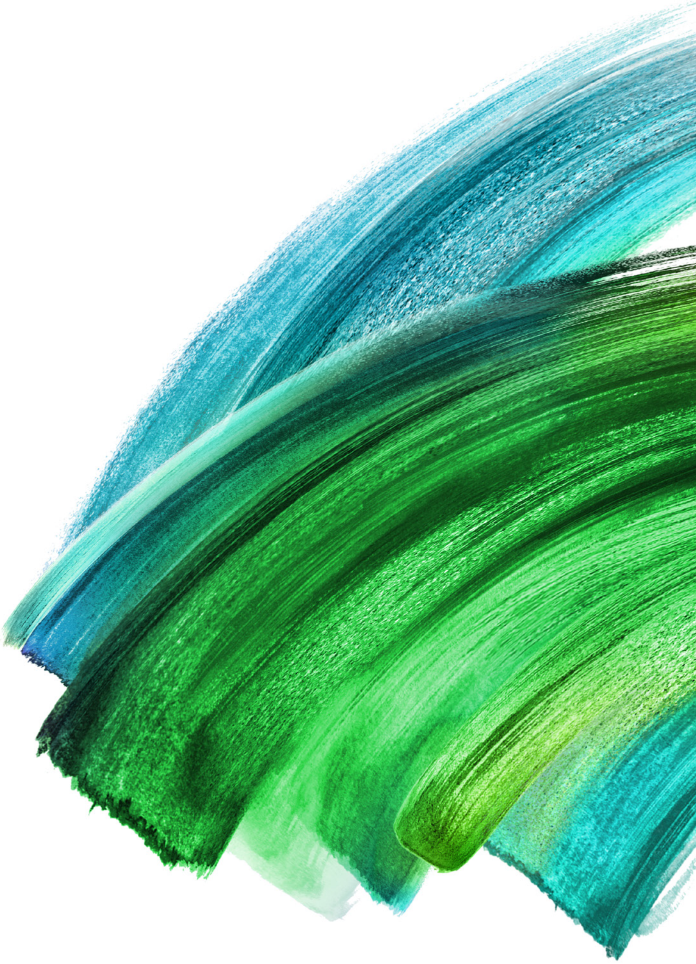
Deloitte.

IBM Modernization Reference Architecture

A technical perspective

2018 - Deloitte Consulting LLP

Abstract: Executive summary



A reference architecture serves as foundational guidance to support an organization’s business and IT objectives and is based on how an organization defines the key aspects of its IT landscape. A reference architecture is a fundamental component for any organization seeking to modernize their business solutions and IT platforms, and changes incrementally during each modernization phase as it transforms from its current state composition to its final, future state architecture.

Some organizations may be able to modernize their systems with a “big bang” approach driven by the nature of the system functions, or may need to overhaul their systems entirely due to a major organizational change or large business process overhaul. However, many organizations need to maintain important business applications and continuous operations while they modernize. Because their systems and solutions are often heavily interdependent, and they need to modernize with minimal disruption, many organizations often choose incremental modernization strategies rather than “big bang” approaches.

During incremental modernization projects, technology and application architects often focus heavily on the current and future state architecture definitions at inception and elaboration phases. Without an interim architecture in place to support a large, incremental transformation, delays or failures in modernization efforts can occur. Therefore, incremental architectures should be rigorously engineered, especially if they must sustain long-term projects. Even in an era in which development teams focus on agile, feature-centric development for the end-state, it is imperative that architects consider the fluid nature of incremental architectures, and build a solid foundation to reduce the implementation risks associated with performance engineering, security, and extensibility.

This paper focuses primarily on IBM offerings to help CTOs and architects navigate the inception and elaboration phase decisions of incremental modernization. It focuses on how a reference architecture shifts through iterations of modernization by addressing specific, commonly experienced use cases and business drivers, and how changes in reference architecture support the incremental modernization of business capabilities.

The focus is primarily on IBM offerings that support modernization, as many large organizations currently deploy and operate their solutions on IBM mainframe platforms using a common set of IBM software products and utilities. IBM products can unite and integrate mainframe and open system platforms, and IBM hardware and software can be configured and incrementally implemented to help organizations gain efficiencies within the value chain. Organizations looking to migrate from the mainframe to distributed, open system platforms should look to IBM. Their longstanding history of industry experience and market-leading mainframe, hardware, and software product offerings support complex IT modernization initiatives and projects.

Why modernize?

There are many challenges associated with using legacy solutions, including:

1

The high cost of maintaining and operating legacy systems, especially compared to the agility and lower pricing of more modern platforms.

2

Difficulty integrating legacy solutions with other tools and applications, as legacy systems do not avail themselves to changing modern technical opportunities such as those available through hosted infrastructure platforms or software products.

3

Lack of resource and talent available to program and maintain legacy systems. Developers well-versed in legacy programming languages are difficult to find and expensive to compensate. Additionally, the talent pool for legacy programmers continues to diminish as many reach retirement age, and organizations must pay a premium price for these resources, as the market lacks talent adept in programming languages such as COBOL. Developers who know modern languages, like Java, are much easier to find and compensate.

4

Difficulty adapting to a changing business environment and scaling to growing business needs. Organizations continue to rely on legacy systems for essential function and business decision-making, often because they do not have a thorough and accurate reference architecture that provides a clear view into their existing systems. Additionally, organizations do not typically have an existing comprehensive reference architecture framework, or may have difficulty forming a reference architecture that reflects an integrated legacy and modernized framework.

Because of the limited capabilities of legacy systems, organizations should consider a modernized reference architecture that reflects both legacy and a modernized framework. Organizations should consider an incremental approach to modernization versus a "big bang" approach, as there are significant risks to a big bang approach. This approach will allow an organization to shift from a legacy based referential architecture to a more modernized framework to better align with their IT vision and reduce cost associated with maintaining the legacy mainframe while minimizing risks.

Middleware products can support this type of modernization approach. For example, products such as IBM MQ, IMSConnect or an IBM Enterprise Service Bus (ESB) are architected to support the continued interaction of mainframe and distributed platforms. In addition, storage continues to decrease in cost, and the application of data across different tiers of storage have different infrastructures to support those tiers. An updated architecture can be leveraged to identify how the infrastructure can help to support data response needs.

Organizations could make investments to integrate other third party products, but this might be at the expense of leveraging an existing product and shifts to a more heterogeneous IT ecosystem. It is cumbersome for businesses to integrate third party tools or web applications with preexisting products without visibility into the current framework and insights about third party tools.

Modernization business drivers

Executives continue to struggle with several inhibitors to IT modernization. They seek to reduce costs and optimize business or IT resources, while considering approaches that take their current offerings and visions for the future into account. Modernization drivers include infrastructure platform hosting, resource availability, decreased agility to shift business portfolio offerings to meet market trends or changing or complex policy, regulatory or compliance demands, and resource availability and longevity.

Reduce Total Cost of Ownership (TCO):

- Reduce Capital Expenditure (CAPEX)/ Operating Expense (OPEX)
- Optimize skills and resources

Increase agility:

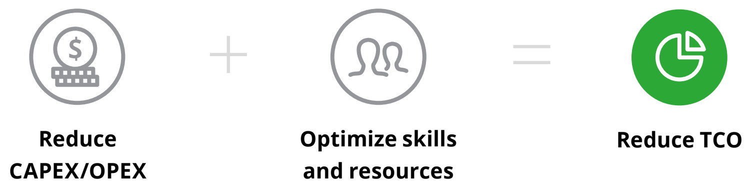
- Increase innovation
- Increase efficiency
- Increase regulatory compliance

Increase maintainability:

- Unlock data
- Reduce technical debt
- Increase interoperability

Reduce Total Cost of Ownership (TCO)

Organizations frequently seek to identify ways to reduce their total cost of ownership (TCO). Often, these can be achieved through modernization efforts that can help to reduce both direct and indirect costs of business and IT operations. These costs may comprise a large percentage of the IT budget. Many organizations, especially those in the commercial space, view IT TCO through two lenses: cost incurred and opportunity lost.



Reduce CAPEX and OPEX

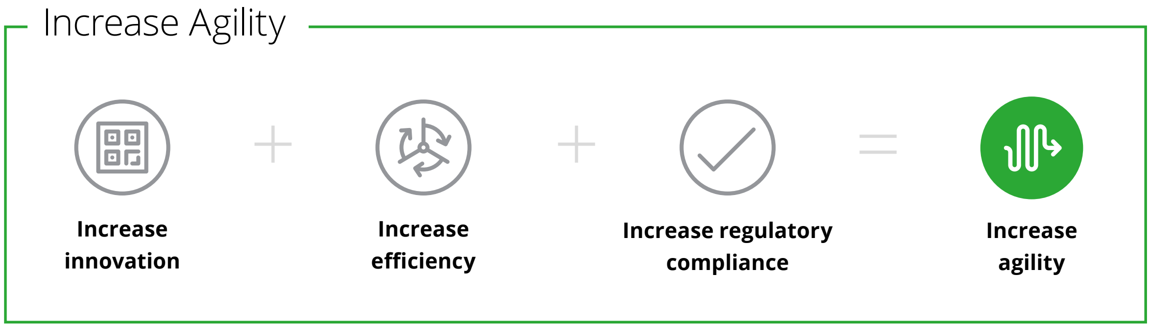
As systems evolve through business expansion or mergers and acquisitions, IT landscapes become very complex and are often made up of loose configurations of disparate hardware and software with hard to control CAPEX and OPEX. Due to this, organizations commonly look to migrate all or part of their solution to a hosted infrastructure platform or software as a service (SaaS) solution, which cost less and are consistent in hardware and software composition. Shifting to a distributed system platform in turn expands the pool of hardware and software products available to support different lines of business. Because these products often offer competitive licensing fees compared to mainframe software, organizations are faced with the challenge to modernize in a cost-effective and highly utilized manner, while continuing to support business functionality throughout the process. To support this challenge, a modernized or well-defined reference architecture can help to identify the components involved in the process, and help business leaders to easily visualize any additional IT capabilities that result from the modernization process.

Optimize skills and resources

Talent costs associated with supporting and maintaining a mainframe platform and legacy solutions can result in higher TCO. Therefore, it is crucial to optimize skills and resources within IT in order to maintain a lean, efficient organization that can support constantly evolving technology landscapes. Leaders should consider several factors when determining how software developers and IT resources factor into TCO.

