Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Modernizing Cruise Reservation Systems: A Scalable Data Access Architecture for Digital Transformation

Rajkumar Chindanuru
Anna University, India
Corresponding Author: Rajkumar Chindanuru, **E-mail**: rchindanuru@gmail.com

| ABSTRACT

The legacy cruise reservation system, being nearly thirty years old, has built up a lot of business logic that makes its modernization a major challenge due to the aging technological support structures. The document presents a new architectural solution in the Cruise Reservation Life Expansion project, which came up with a modular Data Access Layer so as to separate database interactions from business logic. Instead of undertaking wholesale system replacement, the architecture facilitates an incremental change without destroying the functional areas of operation. The design establishes a dual mode functionality of the traditional file I/O and SQL-based access, which are contained within service programs that offer homogeneous interfaces of database functions. The architecture that resulted shows significant gains in performance, maintainability, and extensibility at minimum operational disturbance. The patterns of architecture mentioned can be seen as an important input to the organizations in various fields that have similar modernization challenges of the legacies.

| KEYWORDS

Legacy System Modernization, Data Access Layer, IBM iSeries Architecture, Incremental Transformation, Cruise Reservation Systems.

| ARTICLE INFORMATION

ACCEPTED: 03 October 2025 **PUBLISHED:** 06 October 2025 **DOI:** 10.32996/jcsts.2025.7.10.19

1. Introduction

The cruise sector is dependent on the reservation systems that were developed in the early days of computing, and most of the online platforms are outdated compared to contemporary digital demands. These legacy systems are the technological platform of key operations, such as cabin inventory to passenger manifests. However, in a fast-changing and digital market, such aging technologies are finding it difficult to cope with the modern business demands despite their outstanding durability. These systems involve a significant organizational investment that goes beyond the financial aspect to decades of business logic and business operations that are necessary in the cruise operations. [1].

Conventional reservation systems pose a serious challenge to digitisation initiatives. The legacy architectures are usually characterized by tightly coupled designs that have moderate modularity and present challenges to add new functions or be integrated with newer platforms. With the complexities of maintenance increasing exponentially with the age of the system, the IT budgets are under increasing pressure, yet yielding lower returns. The difficulties of integration are especially apparent when it comes to the introduction of mobile booking opportunities, real-time inventory control, or customized customer experience. These are technical constraints that the positioning of competition immediately affects because the digital touchpoints are ever more at the center of the customer journey. [1].

The Cruise Reservation Life Expansion project is a strategic project that tackles these problems by being innovative in terms of architecture, instead of being wholesale. This solution embraces the prohibitive cost, operational risks, and business disruption that would be involved in the total replacement of the existing reservation systems. The project created a modular Data Access

Layer that separates database interactions and business logic, providing a baseline of gradual modernization. This technology maintains useful business regulations but adds contemporary architectural elements that increase flexibility, performance, and integrations. [2].

The study explores a basic quandary of businesses in any given industry: developing technical infrastructure without leaving behind huge investments in current systems. Its meaning is not confined to cruise operations to any industry that is dependent on legacy transaction processing, such as retail, banking, healthcare, and other travel services at large. The architectural strategy provides a compromise to the preservation of old systems and the adoption of high-risk replacement, bringing viable tactics to increase useful lifespan as well as provide contemporary potentials. [2].

The paper discusses the entire process of modernizing cruise reservation systems, which starts by examining the legacy systems, architecture design, and implementation, evaluating the business implications, and finall,y lessons learnt. The focus is all through on the feasible architectural designs applicable in an industry that struggles with comparable issues of modernization, and how careful designs can help in a transition between the old system and the new technological demands.

2. Legacy System Assessment and Modernization Strategy

The old cruise reservation systems provide a complicated terrain that needs careful evaluation of the system before any modernization efforts are taken. The systems are decades old and have been built with functionality layers that are essential to the business processes. The current state analysis also demonstrates the architectural tendencies that were common to the systems created during the pre-Internet period: the monolithic design, a close-knit system, and procedural programming paradigms. A combination of technologies across generations is often part of the operational infrastructure, posing serious integration challenges. These systems usually deal with the most important reservation activities such as inventory management, prices, booking, processing of payments, and passenger information. With the rising popularity of digital transformation efforts in the industry, such legacy architectures are becoming a major impediment to innovation and responsiveness to market forces. The resulting technical debt that builds up in such environments has shown itself in maintenance challenges, a lack of extensibility, and difficulty in integrating with modern customer-facing technologies. [3].

