

Mainframe Modernization Strategies

Malik Syed

Systems Integration Consulting, CGI, Pittsburgh, USA

Abstract This article is based on a thorough research and analysis of different solutions available to modernizing mainframe. It stresses that rather completely replacing legacy core systems, businesses are increasingly looking to link them to emerging technologies using innovative new connectors so that each family of systems can do what it does best. This article presents the details of analysis, planning, design, and implementation steps of two such available solutions i.e., re-platforming and refactoring for the successful mainframe modernization. It also highlights pros and cons of re-platforming and refactoring and even touches other available solutions that help businesses make a right decision. The technical design section summarizes the required skills and lays out the steps from planning to implementation using an example of Amazon Web Services (AWS) mainframe modernization solution. Finally, the article concludes summarizing the importance, potential benefits, results, and success stories of mainframe modernization.

Keywords Mainframe Modernization, Re-platforming, Refactoring, Legacy Systems, COBOL, Cost Savings

1. Introduction

1.1. Overview

Mainframe modernization refers to the process of updating legacy mainframe systems to meet modern business needs using modern technology. While some mainframe manufacturers continue to invest in their platforms to incorporate latest capabilities like integrations using DevOps, many other mainframe customers have stopped investing in upgrading their platforms. Modernizing a mainframe can be a complex process, however it is essential to keep up with the never-ending and ever-evolving technology landscape. It involves migrating applications and data from outdated mainframe systems to modern platforms, such as cloud computing, and integrating them with emerging technologies like artificial intelligence, machine learning, and blockchain. It is done to improve the efficiency, agility, and flexibility of the system while reducing operational costs.

1.2. Problem Statement

There are several challenges that organizations face when modernizing their mainframe systems. According to a study by IBM, the lack of required resources and skills is the top challenge faced by businesses pursuing mainframe application modernization. Maxima Consulting lists five mainframe modernization challenges that organizations should consider in their strategy: [1]

- Thinking that the technology alone will solve the problem.
- Integrating database administration into modern DevOps practices.
- Identifying, and reducing shadow IT projects.
- Improving tool standardization.
- Avoiding poorly directed modernization efforts.

1.3. Challenges

Mainframe modernization is not a simple or straightforward task. It requires careful planning, execution, and evaluation to ensure a successful outcome.

There are many challenges that can affect the decision-making process for mainframe modernization. Some of the key challenges that can influence your decision making for mainframe modernization are:

Business goals and outcomes: Mainframe modernization should align with the strategic objectives and vision of the organization. It should also address the pain points and challenges that the mainframe system poses to the business, such as cost, complexity, latency, or lack of agility. Mainframe modernization should enable business transformation by delivering value-added services, enhancing customer experience, increasing operational efficiency, and fostering innovation.

Application characteristics and dependencies: Mainframe modernization should consider the size, complexity, heterogeneity, and criticality of the applications that run on the mainframe system. These factors will determine the appropriate modernization approach and platform selection for each application. For example, some applications may require a full rewrite or rehosting on a different platform, while others may benefit from incremental updates or enhancements on the same platform.

* Corresponding author:

syed.mallik@gmail.com (Malik Syed)

Received: Jan. 10, 2024; Accepted: Jan. 29, 2024; Published: Feb. 5, 2024

Published online at <http://journal.sapub.org/ajca>

Skills and resources availability: Mainframe modernization requires a skilled and experienced team that can handle both technical and business aspects of the project. The team should include developers, architects, testers, analysts, managers, end users, and stakeholders who have a deep understanding of both mainframe technologies and business processes. The team should also have access to adequate resources such as budget, time frame, tools, infrastructure, security measures, and support services.

Cultural change readiness: Mainframe modernization involves adopting new ways of working that are different from the traditional ways of maintaining legacy systems. This requires a shift in mindset from being risk-averse to being risk-tolerant, from being siloed to being collaborative, from being reactive to being proactive, from being rigid to being flexible, from being manual to being automated, from being closed to being open.

1.4. Proven Procedures for Migration

The simple yet effective proven procedures to modernize the mainframe are:

- Structured migration planning
- Technical analysis of legacy systems
- Process analysis of legacy systems
- Design of the target architecture
- Cloud enablement
- Migration reference scenarios
- Data migration
- Tools and processes
- Modernization by means of a central factory
- Test automation and regression testing

2. Why to Modernize Mainframe

Below are a few from several reasons that tells why to choose mainframe modernization.