Legacy software developers and IT resources who maintain and operate the mainframe are often more expensive to compensate than their modern developer counterparts. Additionally, the talent pool for programmers familiar with legacy coding continues to diminish as many retire from the workforce. In comparison, resources who are familiar with open source products and third-party tools have saturated the market.

The reality is that many organizations require both legacy and modern-day developers and support resources to manage legacy systems while adopting modern systems. Therefore, organizations must consider modernization from a business, IT leaders, and talent perspective in order to most effectively optimize the skills and resources available for modernization work.



Increase Agility

Increase innovation + Increase efficiency + Increase regulatory compliance = Increase agility

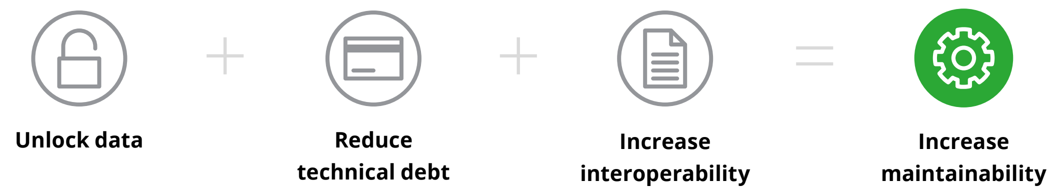
Increase innovation, increase regulatory compliance, increase regulatory compliance

Many organizations across industries require greater agility to meet policy, compliance, or regulatory requirements. Legacy solutions—which are based on larger monolithic code structures and tightly coupled programs—typically do not lend themselves to the fluidity of industry regulations. Despite this reality, organizations expect technology to provide real-time information to drive decision-making as quickly and efficiently as possible.

The legacy mainframe environment requires years of custom coding and has limited integration capabilities, making it difficult to increase straight-through processing that addresses changes in regulation. Incrementally modernizing from the legacy mainframe enables business to have a responsive and a well-integrated system, as ever changing modern architectures and technologies can provide the platform and tools necessary to respond to rapidly to changing requirements.

Because legacy systems lack the capacity or computing power to meet constantly changing business, they become hard to adapt to changing environments and difficult to scale for growing business needs. Implementing modern technologies can enable organizations to become more nimble, help them adapt to new regulatory and compliance standards, and promote efficient adaptation of new business and IT processes such as agile development methodologies and DevOps principles.

Increase Maintainability



Unlock data + **Reduce technical debt** + **Increase interoperability** = **Increase maintainability**

Unlock data

Today, most businesses believe that unlocking data is crucial to propelling growth. Because legacy architectures make it difficult to share data within the organization, the cost of maintaining, exchanging, and reporting on data is significant. Unlocking data through modernization can lead to better collaboration across the business, easier data integration in large merger and acquisition scenarios, better reporting and analysis, and ease of maintenance across the overall system.

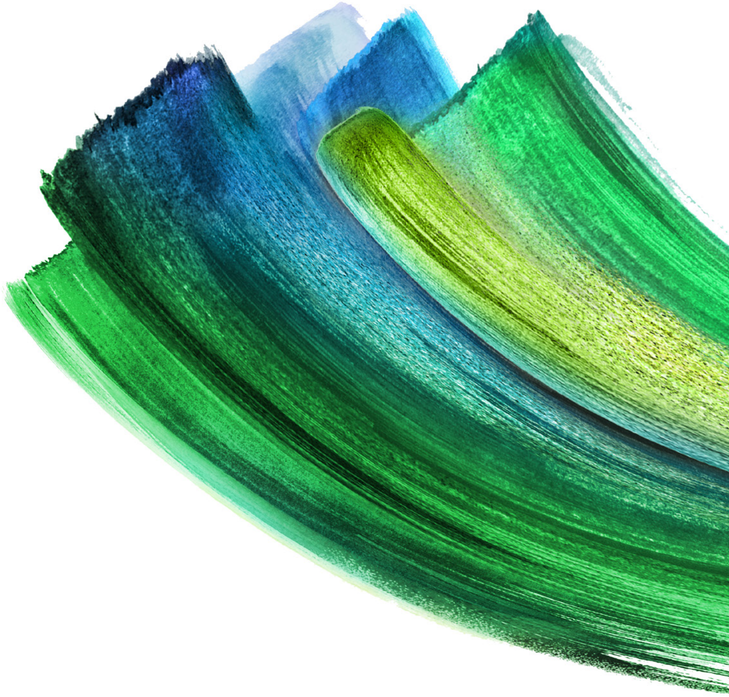
Reduce technical debt

Any information system accrues ongoing technical debt that can grow exponentially if the code is not updated with the latest standards. Therefore, businesses should have the ability to easily detect changes for anti-patterns using automated code quality tools and continuous integration processes to reduce debt.

In addition, it is beneficial to have a programming platform that has readily available proper coding patterns with cost effective resources to fix the findings from the code quality tools. This also allows platform technologies and interoperability to remain maintainable and increases the life of the application on supported technology platforms.

Increase interoperability and maintainability

Trends in referring to "interoperability" have changed over time: first, architects utilized "distributed computing interoperability," then "SOA" to commonly used terms such as "APIs" and "economies of APIs". However, too many organizations focus on discussing the concept externally, rather than choosing technologies to promote interoperability internally. During modernization projects, executives should pick technologies that not only increase the interoperability with new open systems platforms, but also interact with the mainframe in order to support the interim architecture. In other words, it's best to choose technologies with patterns that can scale for performance and security—and that are easily maintainable in the long term.



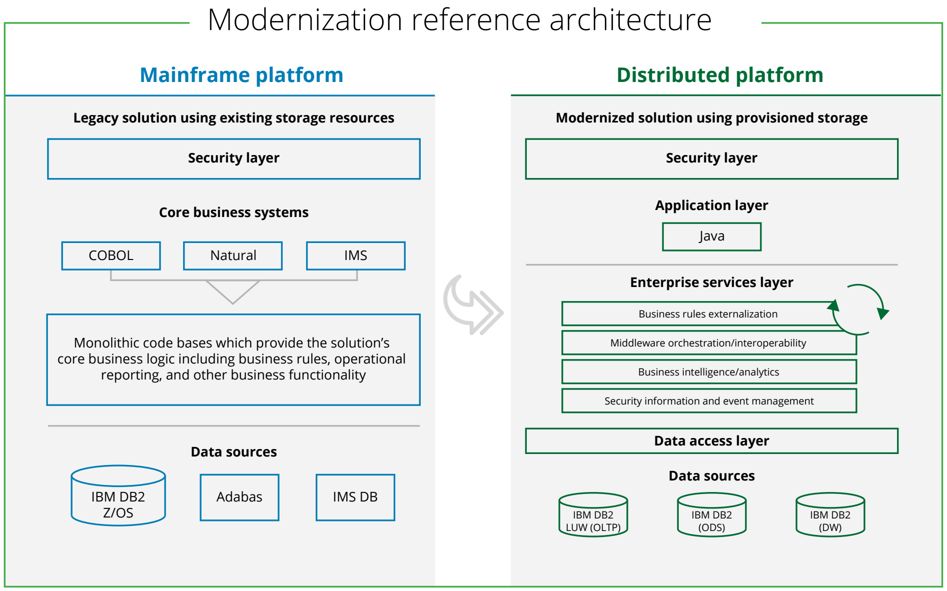
Modernization reference architecture

The following sections focus on key reference architectures to modernize solutions, and how these changes address common business drivers.

Modernization automated refactoring through a change in reference architecture can help to support an organization's move to a modernized architecture, as leveraging refactoring works for both legacy and distributed, open system technologies. Refactoring can be an ideal solution for many organizations.

Organizations should consider two questions when working through modernization reference architectures:

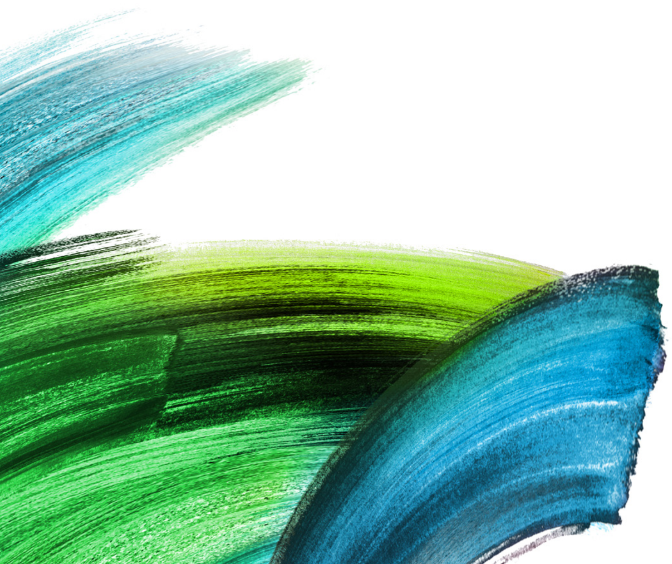
- What are the core changes to platform and tools we should focus on for the initial shift in modernization?
- How will the platform and tools integrate with the legacy solution to continue business operations during the transitional modernization period?


Modernization reference architecture
Mainframe platform
Legacy solution using existing storage resources
Security layer
Core business systems
COBOL
Natural
IMS
Monolithic code bases which provide the solution's core business logic including business rules, operational reporting, and other business functionality
Data sources
IBM DB2 Z/OS
Adabas
IMS DB
Distributed platform
Modernized solution using provisioned storage
Security layer
Application layer
Java
Enterprise services layer
Business rules externalization
Middleware orchestration/interoperability
Business intelligence/analytics
Security information and event management
Data access layer
Data sources
IBM DB2 LUW (OLTP)
IBM DB2 (ODS)
IBM DB2 (DW)

This illustration depicts a refactoring opportunity where a subset of the overall solution is migrating off the mainframe and being implemented on a distributed, open system platform while retaining its continued interaction to support the current business process.

In this reference architecture, there is a shift from mainframe-based solution and platform to a solution comprised of multiple hardware platforms and software products.

The objective is to deploy a solution on a distributed platform, but maintain the integrity of the system and interoperability of the solutions during the modernization effort. The two hosted solutions take into account three common areas of focus: **platform, security,** and **middleware.** In modernization, these areas will be implemented and used incrementally over time from the initial modernization phase through the transition period of modernization.

