitive landscape.

[*]sdas76@asu.edu
[†]achakr40@asu.edu

2 Case Study

This section presents an example case of a legacy mainframe system that can be modernized using AWS. It highlights how to modernize traditional mainframes by leveraging AWS services. By adopting a hybrid approach, the case study demonstrates the integration of AWS with existing mainframe systems, preserving reliability and security while enhancing scalability, cost-efficiency, and innovation. The application is a bank's payment routing system. It handles cash and funds-related payments and is an intermediary for validating and routing payments to ensure seamless and accurate processing. The system communicates with external entities to validate transactions (as needed) before forwarding them to an accounting platform for further processing. Its ability to integrate with multiple bodies and handle diverse payment types makes it a complex yet integral part of the organization. The system supports multiple payment origination channels. For cash and funds payments, inputs can be received through batch files processed once or multiple times daily, online user entry through a front-end interface, or manual banker entry. Similarly, funds payments are submitted via batch files or banker entries. These flexible input methods accommodate various operational workflows and ensure the system can handle diverse submission formats effectively.

Payments progress through clearly defined states, from "ready" to "verified (ready to be picked by the accounting platform's mainframe)," ensuring a systematic and traceable life cycle. At the end of each business day, the system performs extensive batch processing to finalize all transactions, settle payments, and generate comprehensive reports, including financial ledgers. These operations ensure that all transactional data is reconciled and accounted for, providing a reliable financial snapshot of the day's activities.

Security, speed, and accuracy are critical requirements for the system, given its role in handling sensitive financial data and large transaction volumes. The existing system relies on an in-house mainframe and DB2 database. Quarterly, older data is offloaded and archived to free up space while maintaining accessibility for historical analysis. The system's ability to manage high transaction throughput, maintain precision in financial calculations, and provide high availability underpins its value to the organization. Furthermore, other applications perform analytics and apply artificial intelligence to customer insights and transaction patterns, positioning the system for enhanced operational efficiency and strategic innovation.

3 Road-map

Phase	**Objective**	**Time Estimate**
Assessment & Requirements Analysis	Analyze mainframe dependencies, workflows, and compliance for AWS migration.	2–3 months
Architecture Design & AWS Setup	Establish hybrid architecture, integrate AWS with DB2, ensure BFSI compliance.	3–4 months
Workflow Migration	Incrementally migrate business logic and batch processes to AWS.	6–8 months
Testing & Benchmarking	Validate system performance, accuracy, and regulatory compliance.	2–3 months
Deployment & Optimization	Deploy AWS workflows, optimize performance, and phase out mainframe workloads.	1–2 months

Table 1: Hybrid Cloud Migration Plan

4 Proposed Architecture Solution

We've integrated AWS services [8] to develop this solution while retaining the in-house DB2 database as the central operational data store. This hybrid architecture ensures data residency compliance and leverages the scalability and efficiency of cloud-native services. Two synchronized DB2 production regions operate parallel, using IBM Q [9] Replication to maintain backup. Note: There are two assumptions: 1) the z/OS COBOL codes need to be compiled as micro-focus COBOL, and 2) the entire Job Entry Subsystem (JES) tasks have been replaced using AWS services (EventBridge, ECS, Fargate, Lambda, SQS) for batch and online payment workflows.