Cost Savings: Reduce costs by optimizing the mainframe usage, reducing maintenance costs, and eliminating the need for expensive hardware upgrades. It helps reduce the cost of maintaining legacy systems by migrating them to modern technology ultimately help organizations save money on hardware, software, and staffing costs. According to Gartner, the cost of maintaining legacy mainframe applications can be as high as 70% of an organization's IT budget.

Improved Performance: Improve performance by optimizing the applications, improving data processing speeds, and reducing downtime.

Improved Efficiency: Improve the efficiency by making the systems more agile and flexible. This helps organizations respond more quickly to changing business needs and improve overall productivity.

Improved Agility: Improve agility by enabling systems to quickly respond to changing business needs, integrate new technologies, and improve collaboration across teams.

As enterprises adopt newer technologies, such as cloud computing and artificial intelligence, modernized mainframes can integrate seamlessly with these technologies.

Improved Security: Mainframe applications have always been known for their security and reliability. However, as cyber threats become more sophisticated, it is essential to maintain the safety of these systems. Modernized mainframes can integrate with newer security technologies, such as encryption and biometric authentication, providing businesses additional protection against cyber threats.

Risk Mitigation: Mitigate the risk of system failures and downtime by upgrading legacy systems to modern technology. This helps organizations avoid costly downtime and ensure business continuity.

Enhanced Security: Enhance security by improving data encryption, reducing the risk of data breaches, and ensuring compliance with industry regulations. It helps improve their security by upgrading systems to modern security standards. This helps organizations protect sensitive data and ensure compliance with industry regulations.

Enhanced Scalability: Businesses scale their systems to meet increasing demands for data processing. With the growth of big data, businesses need to process vast amounts of information quickly and efficiently. Modernized mainframes can provide the scalability required to handle these demands, improving overall system performance.

Better Customer Experience: Improve customer experience by providing faster, more reliable services, and enabling to offer new services and features.

Access to New Features and Capabilities: Modernizing mainframes can provide businesses with access to new features and capabilities that were previously unavailable. For example, modern mainframe applications can take advantage of newer programming languages and technologies, improving the maintainability of applications. Additionally, modern mainframes can integrate with newer technologies, such as machine learning, allowing businesses to take advantage of the latest advancements in the data processing.

Retaining Critical Business Knowledge: The qualified talent pool is shrinking as the workforce responsible for maintaining mainframes ages and retires. Modernized mainframes can provide better documentation and automated processes, improving knowledge retention across the organization.

3. Ways to Modernize Mainframe

3.1. Different Solutions

Different solutions to mainframe modernization include APIfication, Native Microservices Development, Re-Platform, Re-Factor, and Rebuild/Re-Write/Re-Imagine. Each approach has its own advantages and disadvantages and hence organizations must carefully consider their goals and resources before choosing a specific modernization approach. [2]

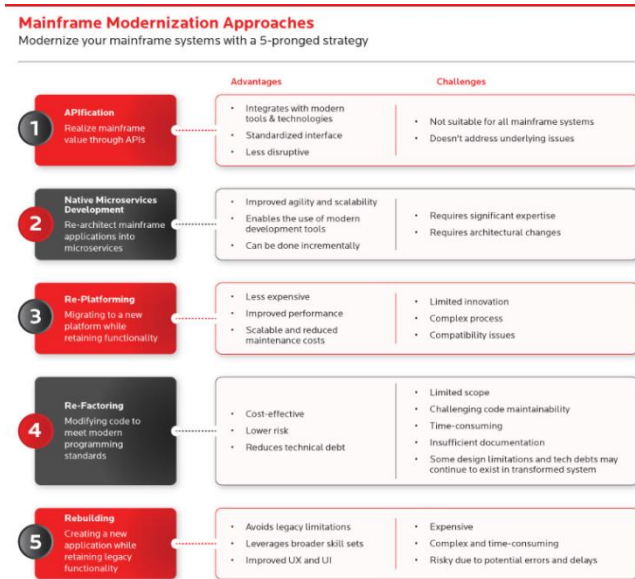


Figure 1. Mainframe Modernization Approaches

3.1.1. APIfication

APIfication exposes mainframe data and functions as APIs that other systems can access and consume. This approach enables organizations to unlock the value of their mainframe applications without needing a full-scale modernization effort. By APIfying mainframe applications, organizations can leverage existing systems and reduce the complexity and cost of integration with modern systems.