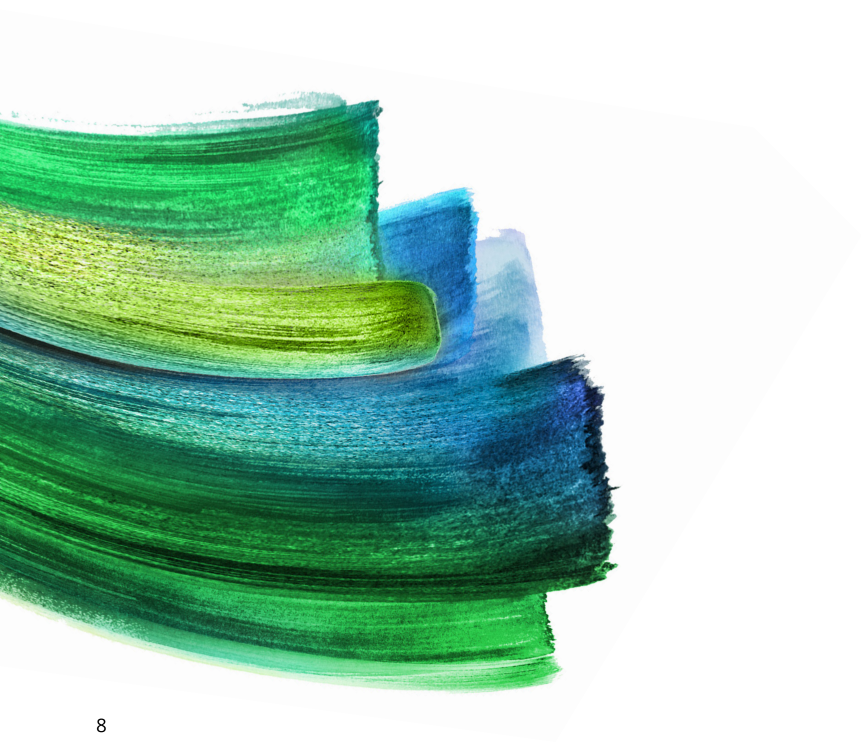


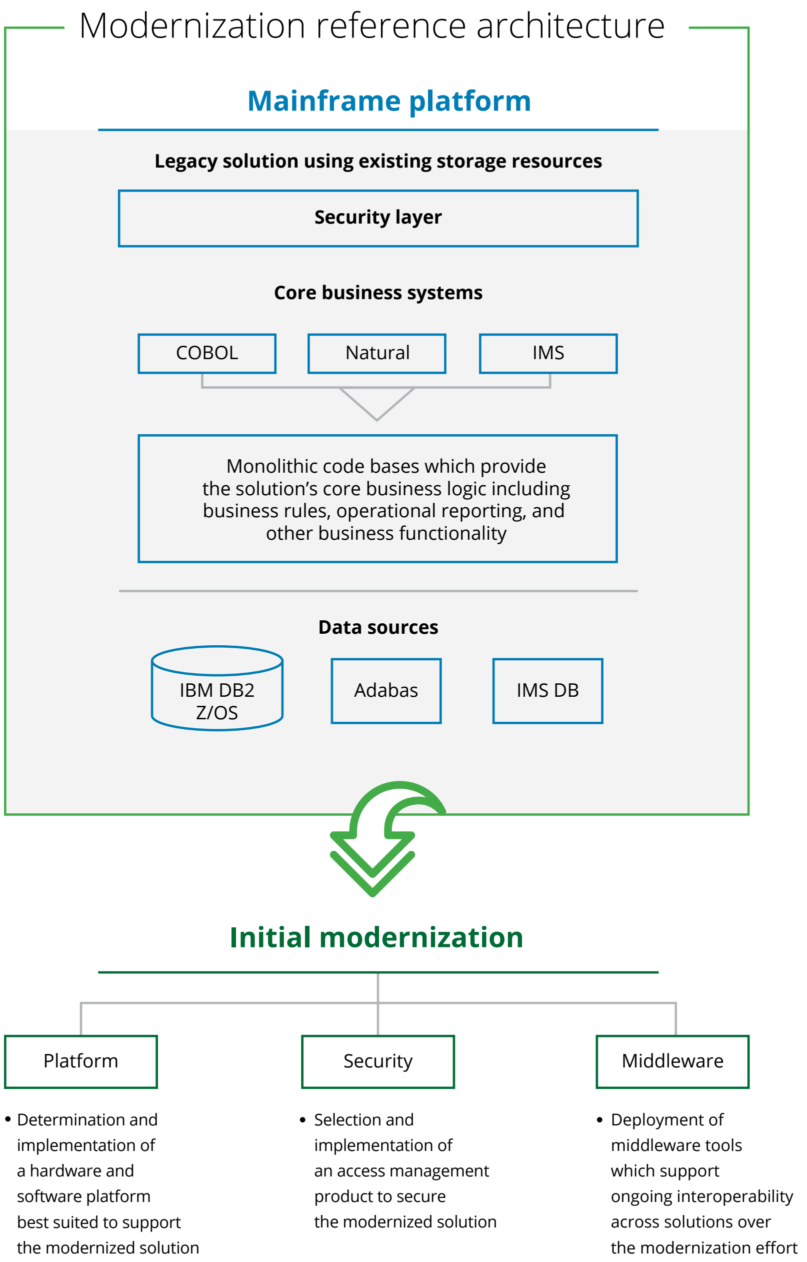
Initial modernization phase

The initial phase of the modernization effort is critical, and requires a fundamental shift in several architectural components of the solution. Most organizations choose to focus on:

Platform **Security** **Middleware**

The modernization change requires an organization to rethink how its hardware platform, software, and middleware are implemented. This initial modernization step is the starting point from which the reference architecture will iteratively change until the organization's strategic and operational goals are met. These fundamental changes are illustrated in the graphic ("Modernization Reference Architecture"), which identifies the initial modernization changes in the architecture.





Modernization reference architecture

Mainframe platform

Legacy solution using existing storage resources

Security layer

Core business systems

COBOL | Natural | IMS

Monolithic code bases which provide the solution's core business logic including business rules, operational reporting, and other business functionality

Data sources

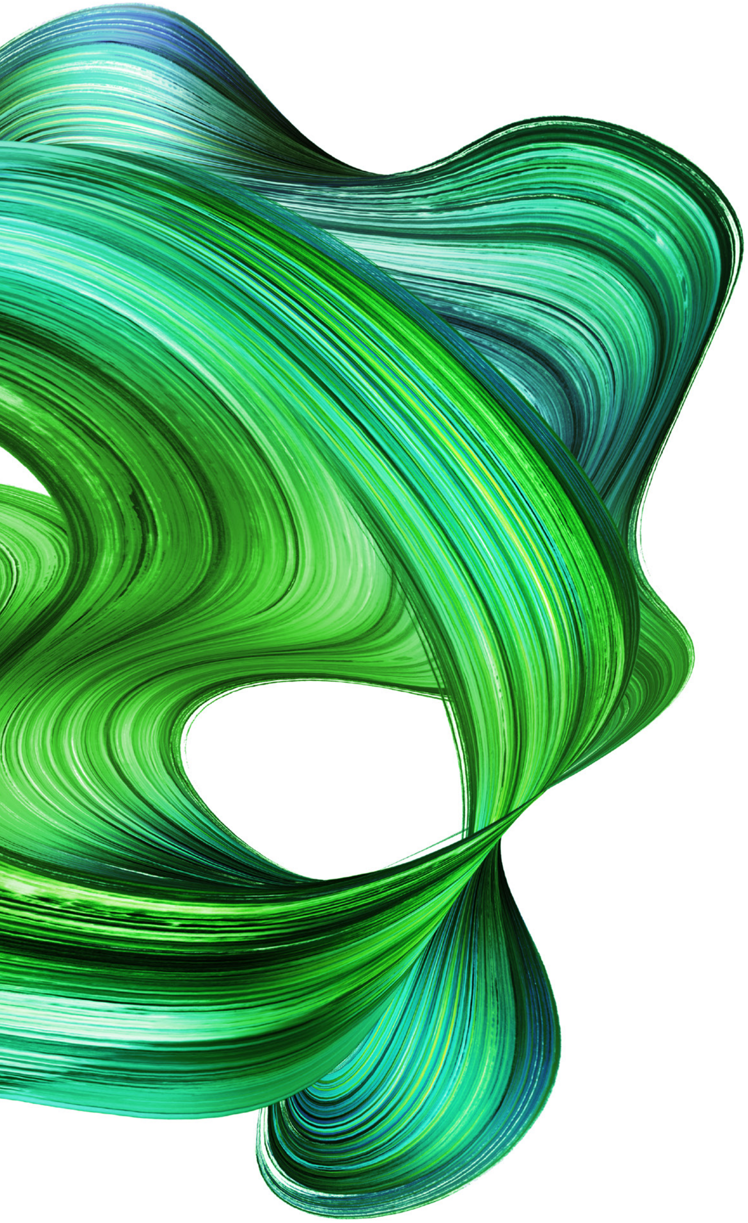
IBM DB2 Z/OS | Adabas | IMS DB

Initial modernization

Platform
- Determination and implementation of a hardware and software platform best suited to support the modernized solution

Security
- Selection and implementation of an access management product to secure the modernized solution

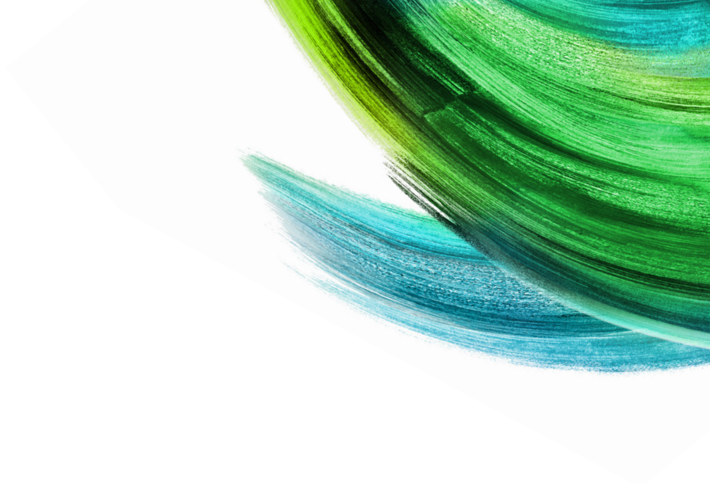
Middleware
- Deployment of middleware tools which support ongoing interoperability across solutions over the modernization effort



Platform considerations

The following sections provides perspective on how IBM technologies can provide solutions which address business drivers and assist the solution to effectively support the business. This change in platform may be the first time an organization has departed from the mainframe into a cloud-hosted infrastructure implementation. In that case, an organization can determine what platform may help them reduce their TCO, provide agility, and offer a high degree of maintainability.