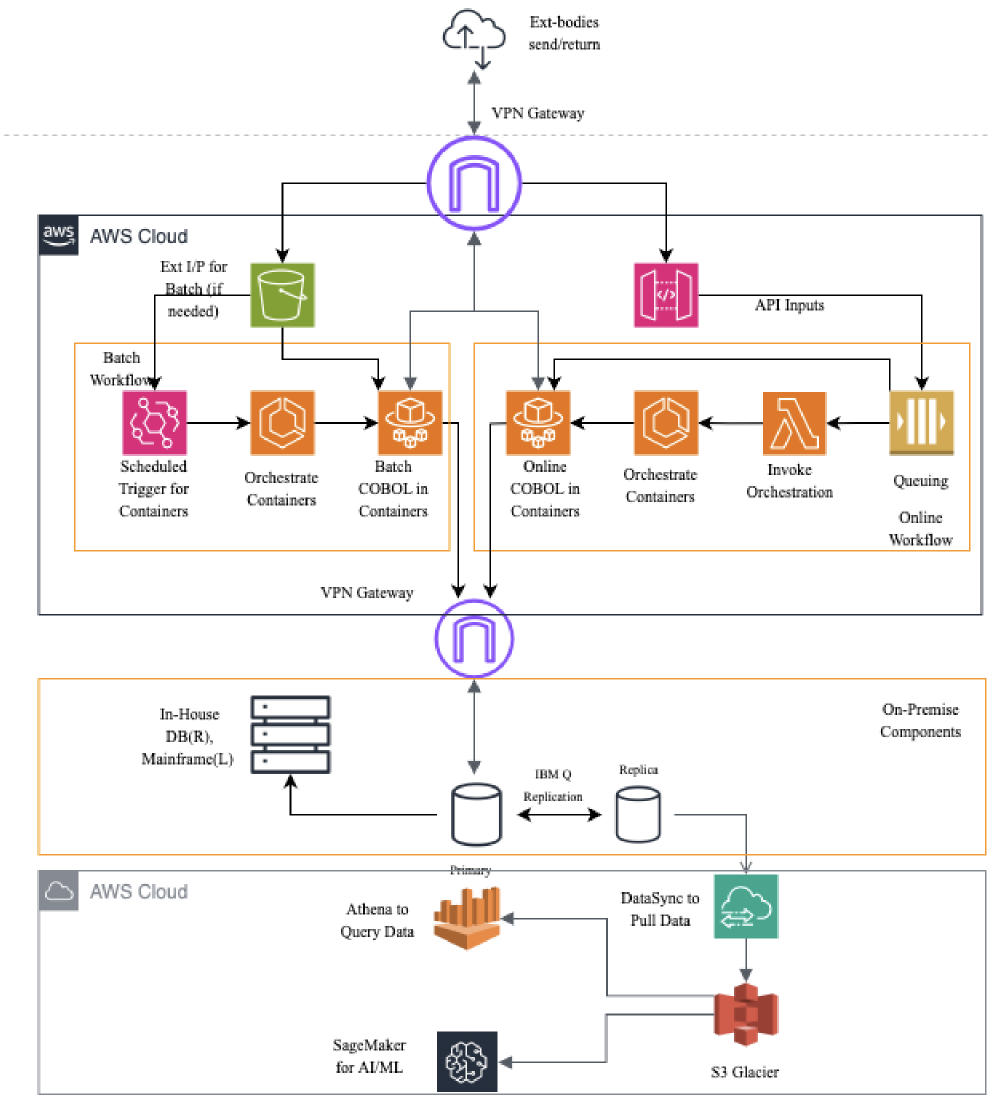

Figure 1: Solution Architecture

5 Cost Benefit

Aspect	Current	Proposed	Source
Annual Costs	1M–1.5M, h/w maintenance, licensing, energy	400K–500K for compute, storage, analytics, and monitoring	[8.9]
Billing Model	Fixed annual contracts	Pay-as-you-go for elasticity, with reserved instances for predictable workloads	[10]
Scalability	Fixed capacity; scaling needs significant h/w cost	Elastic infrastructure; services scale dynamically with workload	[11]
Resiliency	Centralized architecture; prone to bottlenecks and downtime	Distributed, fault-tolerant to ensure high availability	[8]
Migration Costs	NA	300K–500K one-time cost for development, testing, and integration	[12]
Energy Efficiency	High costs: approx 100K per year	AWS offloads computations, reducing energy cost	[11]

6 Conclusion

In conclusion, this paper has demonstrated how AWS can be integrated with existing mainframes to enhance transaction processing, optimize infrastructure, and introduce automation without disrupting mission-critical workloads. The proposed solution leverages Amazon EC2 Auto Scaling for dynamic workload management, AWS Batch for executing batch processing jobs, and AWS Step Functions for workflow orchestration. Amazon S3 Glacier provides cost-efficient archival storage, AWS Athena enables querying of historical data, and Amazon SageMaker powers AI-driven insights. A VPN Gateway enables secure, low-latency communication between AWS cloud, on-premise DB2, and mainframe systems, ensuring data consistency and operational continuity. The case study validates that a phased hybrid migration strategy ensures seamless modernization, allowing financial institutions to adopt cloud capabilities while incrementally maintaining existing mainframe reliability. The cost-benefit analysis further highlights that AWS-based hybrid solutions significantly reduce infrastructure costs, enhance resilience, and provide the scalability required to meet evolving regulatory and operational demands. AWS does not replace mainframes; it extends and optimizes them. Implementing this structured hybrid modernization approach will achieve greater agility, lower operational costs, and improved performance, ensuring their continued competitiveness in an increasingly digital financial ecosystem.

7 Acknowledgment

This work acknowledges the assistance of a Large Language Model (LLM) used explicitly for grammar and spelling-checking purposes. The author has reviewed all content to ensure accuracy, integrity and correctness.

8 References

1. KR Gade, *Migrations: Cloud Migration Strategies, Data Migration Challenges, and Legacy System Modernization*, Journal of Computing and Information, 2021.
2. C Lekkala, *Modernizing Legacy Data Infrastructure for Financial Services*, International Journal of Science and Research (IJSR), 2021.
3. B Dash, *Life on the Edge from Legacy to Cloud Computing: A Case Study on Insurance Industry*, SSRN, 2020. Link
4. P Raj and A Chaudhary, *Cloud-Native Computing: How to Design, Develop, and Secure Microservices and Event-Driven Applications*, 2022.
5. ME Rana and VA Hameed, *Revolutionizing Finance: The Transformative Impact of Cloud Computing in Finance Shared Service Center (FSSC)*, IEEE, 2023.
6. AWS, *AWS Mainframe Modernization.*
7. E Øvrelid and Z Wang, *Restructuring Digital Infrastructures: Architectural Transformation through Sociotechnical Interplay*, Norsk IKT-konferanse, 2024.
8. AWS, *AWS Overview Handbook*
9. `https://www.ibm.com/docs/en/idr/11.4.0?topic=po-q-replication`
10. Kaya, F., Van Den Berg, M., Wieringa, R., & Makkes, M. (2020, June). *The banking industry underestimates costs of cloud migrations.* In 2020 IEEE 22nd Conference on Business Informatics (CBI) (Vol. 1, pp. 300-309). IEEE.
11. `https://aws.amazon.com/ec2/pricing/reserved-instances/`
12. Fernando, C. (2022). *Building Enterprise Software Systems with Hybrid Integration platforms.* In Solution Architecture Patterns for Enterprise: A Guide to Building Enterprise Software Systems (pp. 109-146). Berkeley, CA: Apress.
13. Hosseini Shirvani, M., Rahmani, A. M., & Sahafi, A. (2018). *An iterative mathematical decision model for cloud migration: A cost and security risk approach.* Software: Practice and Experience, 48(3), 449-485.