The IBM iSeries environment has proven to be very reliable and capable of processing transactions; it has very specific technical limitations that affect modernization strategies. The architecture has a distinctive single-tiered storage design, which is quite different from distributed computing environments. RPG and COBOL are considered the traditional languages of development of program objects in these systems, which creates a challenge in the availability of skills in the modern markets of technology. Database access techniques have been native record-level access, not SQL, making it difficult to interact with the current data processing system. The combined form of the operating system and database management components brings both benefits in terms of stability in operations and problems in terms of modernization. The process of development tooling and deployment is very different compared with mainstream platform, which demands domain-specific knowledge during the modernization process. Although these limitations exist, the iSeries platform still shows impressive performance features for transaction processing loads, which present a solid base on which the updated architectures can be developed. [3].

Business continuity is another important factor in the modernization of the reservation system. Cruise booking services are available 24/7, and this means that traditional implementation methods with long periods of downtime can hardly be practiced. Reservation systems are at the core of revenue-generating processes, and therefore, any failure has a financial impact. The customer experience is a completely changed aspect, and the experience of booking online has become an important factor in making purchases. Distribution channel integration is now more valuable with the change in the booking trends toward third-party services that need real-time access to inventory. These business requirements drive the need to modernize in a fashion that allows the continuity in operations, but gradually adds new, improved functionality. This intensity is further enhanced by the competitive environment, with digital capabilities becoming the key to differentiating the offerings by the market leaders to grab the market share. [4].

Modernization strategies are characterized by entirely different risk profiles, resource demands, and transformational schedules. Total system renewal entails the creation of new reservation platforms in totality and the retention of the existing systems until a cutover. This strategy provides a full transformation, but it is associated with huge business risk, long implementation periods, and requires a lot of resources. On the other hand, strategic rejuvenation with incremental modernization aims at modifying the existing system by gradually converting it to architectural additions, component re-factoring, and selective replacement. This method usually portrays better results in reducing business discontinuity, besides facilitating incremental adoption of advanced abilities. The choice between the two approaches will be based on the organizational variables such as risk tolerance, availability of resources, competition, and the technical status of the current systems. [4].

Aspect	Characteristics	Challenges
System Age	Decades old with accumulated functionality layers	Maintenance difficulties increase exponentially with age
Architecture	Monolithic design, tightly coupled components, procedural programming paradigms	Limited extensibility, difficult integration with modern systems
Technical Environment	IBM iSeries with a unique single-level storage model	Skills availability challenge in contemporary markets
Programming Languages	Traditional RPG and COBOL	Limited developer pool, outdated paradigms
Database Access	Native record-level access methods	Complicates integration with modern data processing frameworks
Business Criticality	24/7 operation supporting revenue-generating processes	Downtime has a significant financial impact
User Expectations	Digital booking experiences influence purchase decisions	The gap between current capabilities and customer expectations

Table 1: Legacy System Assessment and Challenges [3, 4]

3. Data Access Layer Architecture Design

The concept of the Data Access Layer (DAL) architecture of modernizing cruise reservation systems adopts key layout rules that provide the equilibrium of innovation and pragmatic implementation factors. The architectural design uses high cohesion, loose coupling, and information hiding to provide a sustainable base for the system's evolution. The architecture allows the separation of concerns between business logic and data persistence mechanisms by having a multi-layered architecture to allow independent component evolution. The design is based on known patterns such as Facade, Repository, and Service Locator to control complexity as well as code reuse. Such a principled approach provides some ground upon which to proceed with the gradual modernization, which will not interfere with the most important business processes.

The modular component structure takes a structured approach to functionality by breaking down functionality into discrete and interchangeable units that have clearly defined responsibilities. This structure is inclusive of service interfaces, business entities, repository implementations, and technology-specific adapters, all of which operate autonomously by having clearly defined contracts. The modular method provides a phase implementation strategy; it is possible to implement an implementation strategy based on the business priorities rather than the technical dependencies. Boundaries of components provide natural testing, deployment, and scalability planning seams, providing realistic channels of evolutionary system development in complex operational environments. [5].

The generic data access API specification defines a standard interface used to interact with a database, making the common operations, such as retrieval, creation, modification, and removal of business entities standard. The API design has a more business-focused approach, using domain objects instead of database-specific objects, and increases the level of abstraction for application developers. Handling of errors, managing transactions, and security are explicitly considered and are therefore always implemented in all components. The API specification provides a baseline on which business logic and data storage can be developed together and migrated incrementally between storage technologies by providing a stable contract between the two. [6].

CRUD operations abstraction protects business logic by the degree of complexity of data persistence operations and converts domain-focused operations into data optimization interactions. The technical issues dealt with in this abstraction include connection management, statement preparation, and mapping results in all types of operations. The architecture isolates such basic operations in a consistent structure, which minimizes repetitive code and enhances quality with standard patterns of implementation. This abstraction layer forms the base in supporting various access patterns, including both traditional record-level access as well as the latest SQL-based methods necessary to support gradual migration strategies in modernization efforts on reservation system initiatives. [6].