Hosting the solution on a new platform is a fundamental change in the IT infrastructure and should be clearly defined in the reference architecture, as the new platform may require additional changes to key supporting components of the core solution. Another key consideration is the use of storage across platforms. Mainframes and distributed servers may operate independently, but with a high performance storage appliance and common management suite of tools, businesses can optimize their use and reduce the TCO. Additionally, maintaining disparate platforms through a uniform product set eases the burden on IT staff to provision and maintain the infrastructure.



Recommended IBM technology

Product

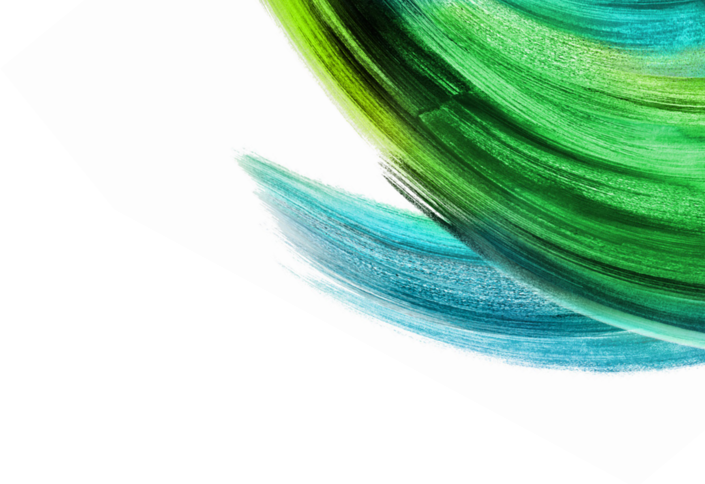
Server Options Z and LinuxOne

Usage rationale

IBM Z systems and LinuxOne provide significant advantages in serving up large compute nodes. This is crucial for complex processing systems that have been modernized using code conversion tools that may not be fully compute resource optimized for a target platform. They provide significant advantages in running large parallel threads without the hassle of node distribution, which is crucial to simplify topologies during both the incremental and target modernization architecture phase.

Modernization drivers

- **Reduce OPEX/ CAPEX**
- **Optimize skills and resources**
- **Increase innovation**
- **Increase efficiency**
- **Increase interoperability**
- **Increase regulatory compliance**
- **Unlock data**
- **Reduce technical debt**



Recommended IBM technology

Product

IBM All Flash Array (AFA) Storage Appliances

Usage rationale

IBM All Flash Arrays provide significant Input/Output Operations Per Second (IOPS) gain for high performance workloads, which can be crucial for data exchanges between mainframe and target architecture during the interim architecture phase. With their low form factor, low latency, and high throughput (including storage optimization), they can significantly reduce OPEX/CAPEX for organizations.

Modernization drivers

Reduce OPEX/ CAPEX

Increase innovation

Increase efficiency

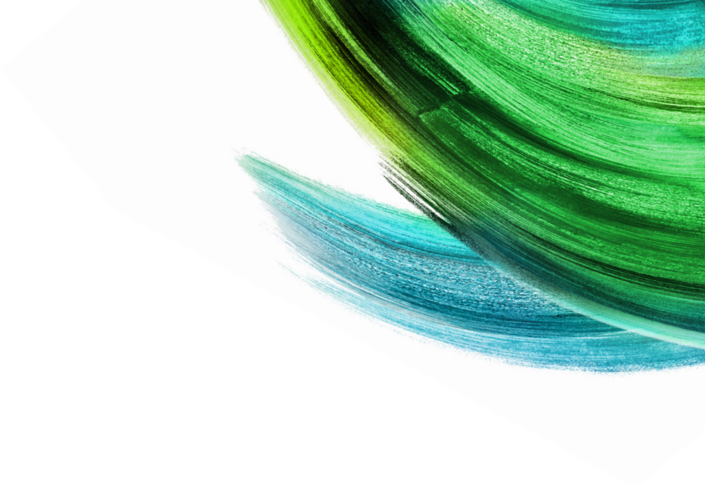
Increase regulatory compliance

Unlock data

Reduce technical debt

Optimize skills and resources

Increase interoperability



Recommended IBM technology

Product

IBM Spectrum Software Defined Storage, Compute and Network

Usage rationale

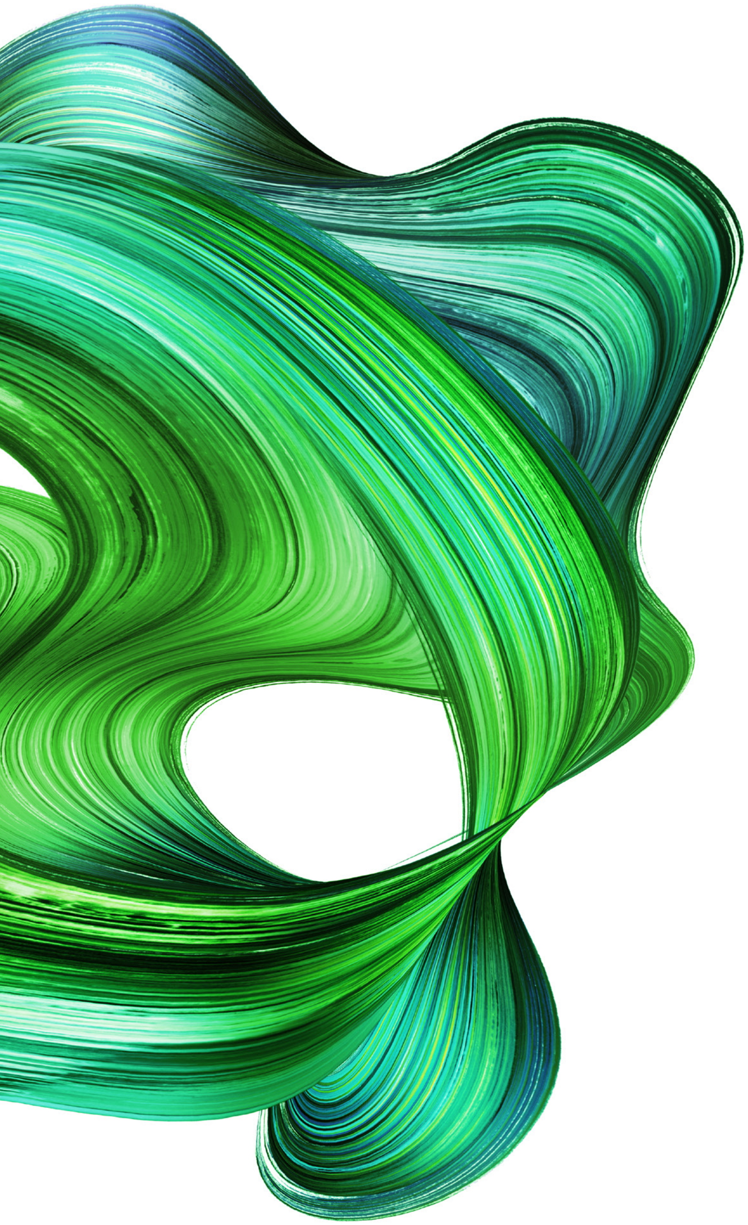
IBM Spectrum Software Defined Storage, Compute and Network provide significant advantages in single pane management of systems across disparate platforms. They offer advanced capabilities around ease of provisioning, protection, and template based deployments leading to reduced OPEX/CAPEX as well as increasing efficiency in infrastructure maintenance.

Most organizations consider one or more public cloud platforms to integrate within some hybrid cloud architecture pattern in which they can use their on-premise investments in hardware as well as burst to cloud for easier commodity use cases. This presents unique challenges, especially considering data management in multi-cloud environments can pose an array of problems around management, security, visibility, portability, and replication. As DevOps continues to rise in popularity, software defined storage in multi-cloud deployments becomes even more critical as a technology as opposed to a nice to have feature. IBM Software Defined Storage solution with the Spectrum suite of products offer a great way to reuse existing Storage arrays, enhance the flash usage, and provide unified capabilities around protection, replication, and portability in these multi-cloud environments.

Modernization drivers

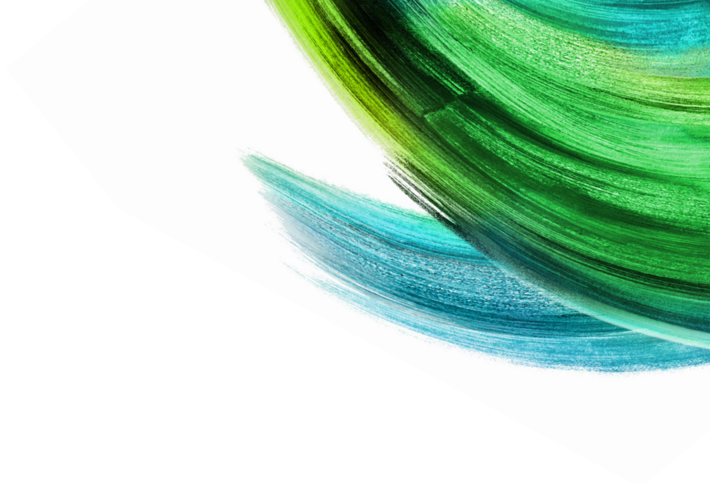
- **Reduce OPEX/ CAPEX**
- **Optimize skills and resources**
- **Increase innovation**
- **Increase efficiency**
- **Increase interoperability**

- **Increase regulatory compliance**
- **Unlock data**
- **Reduce technical debt**



Security considerations

The security products used to support access management will be different between the mainframe and distributed platforms. Modernization to an open system platform will require an access management product which can support the access requirements of the solution. IBM WebSEAL provides similar access management functionality for web based solutions as IBM’s Resource Access Control Facility (RACF) for mainframe solutions.



Recommended IBM technology

Product

IBM WebSeal

Usage rationale

IBM WebSEAL is an easily integrated access management solution that provides granular access control for web-based applications. This solution can scale to meet an organization’s authentication volumes while also offering the ability to support single sign-on (SSO) and multifactor authentication across an enterprise.