Pros:

- APIfication allows organizations to integrate mainframe applications with modern tools and technologies, such as cloud services and mobile applications, while maintaining the integrity and security of the mainframe data.
- APIs provide a standardized interface that allows different systems to communicate effectively and can be developed quickly and efficiently.
- This approach can be less disruptive than other modernization approaches as it does not require significant changes to the mainframe system.

Cons:

- APIfication may not be suitable for all types of mainframe applications, as some legacy systems may be unable to provide data and functions in a modern, API-friendly format.
- Additionally, this approach may not address underlying issues with the mainframe system, such as outdated programming languages or difficult maintainability.

3.1.2. Native Microservices Development

Native Microservices Development is a modernization approach that involves re-architecting mainframe applications to adopt a microservices-based architecture. This approach provides more agility and flexibility in development and

deployment, allowing organizations to respond more quickly to changing business needs. It breaks down monolithic mainframe applications into more minor, independent services, which can be developed and deployed independently.

Pros:

- Native Microservices Development can improve the agility and scalability of mainframe applications, allowing organizations to respond more quickly to changing business needs.
- It can also enable modern development tools, such as containerization and orchestration technologies, which can increase development speed and reduce costs.
- This approach can be less disruptive than other modernization approaches as it can be done incrementally, with individual services migrated over time.

Cons:

- Native Microservices Development requires significant expertise in modern development technologies and mainframe applications, making it a more challenging approach for some organizations.
- It may also require changes to the mainframe system's architecture, which can be difficult and time-consuming.
- Additionally, it may not address underlying issues with the mainframe system, such as outdated programming languages or difficult maintainability.

3.1.3. Re-Platforming

Re-platforming is migrating a legacy application to a new platform while retaining its existing functionality. This approach typically involves migrating mainframe applications to a modern server-based infrastructure like Cloud or virtualized environment. It enables organizations to maintain the business logic and processes built into the original mainframe applications while leveraging new technology for better performance, scalability, and cost-effectiveness.

Pros:

- Re-platforming is less expensive than other modernization approaches, requiring fewer modifications to the existing application code.
- Moving to a modern infrastructure can improve the performance of the application.
- The re-platformed application can be scaled more quickly than the legacy system.
- Organizations can reduce the maintenance costs associated with a legacy mainframe system by migrating to a modern infrastructure.

Cons:

- Re-platforming does not allow for significant innovation in the application. The application's functionality remains unchanged and might miss out on modern business processes.
- It can be a complex process requiring significant planning and testing to ensure a smooth transition.

- Compatibility issues may arise when migrating to a new platform. The application may not work as expected, and additional modifications may be necessary.

3.1.4. Re-Factoring

Refactoring is the process of restructuring the code combining reverse engineering with forward engineering for enabling better performance and portability making your applications significantly faster in future. The methodology retains the business logic and core functionality, freeing up from mainframe-based Cobol & is predominantly carried out using automated tools that are available for most languages, databases, VMs and platforms.

Pros:

- Refactoring can be a cost-effective approach to modernizing legacy mainframe applications. It enables organizations to enhance the performance and reliability of the system without the need to rewrite it entirely.
- It is a lower-risk approach compared to a complete system rebuild. Since the core business logic of the system remains unchanged, there is less risk of introducing new bugs or issues into the system. This can result in a smoother and less risky modernization process.
- It reduces organizations' technical debt. By eliminating unnecessary code and optimizing the existing code, organizations can reduce the complexity and improve the system's overall quality, making it easier to maintain and upgrade.

Cons:

- Refactoring has a limited scope and may not be suitable for all legacy systems. It is only effective in systems where the core business logic remains unchanged, and the system can be improved by optimizing the existing code.
- Although tools are available for refactoring, readability and code maintainability can be challenging.
- It can be a time-consuming process, requiring a significant amount of effort and resources to achieve the desired results.
- Legacy mainframe applications may not have sufficient documentation, making refactoring more challenging and time-consuming. Developers may need to spend additional time understanding the existing system's functionality and design before they can start refactoring.

3.1.5. Rebuilding

This approach involves creating a new application that meets the requirements of modern business processes while retaining the functionality of the legacy application. In this article, we will discuss the advantages and challenges of this approach.

Pros:

- By starting from scratch, organizations can avoid the

limitations and constraints of the legacy system and build a new system better suited to their current and future needs.