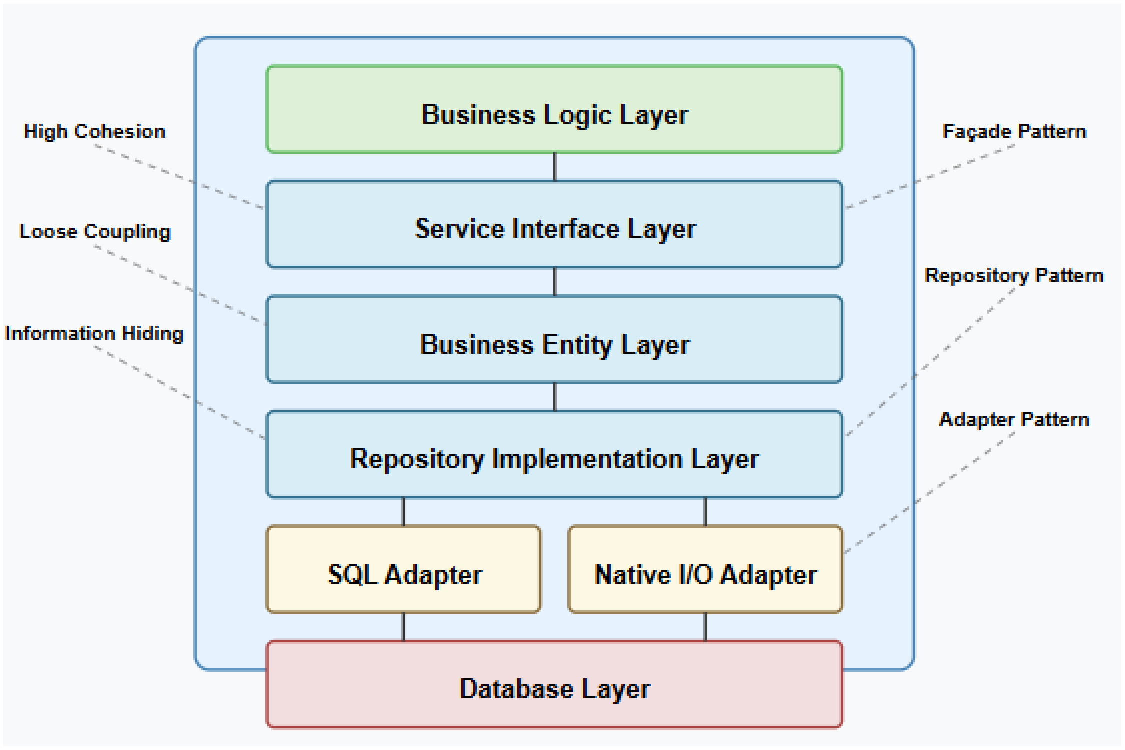

Fig 1: Data Access Layer Architecture [5, 6]

4. Technical Implementation

The technical side of the Data Access Layer had been implemented with the use of ILE RPG together with embedded SQL, the combination of which served to provide an interface between legacy file-based programming and the newest database access methods. Integrated Language Environment (ILE) RPG has close merits in terms of modernization efforts as it is modular by nature and easily integrates with the IBM i platform. The syntax of the free-format RPG language is more readable and maintainable than fixed-format code, and supports modern programming paradigms. Its embedded SQL statements, embedded directly into RPG procedures, allow manipulation of data sets in addition to the old-fashioned record-based access. Such a blend maintains consistency with current systems but offers contemporary data access patterns without the need to completely rewrite applications. The kind of implementation followed was the incremental change approach, which enabled incremental integration of SQL techniques in the existing business processes. [7].

Service program encapsulation is another implementation methodology of basic implementation technology, where related processes are organized together in a reusable, coherent module to support the entire architecture. The service programs (SRVPGMs) apply normalized interfaces and conceal the implementation details, establishing natural boundaries between functional domains. The binding directories promote a dynamism in the deployment and servicing of procedures by facilitating the resolution of calls at runtime. Export control by judiciously selected binder source members has the advantage of restricting public interfaces to only necessary procedures, ensuring internal implementation information remains secret, and minimizing coupling between components. This encapsulation style enhances the maintainability, as alterations are localized and the amount of duplicated code across the system is minimized, and more importantly, parallel development within different business domains is supported. [7].