Modernization drivers

Optimize skills and resources

Increase efficiency

Increase regulatory compliance

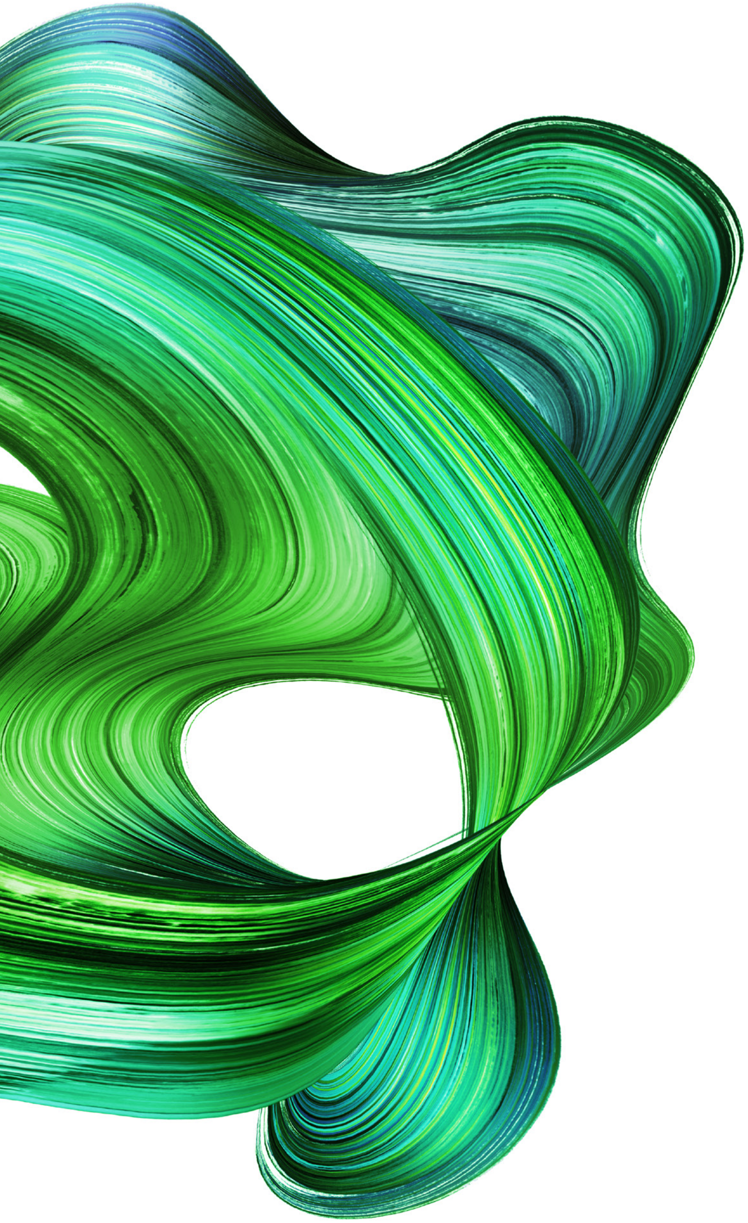
Increase interoperability

Reduce OPEX/ CAPEX

Increase innovation

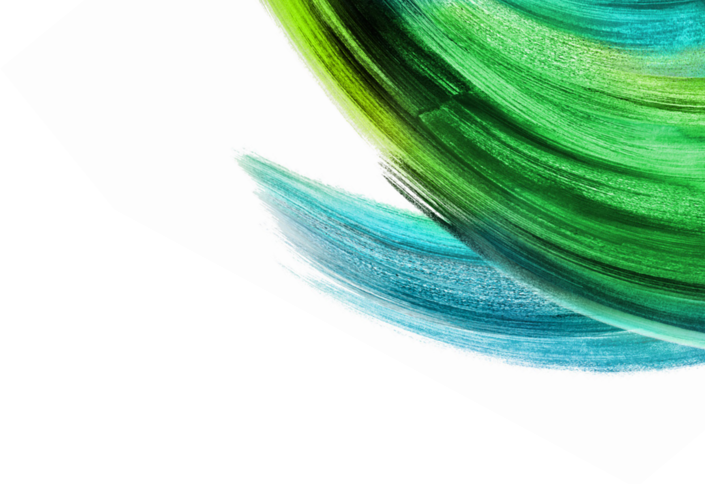
Unlock data

Reduce technical debt



Middleware considerations

Unless a modern solution is used completely separately from a legacy solution, the two will have points of interaction. When modernizing a solution, the interactions need to persist so that ongoing business activities continue without interruption. A reference architecture can assist in defining how solutions communicate within their own boundaries or between platforms. A Java or .NET based solution still needs to communicate with legacy COBOL, Natural, or IMS programs and exchange data. These communications can be orchestrated and controlled through middleware products. There are several IBM products which can support this.



Recommended IBM technology

Product

IBM IMS Connect

Usage rationale

IMS Connect offers several options which allow open systems to integrate with the legacy IMS solution, which is critical while modernization is in progress and tightly coupled solutions need to continue functioning as before. IMS Connect provides this integration though its native functionality and other available adapters. For example, Open Database Manager allows Java solutions to communicate with the legacy solution IMS DB.

Modernization drivers

Reduce OPEX/ CAPEX

Optimize skills and resources

Increase efficiency

Unlock data

Increase interoperability

Increase innovation

Increase regulatory compliance

Reduce technical debt

Recommended IBM technology

Product

IBM MQ

Usage rationale

IBM MQ is a core middleware tool used by companies with legacy solutions to support their messaging needs. For many organizations, the continued use of this tool across their mainframe and open system platforms can increase the ease of integration of solutions across different platforms while adding a level of security to protect their data.

Modernization drivers

Reduce OPEX/ CAPEX

Optimize skills and resources

Increase efficiency

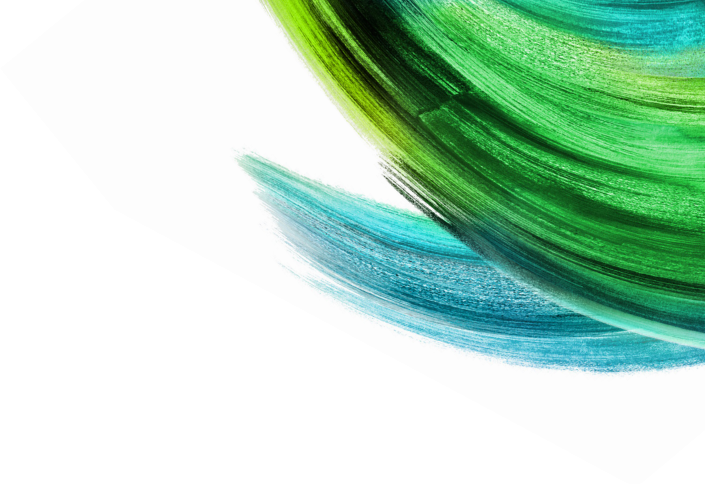
Unlock data

Increase interoperability

Increase innovation

Increase regulatory compliance

Reduce technical debt



Recommended IBM technology

Product

IBM WebSphere Application Server

Usage rationale

This product has extensive capabilities that support the scale and functionality of applications. This allows the product to scale up with hardware and storage implementation, as well as easing integration with other middleware and security information and event management Security Information and Event Management (SIEM) products.

Modernization drivers

- **Optimize skills and resources**
- **Increase efficiency**
- **Increase interoperability**

- **Reduce OPEX/ CAPEX**
- **Increase innovation**
- **Increase regulatory compliance**
- **Unlock data**
- **Reduce technical debt**

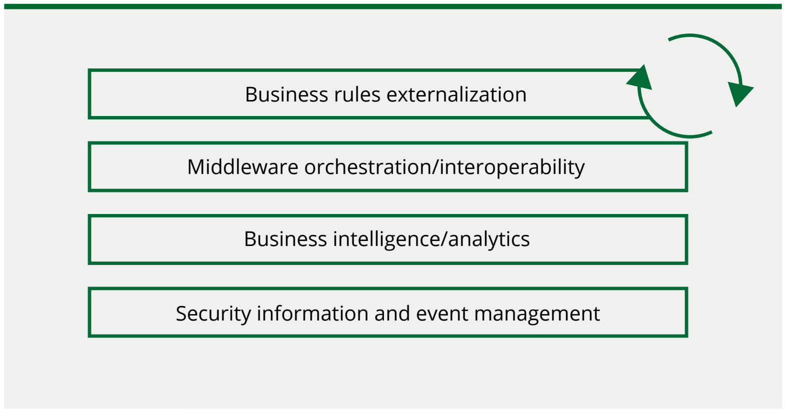
Transitional phase of modernization

Beyond the initial modernization phase, many organizations focus their efforts on expanding or improving business solutions through the implementation of customer-centric products due to newfound and adept solution capabilities and more efficient IT resources. These modernization components generally address multiple drivers including reduced TCO, business agility, and maintainability. Modernization efforts occur over a series of iterations that build on each other, and tend to focus on strategic goals that add value.

Modernization iterations will vary based on organization’s needs. At the end of this effort, the core solution footprint is reduced as components move out of it into modular components. As noted at the onset of the document, many organizations focus on business changes which are supported by technology to deliver the functionality they require. The graphic at right reflects several common focus areas for organizations looking to reduce their TCO, increase agility, or improve maintainability.

Because each organization will have its own challenges, business drivers, and strategic objectives, there is no singular modernization path. The following use cases provide insight into different organizations who chose to modernize using different strategies.