- Rebuilding can also open up opportunities for organizations to leverage broader skill sets and expertise. With modern technologies and development practices, organizations can attract a new generation of developers skilled in the latest programming languages, frameworks, and tools.
- It can improve the system's user experience (UX) and interface (UI). With a new system, organizations can design and develop an intuitive and user-friendly interface that meets the needs of modern users. Additionally, a new system can be more agile and responsive, enabling organizations to adapt to changing business requirements quickly.

Cons:

- It can be the most expensive of all modernization approaches. Developing and testing the new application involves significant resources, time, and investments. The costs of the latest application need to be weighed against the potential benefits to ensure a positive return on investment.
- Rebuilding a legacy mainframe system from scratch can be complex and time-consuming. It requires significant planning and testing to ensure a smooth transition from the old system to the new one. Additionally, the new application may have compatibility issues with other systems, requiring further modifications to the code.
- It carries some risk due to potential errors and delays during development. Organizations need to manage this risk by having a robust testing and quality assurance process to ensure that the new application meets the business requirements.

3.2. Solution Partners

Please note that the below solution partners are just a few examples of the many solutions available for mainframe modernization. The best solution for any organization depends on the specific needs and requirements.

Google: Google Cloud offers mainframe modernization solutions with automation, industry expertise, risk-mitigated migration tools, and a rich partner ecosystem. They provide a free migration assessment and a modernization solution for mainframe and midrange applications called G4. You can learn more about their offerings and benefits by visiting their website. Google Cloud offers a comprehensive mainframe modernization solution that includes automation, industry expertise, risk-mitigated migration tools, and a rich partner ecosystem.

Microsoft: Microsoft Azure provides mainframe and midrange migration tools and services to transform your mainframe workloads on Azure. They have a large ecosystem of Microsoft partners that specialize in mainframe migration tools and services.

Microsoft Azure Logic Apps also offers a solution for mainframe and midrange modernization. Their solution includes ways to maintain the current service level indicators and objectives for your environments, manage coexistence between legacy data along with migrated data, manage application interdependencies, and define the future of the mainframe scheduler and jobs.

Microsoft Azure Logic Apps also provides a mainframe modernization solution that includes ways to maintain the current service level indicators and objectives for your environments, manage coexistence between legacy data along with migrated data, manage application interdependencies, and define the future of the mainframe scheduler and jobs.

Amazon Web Services: AWS Mainframe Modernization service is a unique platform that allows you to migrate and modernize your on-premises mainframe applications to a cloud native fully managed runtime environment on AWS. AWS Mainframe Modernization service is another platform that allows you to migrate and modernize your on-premises mainframe applications to a cloud-native fully managed runtime environment on AWS3.

IBM: IBM offers a range of solutions to modernize mainframe applications and data. The IBM Z and Cloud Modernization Center provides expert resources, guidance, and capabilities to accelerate transformation in your IBM Z environment1. IBM consultants can help solve application modernization challenges with the mainframe, using IBM Cloud and IBM zSystems to improve agility, optimize costs, and lower risk2. IBM's approach focuses on the incremental modernization and optimization of the mainframe landscape, helping you unleash the full potential of the mainframe by modernizing existing mainframe apps and mainframe data and extending them to the cloud. IBM watsonx Code Assistant for Z is a generative AI-assisted product built to accelerate mainframe application modernization at lower cost and with less risk than today's alternatives. This product provides an end-to-end application developer lifecycle that includes application discovery and analysis, automated code refactoring and COBOL to Java conversion. Developers can automatically refactor selected elements of an application and continue modernizing in COBOL, or selectively transform code to Java leveraging generative AI using a highly tuned state-of-the-art large language model.

BMC: BMC AMI is a part of the BMC Automated Mainframe Intelligence (AMI) portfolio, which is designed to help organizations modernize their mainframe environments and improve their overall efficiency and agility. BMC AMI DevX Code Insights and BMC AMI zAdviser Enterprise are two new mainframe modernization solutions that are rapidly transforming the mainframe into a true first-class citizen of the DevOps lifecycle.

4. Steps to Modernize

4.1. Address the Challenges

Ageing, legacy mainframe technologies introduce several

challenges. Inefficiencies can end up being very costly for organizations, with troubleshooting overly complicated by applications with hidden or buried code. Redundant code can also cause the mainframe to become overloaded, impacting performance. One of the challenges is that increasing uptake of digital services namely online shopping or the move to cashless transactions (i.e. contactless) increases the load on the mainframe, pushing up the cost and increasing the operational risk.