Incremental modernization

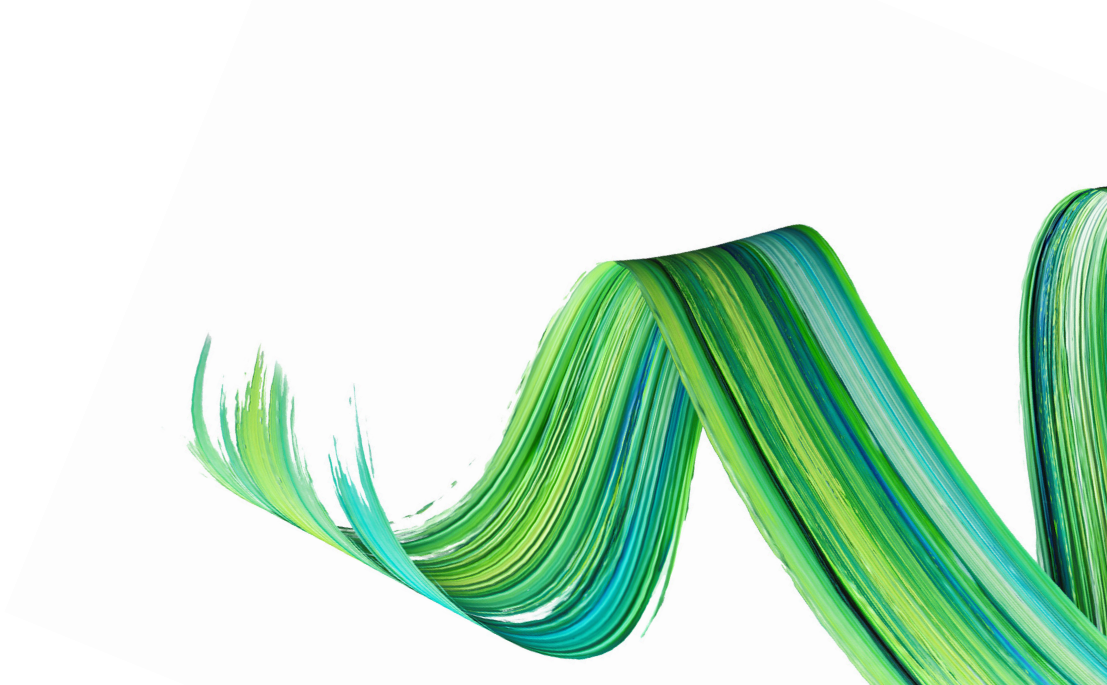


Business rules externalization

Middleware orchestration/interoperability

Business intelligence/analytics

Security information and event management



Recommended IBM technology

Product

IBM Operational Decision Manager

Usage rationale

Business rules externalization
Flexibility and agility are common objectives across industries and are a consistent focus for executives due to the need to remain compliant with business policy, regulatory, or statutory requirements. IBM Operational Decision Manager can help to externalize business rules and separate them from the application code, thereby increasing agility of the business logic. It also allows improved code management with a centralized code set and distributed server resources for Business Rules and Application services.

Modernization drivers

Increase regulatory compliance

Increase efficiency

Reduce technical debt

Increase interoperability

Reduce OPEX/ CAPEX

Optimize skills and resources

Increase innovation

Unlock data

Recommended IBM technology

Product

IBM Integration Bus

Usage rationale

Middleware orchestration/interoperability
For many organizations, the act of modernization focuses on shifting from monolithic, coupled solutions to more modern solutions. Incremental modernization focuses on breaking the solutions down into a service-based model which can orchestrate messaging and APIs through a shared orchestration tool. This lowers the TCO as the use expands across the enterprise, increases agility as business changes can be introduced more quickly, and improves maintainability through reduced technical debt.

Modernization drivers

- **Optimize skills and resources**
- **Increase innovation**
- **Increase efficiency**
- **Increase interoperability**

- **Reduce OPEX/ CAPEX**
- **Increase regulatory compliance**
- **Unlock data**
- **Reduce technical debt**

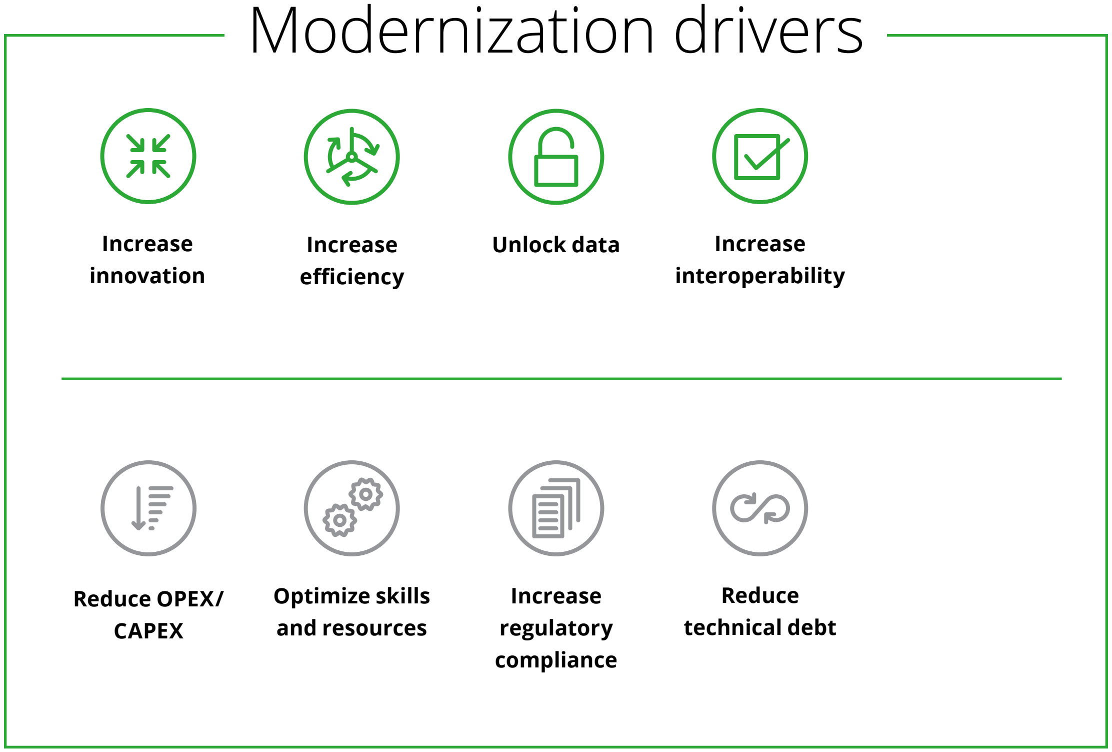
Recommended IBM technology

Product

IBM Cognos BI

Usage rationale

Business intelligence/analytics
IBM Cognos provides businesses with a software product that can unlock data to provide executives with critical business insights, and staff with additional capabilities to perform their work. Cognos introduces dynamic and ad hoc reporting ability to increase visibility of key business performance indicators across the organization and allows for informed decision-making.



Modernization drivers

Increase innovation

Increase efficiency

Unlock data

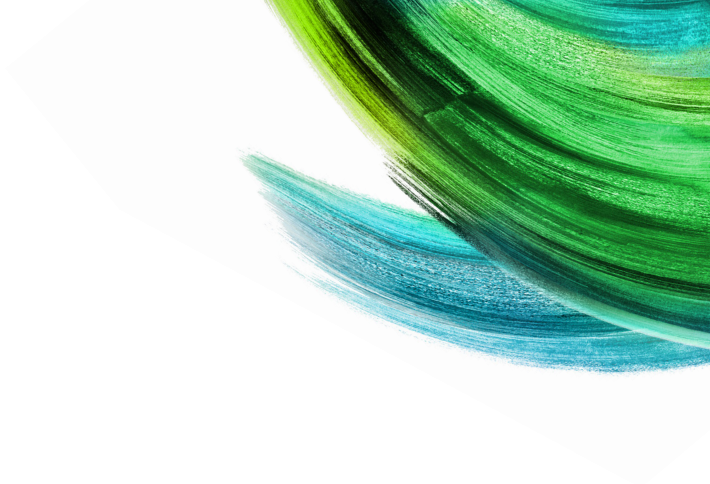
Increase interoperability

Reduce OPEX/ CAPEX

Optimize skills and resources

Increase regulatory compliance

Reduce technical debt



Recommended IBM technology

Product

IBM QRadar

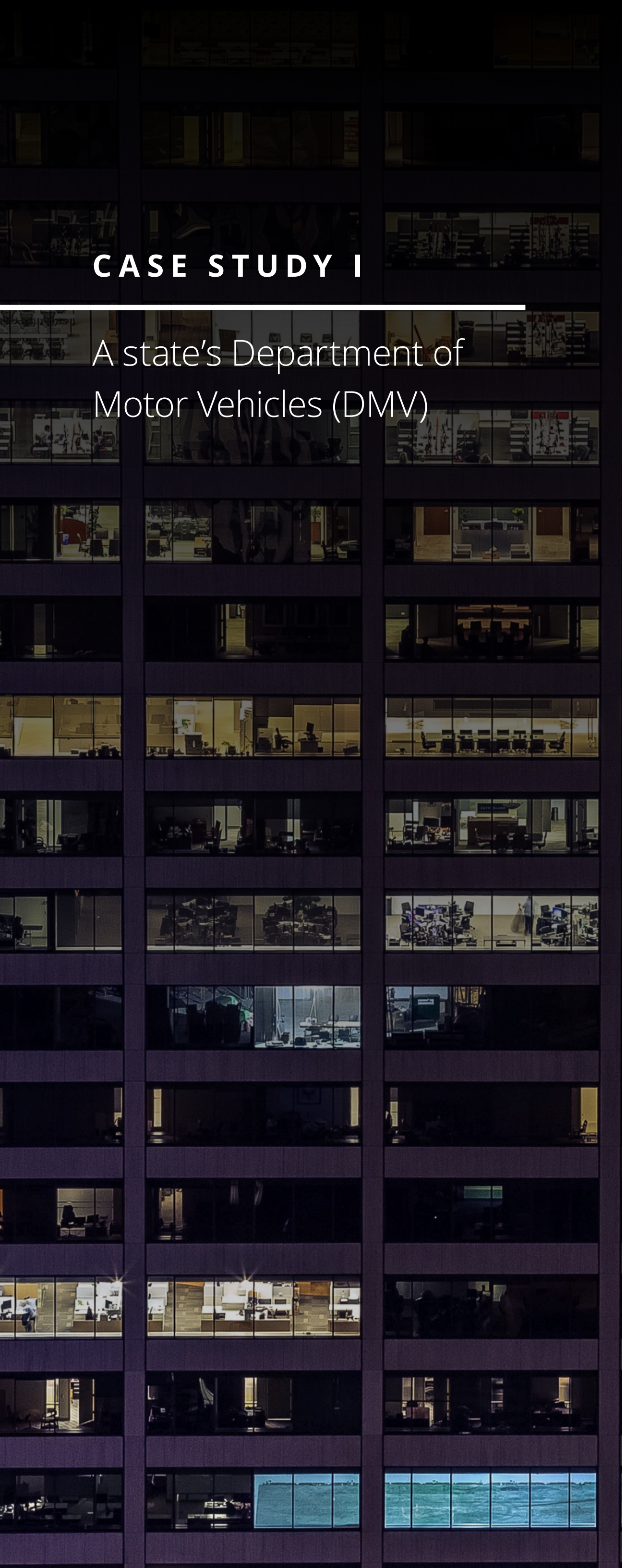
Usage rationale

Security information and event management (SIEM)
IBM QRadar is a Security Information and Event Management (SIEM) product which provides insights for security-related information. The implementation of QRadar makes sense for many organizations due to its ability to gather and report security-related information produced from RACF on the mainframe and WebSEAL on the open system platform. The use of QRadar across both platforms provides broader visibility through a more efficient, integrated mechanism and provides operational efficiency in reviewing events occurring across platforms and middleware products through a single product.