Despite the mainframe being a critical backbone of their business, many organizations often lack the specific skills and expertise needed to maintain and manage their mainframe technology. Skilled mainframe programmers are now reaching retirement, with mainframe programming languages not as popular with developers now. Without a sense of what's needed to update or maintain mainframe code, or the lack of clear strategy around what their options are, developers may be confused about the direction they should be moving in with their mainframe.

Mainframes have been reliable and efficient, serving critical functions for large organizations, including banks, insurance companies, and governments. However, as the IT landscape evolves rapidly, mainframe applications face several challenges.

High cost of hardware/storage: One of the primary challenges of mainframes is the high cost of hardware and storage. Mainframes are designed for high performance and reliability, which requires specialized hardware and storage devices. These devices can be costly, and as data storage needs increase, so does the cost. Additionally, mainframes often require expensive software licenses and maintenance fees, making it difficult for businesses to justify the cost.

Shrinking talent pool of mainframe expertise: Many experts who have mastered this technology are aging and retiring, leaving a gap in the workforce. Finding new talent with the skills to maintain and enhance mainframe applications is becoming increasingly difficult, creating a shortage of mainframe experts that can impact the ability of companies to keep up with evolving technology and business needs.

Lack of business agility: Mainframes are known for their stability and reliability but are not typically associated with agility. Updating or changing mainframe applications can be time-consuming and complex. This can slow down a company's ability to respond to changing business needs or market trends. Mainframes tend to use older programming languages and software development practices, making integrating with newer technologies and platforms difficult.

Difficult maintainability: Maintaining mainframe applications can be complex and time-consuming, requiring specialized knowledge and expertise. With the complexity of modern business environments, keeping up with the maintenance and updates needed can be challenging, particularly when companies have limited access to mainframe expertise. The lack of standardization in mainframe applications also makes maintenance and updates more challenging.

Lack of documentation: Many mainframe applications

have been operating for decades, and the original document may have been lost or outdated. This can make it difficult for new developers or IT staff to understand how the system works, leading to more extended learning curves and potentially costly errors. Maintaining and updating mainframe applications can become even more challenging without adequate documentation.

Outdated programming language: Mainframe applications often use programming languages that are no longer in widespread use, making it difficult to find developers with the required expertise. Additionally, it can be challenging to integrate mainframe applications with modern applications and platforms. Using outdated programming languages can also make adopting newer software development practices, such as Agile and DevOps, challenging.

Lack of Agile and DevOps processes: Mainframe development has historically followed a waterfall approach, with long development cycles and extensive testing. This can make it challenging to keep up with the fast pace of modern software development. Adopting Agile and DevOps processes can help organizations to deliver more value to their customers faster, but integrating these processes with existing mainframe applications can be challenging. The lack of standardization and documentation in mainframe applications can make it difficult to automate testing and deployment processes, slowing the adoption of Agile and DevOps.

4.2. Making a Solid Case for Mainframe Modernization

As businesses grow and evolve, so must their technology infrastructure. This is particularly true for legacy mainframe applications, which have been the backbone of business operations for decades. However, as technology evolves rapidly, these legacy systems are becoming expensive and challenging to maintain.

According to a report by Gartner, many enterprises struggle to determine the appropriate placement of their mainframe in their overall IT strategy. As a result, they may keep the mainframe without fully utilizing its capabilities or fail to allocate workloads to other platforms where they are better suited. This can lead to both inertia and hindrance of digital initiatives.

4.3. Roadmap to Modernization

Please note that the below steps are just a few examples of the many steps involved in mainframe modernization. The best steps for your organization depend on your specific needs and requirements.

Assess the current state: The first step in mainframe modernization is to assess the current state of the system. This includes identifying the applications, data, and infrastructure that need to be modernized, as well as the business processes that are dependent on the mainframe.

Define the future state: The next step is to define the future state of the system. This includes identifying the target architecture, the applications that need to be modernized, and the infrastructure that will be used to support the modernized system.

Develop a roadmap: Once the future state has been defined, the next step is to develop a roadmap for modernization. This includes identifying the steps that need to be taken to achieve the future state, as well as the resources that will be required to complete the modernization process.

Choose a modernization strategy: After developing a roadmap, the next step is to choose a modernization strategy that is best suited for the organization's needs. This could include cloud migration, refactoring, emulation, or rehosting.

Implement the modernization strategy: Once a modernization strategy has been chosen, the next step is to implement it. This includes migrating applications to the cloud, refactoring legacy code, or rehosting applications on a new platform.

Test and validate: After implementing the modernization strategy, the next step is to test and validate the modernized system. This includes testing the applications, data, and infrastructure to ensure that they are working as expected.

Monitor and optimize: The final step in mainframe modernization is to monitor and optimize the modernized system. This includes monitoring the performance of the system, identifying areas for improvement, and optimizing the system to ensure that it is running efficiently.

Deep diving into mainframe modernization approaches: Mainframe applications have been the backbone of large-scale enterprise computing for decades, but the need for modernization becomes more apparent as technology evolves. Legacy systems can impede an organization's agility, efficiency, and competitiveness. Mainframe modernization is a strategic process of updating these legacy systems to meet current business needs.

4.4. Architecture Diagrams

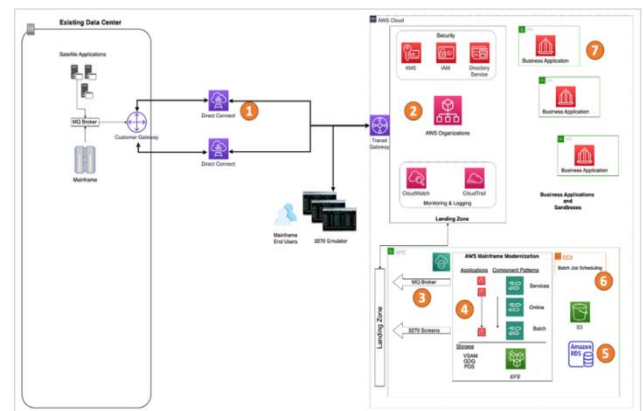


Figure 2. AWS target architecture for Mainframe Modernization [3]

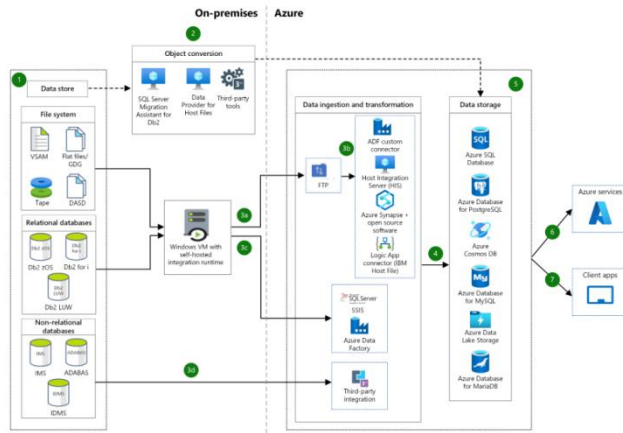


Figure 3. Microsoft Azure target architecture for Mainframe Modernization [4]

5. Refactoring and Re-platforming Explanation Using AWS

5.1. Refactor with AWS Blu Age

AWS Blu Age powers the automated refactoring pattern within AWS Mainframe Modernization service to accelerate the modernization of the mainframe applications complete software stack including application code, dependencies, and infrastructure. Modernized applications implement best practices and design patterns, leveraging industry standards and cloud services to derive business components and macro services. Applications modernized by Blu Age balance legacy constraints and cloud benefits to ensure critical applications meet or exceed scalability and agility requirements for simplified maintenance and operations aligned to business needs.

AWS Blu Age Refactor automatically creates modern applications from legacy monolithic mainframe or midrange source code. Applications in COBOL, generated COBOL, PL/I, NATURAL, RPG/400, COBOL/400 and their respective underlying databases and data files DB2, DB2/400, VSAM, IMS, IDMS are transformed into modern distributed applications relying on Angular, Java/Spring, PostgreSQL or other databases such as Amazon Aurora, RDS for PostgreSQL, Oracle database, IBM Db2.

Benefits:

• Boost application agility

The Blu Age modernized application stack implements DevOps best practices with newer language and frameworks increasing agility both at the application level and at the infrastructure level.

• Increase talent pool

With popular language and technologies, the Blu Age-refactored applications can be maintained and enhanced by numerous Java developers and IT professionals making it easier and quicker to onboard talented professionals.

• Accelerate modernization project

The mature Blu Age automation guarantees functional and performance equivalence increasing the speed and quality of migration and modernization projects.

Use cases:

Customers adopt Blu Age refactor pattern to leave legacy languages and dependencies and automatically transform their applications with newer language, web frameworks, and interfaces.