Modernization drivers

- **Reduce OPEX/ CAPEX**
- **Increase efficiency**
- **Reduce technical debt**

- **Optimize skills and resources**
- **Increase innovation**
- **Increase regulatory compliance**
- **Unlock data**
- **Increase interoperability**



CASE STUDY I

A state's Department of Motor Vehicles (DMV)

The challenge

The state's Department of Motor Vehicles (DMV) had hundreds of employees and millions of registered vehicles to account for, and leveraged a high-volume information system to do so. However, many of the systems that it utilized were obsolete and not integrated with each other—including a 20-year-old system coded in ADABAS/NATURAL and COBOL with a Java Swing point-of-sale application that managed registration and titles. The various platforms involved were expensive to maintain, inflexible, and did not support the agility and analytics operations needed to meet business objectives, such as addressing fraudulent transactions and financial reporting and effectively addressing losses from theft. As a result, the DMV's legacy platform system needed to be incrementally modernized to meet its business demands.

The application modernization approach

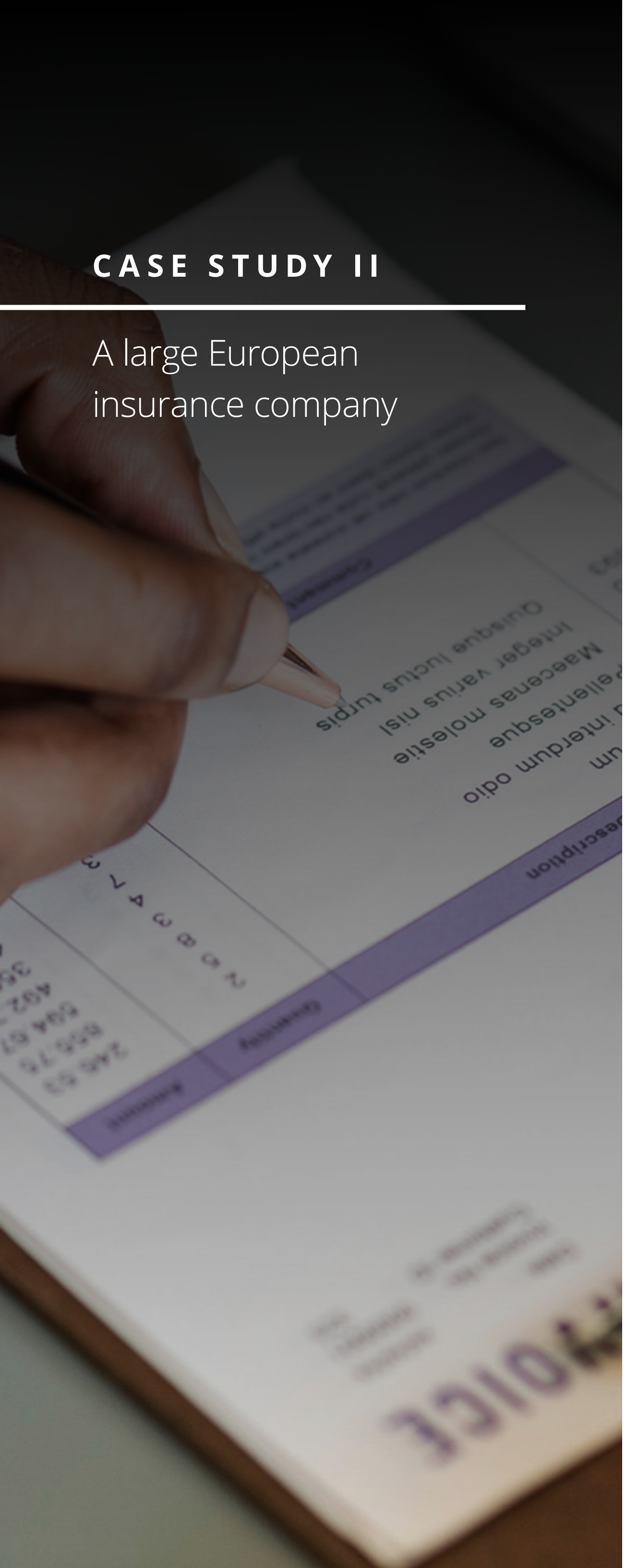
Deloitte worked with the DMV to map out an incremental approach that would enable legacy system modernization. Deloitte determined that the upgraded system needed to handle service enhancement, legislative changes, ad hoc reports, and other key priorities. In addition, Deloitte determined the time, effort, and resources needed for the modernization, and the respective roles and responsibilities of the team. The release schedule was developed based on these factors.

The process

Pete Evans, specialist master, Deloitte Consulting LLP, explains how Deloitte approached the project: "The project refactored millions of lines of code to a new Java-based platform. It updated more than 700 screens, 3,000 batch jobs, and 120 ADABAS Database System files. Core mainframe data structures were refactored from ADABAS to a relational database, which boosted enterprise reporting capabilities. Other technologies, such as PL/1, Assembler, and COBOL, were also migrated to the new Java-based platform and established a common enterprise architecture."

The results

Incremental modernization from the legacy system allowed the DMV to improve their capabilities with point-of-sale fraudulent transactions and financial reporting. "Deloitte used the IBM Cognos reporting platform to condense more than 1,000 legacy reports primarily written in COBOL into 27 customizable, parameter based reports, easing the overall maintenance load and directly providing the business flexibility to the end user," Evans explains. This has allowed the DMV to effectively address losses from theft.



CASE STUDY II

A large European insurance company

The challenge

A large European insurance company with operations in more than 100 countries embarked on an enterprise-wide transformation to help it achieve strategic growth.

At the onset, the insurer faced a problem: its claims-processing application could not be migrated to a flexible, open-systems platform. The software ran COBOL, an outdated programming language, which can be difficult to integrate with modern programming languages, database software, and Java environments. The insurance company faced difficulty finding IT personnel with experience in legacy code languages like COBOL, making the system increasingly costly to support. The insurer also needed to update its claims-processing application without altering the user interface and functionalities to minimize disruption for end users and help ensure that developers and database admins did not face a steep learning curve. Finally, the company needed to retool the applications without system downtime or interruption to operations.

The application modernization approach

The insurance firm engaged Deloitte's application modernization services to help it plan and implement an end-to-end modernization program of its legacy claims-processing application. The project began with a discovery assessment that analysed the COBOL code, mapped interfaces to other applications, and identified mutual dependencies. The Deloitte team worked with the company's IT staff to identify numerous applications that would be affected by the refactoring. The discovery work resulted in a defined scope, duration, and project implementation plan for the modernization project. Separately, the Deloitte team carefully reviewed the roles and responsibilities of the firm's business departments and suggested revisions that could improve process efficiency.

Based on this assessment, Deloitte's application modernization team recommended a migration strategy based on automated refactoring, a technique that uses software to automatically replace old code like COBOL and Natural with newer languages such as Java and .NET. The modernization project would employ Deloitte's transformation tools, a component of our application modernization powered by Deloitte's innoWake™ suite of solutions. The transformation tools can convert all legacy code into modern languages with full accuracy, a process known as 100% automated refactoring. And because the new code mirrors legacy programming, both legacy and Java developers can easily read and maintain the migrated application. This approach preserves the functionality and interfaces of the existing application while building a foundation for future modernization.

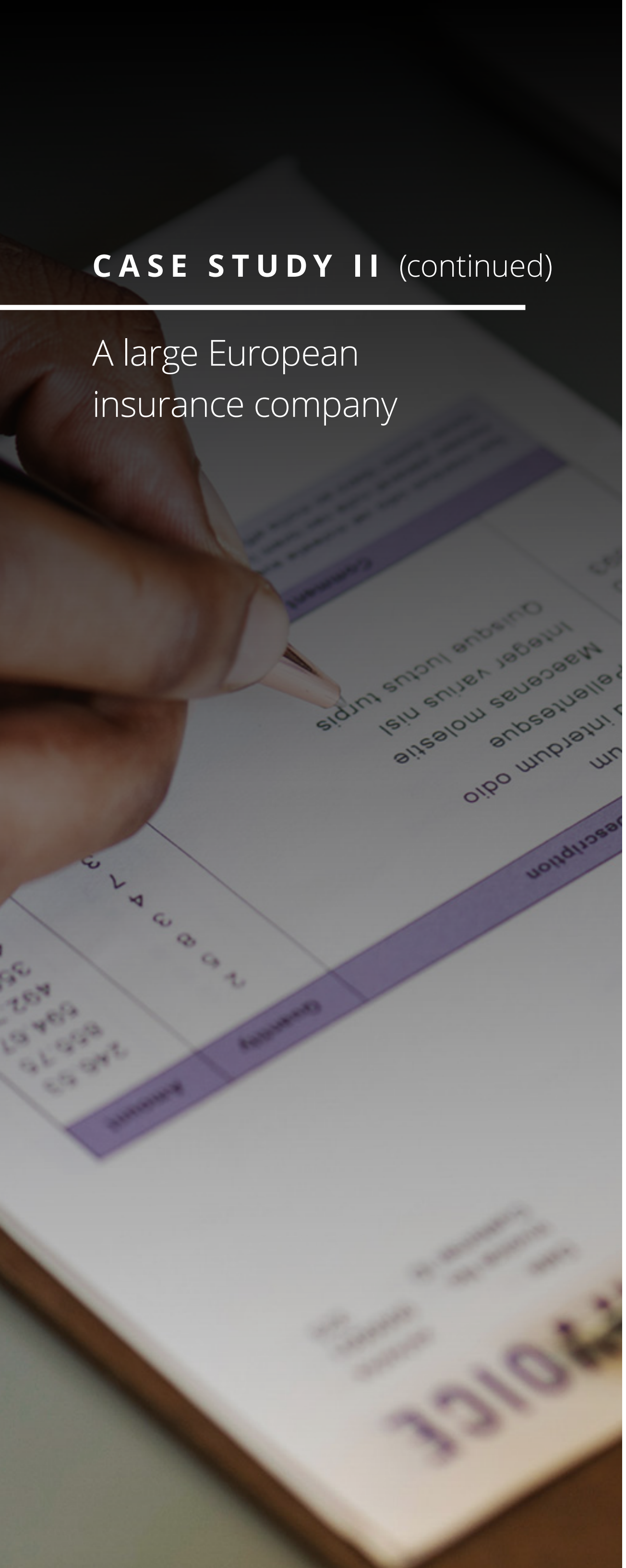
The process

The transformation project was comprised of three major initiatives: migrating the COBOL code to Java, transferring the data from Adabas to Oracle, and updating the mainframe operating system (OS) to Linux from BS2000.