Use popular tools and vibrant ecosystem for innovation.

The modernized application and infrastructure stack allows using popular tools for development, integration, security, and leveraging a wide choice of solutions options for quickly building innovations.

Create macro services or microservices for more agility.

Blu Age toolchain creates service-oriented and object-oriented applications and facilitates the modernization towards granular services for business efficiencies.

5.1.1. Source and Target Technology Mapping

Technology	Current Mainframe	Target Modernized
Programming Language	COBOL	Java
Database	VSAM	Oracle
Batch	JCL (Jobs/Procs)	Spring Batch XML
User Interface Screens	CICS BMS Maps	JSF, HTML/CSS, JavaScript
Security and Access Control	RACF	Spring Security and Microsoft Active Directory
Middleware, Web Services, Database Transactions	CICS	Jetty, Apache-CXF, JPA/EclipseLink, Modern Systems Framework
Reporting	DB2 and QMF	Oracle and Jasper Reports
Encryption	Megacryption	Java Cryptography Architecture
Screen Automation	IBM HATS	SOAP Web Services
Batch Process Scheduling	CA7	BMC Control-M
Monitoring and Alerting	RMF, SMF, Omegamon	New Relic, Nagios, Sumologic
Development and Deployment	Changeman, Endeavor	Git, Gradle, Jenkins, Puppet, Ansible

Figure 4. Refactoring - Source and Target Technology Mapping

5.2. Re-platform with Micro Focus

Analyse:

The first step is to analyse the application using Micro Focus Enterprise Analyzer to understand application complexity, dependencies, and the impact of moving these applications to the cloud. Micro Focus Enterprise Analyzer is available on-demand from the AWS Mainframe Modernization service console.

Develop:

The develop step creates a modern development environment and CI/CD pipeline using Micro Focus Enterprise Developer to enable you to maintain and modernize mainframe COBOL and PL/I applications using an Eclipse based environment for rapid development. Coupled with the application knowledge from Enterprise

Analyzer, developers can understand, develop, build and test application changes. Micro Focus Enterprise Developer is also available on-demand from the AWS Mainframe Modernization service console.

Deploy:

The deployment step automates the creation of tailored, high availability and elastic infrastructure using the AWS Mainframe Modernization managed runtime environment powered by Micro Focus Enterprise Server, a scalable production engine meeting and often exceeding enterprise requirements for reliability, availability, and serviceability.

Operate:

Finally, with the AWS Mainframe Modernization managed runtime, you can operate business critical systems whether this is performing system maintenance, applying application updates, monitoring the health of the system, or diagnosing problems.

Benefits:

• Reduced impact

Reduce the impact of moving your tried and trusted business applications to the cloud with minimal changes and meeting your performance, security, and availability requirements. Impact is greatly reduced and time to modernize is shortened by leveraging the AWS Mainframe Modernization managed service platform which handles many aspects of the development process and provides the elastic, high availability deployment infrastructure with just a few clicks.

• Modernization

Continue to modernize and evolve your applications, taking advantage of cloud services more easily accessible once the application has been re-platformed. Modernization projects are significantly more successful when change is made incrementally. Ongoing modernization efforts may include API enablement, microservices or containers.

• Development velocity

Enable faster development with modern tools and process, allowing you to be more responsive to change. Even if your applications remain on the mainframe for production, you can use the cloud for development and test. Application definitions can be maintained as source and can be plugged in a CI/CD process to automate the deployment of changes. Once an application has been deployed and is running in the AWS Mainframe Modernization runtime it can then be easily access through your existing 3270 interfaces by using Micro Focus Host Access for the Cloud, which is available for a free trial from the AWS Marketplace.

Use cases:

Infrastructure modernization for increased flexibility.

One of the key values of the cloud is the ability to spin up servers as demand increases. Re-platforming enables your existing applications to take advantage of the cloud's flexibility with minimal effort and risk.

Move to a single set of tools addressing skills concerns.

Once applications are moved to the cloud, your team will be able to manage and support mainframe applications in a familiar environment with a common set of modern tools.

CI/CD pipeline for applications running on a mainframe.

Even for applications that remain in production on a mainframe, re-platforming can help modernize the development and testing process. Your existing COBOL and PL/I applications can be developed using modern, eclipse-based development tools, tested on the cloud, and deployed to your mainframe.

6. A Few Success Stories to Mention

Here are some detailed success stories of mainframe modernization from companies: [5] [6] [7]

- The New York Times refactored their mainframe workload onto AWS. The newspaper mainframe was running a core-business billing and delivery workload that was expensive and difficult to evolve. The automated refactoring to AWS transformed the legacy application into a modern stack with an agile development and test life cycle. This provided 70% cost savings and an acceleration of application development and release cycles.
- Kmart Australia migrated their mainframe merchandise system to AWS. AWS partners completed the migration in record time and entirely remotely with a hard end-of-contract deadline. Kmart Australia estimates that in the next five years, the migration will save between \$30-\$40 million for the company, and unlock significant value for its teams, business partners, and customers.
- The US Air Force modernized a COBOL-based system running on aged mainframes. The migration to AWS with TSRI technology modernized the stack to an open platform and modern multi-tier architecture, while retaining all functionality. By leveraging Amazon Web Services (AWS), the U.S. Air Force introduced agility, reduced costs, modernized the technical stack, and enhanced security.
- Deloitte Insights reports that businesses have begun looking for ways to extend the functionality of mainframe systems by linking them to emerging technologies instead of completely replacing them. This approach has been successful in many cases, including that of Meuhedet, an Israeli health insurance and care provider, whose mainframe-based electronic medical record system continues to serve as an effective store of patient data.
- Infosys provides a range of case studies on mainframe modernization. One of the case studies is about a New Zealand-based insurance company that modernized their mainframe systems to reduce cost. The insurer reduced TCO by 25% with modernized administrative system.

- Unisys suggests that incremental modernizations make it easier to mitigate high-risk upgrades. One of the top advantages of incremental modernization is its ability to minimize limitations such as complexity, making it challenging to estimate project timelines and budgets accurately.
- Nagarro helped Aareal First Financial Solutions AG, a client specializing in IT solutions for payment transactions and account management, with their mainframe modernization project. The migration project was a big success.

7. Results and Discussion

According to a report by Deloitte Insights, businesses have begun looking for ways to extend the functionality of mainframe systems by linking them to emerging technologies instead of completely replacing them. This is because mainframes often host applications that can't be moved to the cloud due to the substantial work needed to refactor applications or too risky due to the possibility of breaking system dependencies. The report also suggests that enterprises are increasingly looking to link mainframes to emerging technologies using innovative new connectors so that each family of systems can do what it does best. [5]

Kyndryl commissioned a survey of 500 business leaders for a pulse on what the leaders and their teams think about mainframe transformation and application modernization. The survey is basis for Kyndryl's 2023 State of Mainframe Modernization Survey Report. [8]

IBM Institute for Business Value conducted a double-blind survey of 200 IT executives in North America in April 2023. The survey revealed that mainframe application modernization is key to accelerating digital transformation.

8. Conclusions

This paper concludes that even though the process of mainframe modernization can be complex and challenging, with careful planning and execution, organizations can successfully migrate to modern platforms and technologies which leads to reduce costs, improve efficiency, and gain a

competitive advantage by modernizing their strategies. Collaborating with some partners who have had already successfully modernized mainframe would be one of the success driving factors.

ACKNOWLEDGEMENTS

SAP Productions wishes to acknowledge all the contributors for developing and maintaining this template.

DISCLOSURE

This section is ONLY for those who requested disclosure. The name of the experts that reviewed your paper, in case they accepted selling disclosure to you, will appear here. Each reviewer is allowed to make their own price for that, since that is a public endorsement of your findings and may be used for varied purposes.

REFERENCES

- [1] <https://www.maximaconsulting.com/newsroom/mainframe-modernization-challenges>
- [2] Rohit Dhal, "5 Proven Approaches To Successfully Carry Out Mainframe Modernization", Principal Architect, Birlasoft, <https://www.birlasoft.com/articles/5-proven-approaches-to-successfully-carry-out-mainframe-modernization>.
- [3] <https://aws.amazon.com/blogs/apn/demystifying-mainframe-modernization-with-best-practices-from-aws-and-accenture/>
- [4] <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/mainframe/modernize-mainframe-data-to-azure>
- [5] <https://www2.deloitte.com/us/en/insights/focus/tech-trends/2023/future-mainframe-technology-latest-trends.html>
- [6] <https://aws.amazon.com/mainframe/>
- [7] <https://www.infosys.com/services/application-modernization/case-studies/leading-insurer.html>
- [8] Kyndryl 2023 State of Mainframe Modernization Research Report | Kyndryl.