The Deloitte application modernization team used its proprietary innoWake tools to refactor the legacy code. The tools simultaneously migrated the legacy and refactored code to the mainframe and to the new target architecture. This allowed normal maintenance of code and data to proceed without a code freeze or system downtime, minimizing disruption to the business.

The next step was to fully automate the code migration and the Job Control Language (JCL) implementation. This approach helped create a smooth transition for legacy developers, who could continue to work with COBOL using the Eclipse Integrated Development Environment (IDE).

To verify system functionality, the application modernization team created more than 200,000 record test cases that were used for regression and performance analysis of the new engine. This very high number of test cases not only helped speed implementation, but also delivered insights into the application and helped build confidence that the software would perform well in production.



CASE STUDY II (continued)

A large European insurance company

The results

To help the insurer prepare to use, maintain, and operate the new version of its claims-processing system, the Deloitte team designed a training program that began early in the project.

Rapid implementation, minimal training

Within six months, the team conducted Java toolchain workshops for legacy developers, and by go-live, all developers had been fully coached in the use of the new code and tools. Because the interface and functionality of the application remained the same, claims processors quickly accepted the updated software, with minimal training required.

Directly after go-live, the company's IT staff assumed full management and operations of the application. Almost immediately, the insurer began developing and integrating new functionalities using Java code.

Identical functionality and interface

When the 28-month modernization project concluded, the insurance firm had a highly integrated version of its claims-processing application that provides the same interface and functionality as the legacy version.

The modernization project was completed on time and on budget, with no code freeze or downtime, a critical requirement for the insurer. "Everything works just as it did before refactoring, but the Java environment makes it easier to integrate operational standards and align with best practices," said Mathias Jacobi, project manager, Deloitte Consulting LLP. "End users did not notice any changes in the application. We promised nothing would change in interfaces and functionality, and from a user standpoint, nothing did change—even though everything has changed under the hood."

A solid foundation for the future

The refactored system is now hosted on an open-system platform using modern technologies and supporting tools. All core functions of the application back end are now available as RESTful Web services, making it easier to share business functions of the server. As a result, new functionality is easier, faster, and less complex to develop and integrate.

Finally, the modernization will help to decrease operating costs because the insurer will no longer have to maintain the mainframe after it is switched off next year. The expected savings are millions of dollars a year on systems maintenance and operations.

Beyond an updated claims-processing application that is compatible with the insurance firm's global open-system platform, the software modernization created a solid foundation for future enhancements to meet evolving business and technology requirements.

Refactoring by the numbers

1.2 million lines of COBOL code and 1,400 objects

22 million claims, 2.5 billion database records, 250 tables, and 200 sequential and ISAM files

Deloitte innoWake™ tools used: development, enabler, and operations

Conclusion

The journey to modernization can be daunting, and determining the right modernization options to fit strategic priorities can be challenging. Often a long process, modernization requires fundamental changes to the platform hosting solutions and the implementation of tools to support critical business needs.

It's important for organizations to consider how to approach the interim architectural state while planning for the future target architecture, and many IBM hardware and software product offerings can be reused or incrementally integrated to support ongoing modernization efforts. This transitional period of modernization can take months—or even years—to complete, as many organizations have large, tightly coupled solutions to transform.

In conclusion, the changes required to lay the foundation for a successful application modernization project are often significant departures from current IT ecosystems, including:

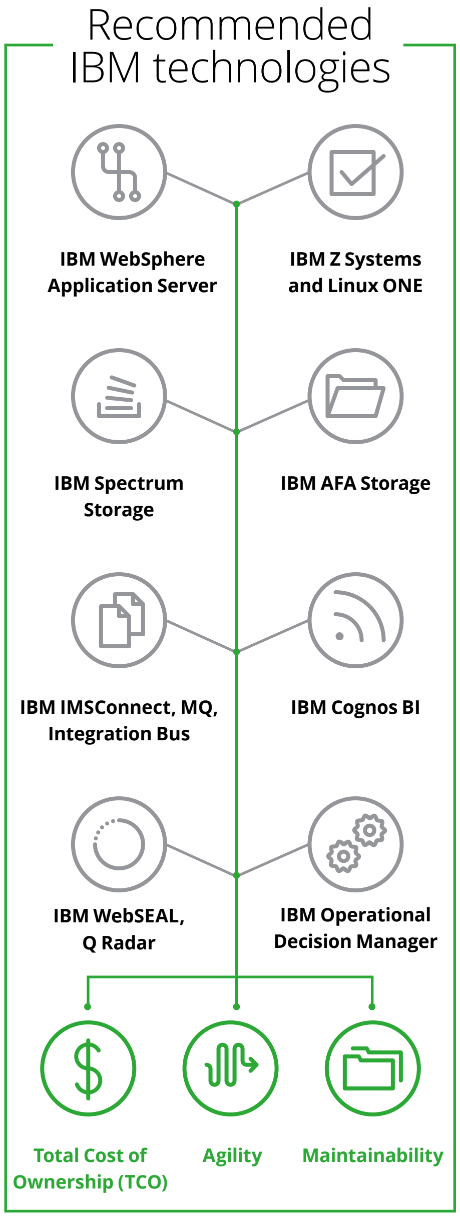
- **Deployment to a new hosting platform**
- **Implementation of a new security product to securely access business solutions**
- **Integration of middleware products, which allow the legacy and modernized solution to continue operating as needed**

Once organizations move beyond the initial modernization step, leaders can focus on how to improve or optimize business solutions while simultaneously managing a complex IT ecosystem. Regardless of where an organization is on the journey to modernization, key business drivers are:

- **Reducing their total cost of ownership**
- **Increasing agility**
- **Improving maintainability**

Over this transformational period, executives can reduce risks by utilizing reference architectures that focus on IBM-supported technologies to bridge legacy and new platforms. Organizations that attempt to supplement their architectures with custom utilities or unsupported tools may experience greater technical debt, unsupported cross-platform support, and complex ecosystems that could be difficult to troubleshoot in the future. As such, leveraging products from a single vendor can ultimately reduce friction, risks, and eventual costs during criticals period of transition for business and IT, even if new resources need to be hired or trained for specific technical expertise.

IBM offers many hardware and software products that support modernization objectives, ranging from platform options to security products and middleware tools—all which can be used to alleviate pain points during initial modernization and transitional periods throughout the process.



Recommended IBM technologies

IBM WebSphere Application Server

IBM Z Systems and Linux ONE

IBM Spectrum Storage

IBM AFA Storage

IBM IMSConnect, MQ, Integration Bus

IBM Cognos BI

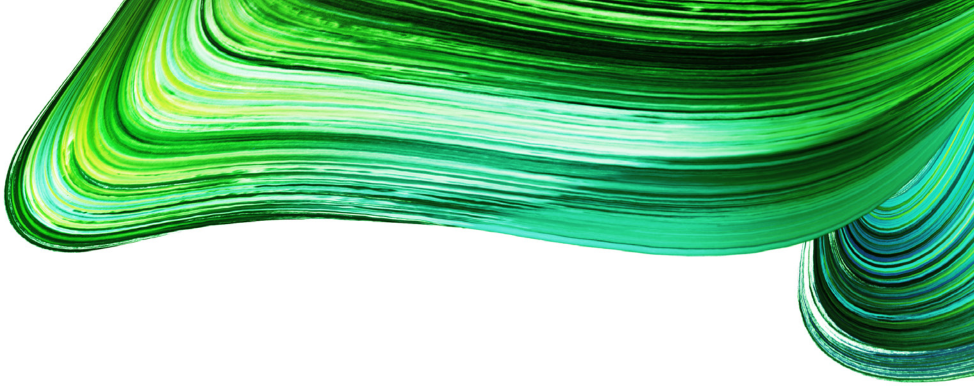
IBM WebSEAL, Q Radar

IBM Operational Decision Manager

Total Cost of Ownership (TCO)

Agility

Maintainability



Authors

Hemang Dholakia

System Architect Specialist Leader,
Deloitte Consulting LLP
hdholakia@deloitte.com

Venu Bommenani

Technology Project Manager
Senior Manager, Deloitte Consulting LLP
vbommenani@deloitte.com

Pete Evans

System Architect
Specialst Master, Deloitte Consulting LLP
petevans@deloitte.com

Mike Zawacki

Technology Project Manager
Manager, Deloitte Consulting LLP
mzawacki@deloitte.com

Sara Butt

Technology Project Manager
Manager, Deloitte Consulting LLP
sabutt@deloitte.com

Rinku Sinha

Business Technology Analyst,
Deloitte Consulting LLP
rinsinha@deloitte.com

Contributors

Robert Miller

Lead Technical Architect | IBM Alliance
Senior Manager, Deloitte Consulting LLP
robmiller@deloitte.com

Rakesh Chandrasekaran

Manager, Deloitte Consulting LLP
rakchandrasekaran@deloitte.com

Falice Wang

Business Technology Analyst,
Deloitte Consulting LLP
faliwang@deloitte.com

As used in this document, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.

This publication contains general information only and none of Deloitte Touche Tohmatsu Limited, its member firms or their related entities (collectively, the "Deloitte Network"), is, by means of this communication, rendering professional advice or services. Before making any decisions or taking any action that may affect your finances, or your business, you should consult a qualified professional adviser. No entity in the Deloitte Network shall be responsible for any loss whatsoever sustained by any person who relies on this communication.