Bidirectional compatibility between native file I/O and SQL-based access offers the necessary flexibility in incremental modernization. It is implemented using the patterns of adapters to abstract particular access mechanisms, where the business logic can access the data using standard interfaces in spite of the underlying technology. Native file I/O still proves to be more efficient in simple single record operations, where direct record access gives the best performance, and SQL-based access gives the best performance with complex queries having more than one condition, join, and aggregation requirements. Such a mixed strategy allows the optimization process to depend on the particular access patterns, taking advantage of the capabilities of each technology and reducing the shortcomings. The dual-mode feature helps in the gradual migration of record-level to SQL operations, and this allows gradual implementation without interfering with critical business functions. [8].

Implementation Aspect	Technologies/Methods	Benefits
Development Environment	ILE RPG with embedded SQL	Bridge between traditional and modern access methods
Code Structure	Free-format RPG syntax	Enhanced readability and maintainability
Encapsulation Strategy	Service programs (SRVPGMs)	Creates natural boundaries between functional areas
Procedure Resolution	Binding directories	Runtime flexibility in deployment and maintenance
Interface Control	Export control via binder source members	Restricts public interfaces to essential procedures
Access Pattern	Adapter patterns for dual-mode support	Consistent interfaces regardless of underlying technology
Performance Optimization	Mode selection based on operation type	Leverages the strengths of each technology
Migration Approach	Incremental transformation	Gradual incorporation without disrupting business processes

Table 2: Technical Implementation Approaches [7, 8]

5. Results and Business Impact

The adoption of the updated Data Access Layer architecture provided quantifiable results in various aspects of system performance and maintainability. Analysis of the performance undertaken during the implementation process revealed a lot of improvement in the efficiency of the transaction processing on the core reservation operations. The response time was also improved, especially in complex booking situations when a number of passengers and stateroom arrangements had to be involved. The simulated load testing in peak conditions showed improved stability, and there was consistency in the transaction completion speed even at peak times. The multi-level caching design of the architecture also added to the performance improvement, especially for the most actively used reference data like sailing schedules and pricing information. The customer experience was directly boosted through these performance improvements, as it was found to reduce the time spent booking complete services and also minimize system latency in times of peak usage. [9].

The building design minimized the redundancy of programming across the system significantly, and it resolved a systemic problem of legacy reservation systems. Before modernization, the patterns of duplicate data access were present all over the codebase, which posed serious difficulties in terms of maintenance and uneven patterns of implementation. Standardized Data Access Layer brought such varied implementations together, creating a standard in the mode of database interactions throughout the entire system modules. This consolidation led to a highly reduced volume of code and functional equivalence. The lower complexity and existing patterns facilitated development efficiency through new feature development, as it allowed delivering business capabilities faster. [9].

The improvements in maintainability and extensibility were the key results of the modernization activity that changed the sustainability profile of the system in the long term. The metrics of the static analysis showed that there were significant improvements in the dimensions of code quality, such as complexity, coupling, and cohesion. The architecture style allowed more successful modularization, making the system components distinct and defining the interface that will be used to communicate between modules. The extension points that were added at strategic points in the architecture gave clear directions on what can be upgraded in the future, and the system could be extended without affecting the architectural integrity. All these enhancements made the system no longer a limiting force on business innovations, but provided new capabilities and business value. [10].

Impact Area	Improvements	Business Value
Performance	Enhanced transaction processing efficiency, reduced response times	Improved customer experience, reduced booking completion times
Stability	Consistent transaction completion during peak periods	Minimized system latency during high-volume periods
Code Quality	Reduced programming redundancy, consolidated implementations	Substantial code volume reduction while maintaining functionality
Development Efficiency	Established patterns for database interactions	More rapid delivery of business capabilities
Maintainability	Improved complexity, coupling, and cohesion metrics	Reduced maintenance effort and improved system sustainability
Extensibility	Strategic extension points throughout the architecture	System evolution without compromising architectural integrity
Business Capability	Transformation from constraint to enabler	Foundation for new capabilities and business value

Table 3: Results and Business Impact Metrics [9, 10]

6. Conclusion

The Data Access Layer design of cruise reservation systems reflects how legacy systems can be upgraded to reflect the new business requirements without damaging existing intellectual property or disrupting the business operations. The architecture forms a basis of sustainable digital transformation by adopting the principles of modular design, service-oriented encapsulation, and dual-mode data access. Incremental modernization has been proven as an effective alternative to risky replacement strategies, especially when dealing with mission-critical transaction systems, through its successful implementation. The patterns of architectural work outlined cut across the cruise industry and have the potential to be used in retail, banking, healthcare, and more of the travel arena in general. Possible future upgrades are further integration with mobile platforms, additional analytics support, and use of cloud-based models of deployment, further developing the trend of the fully developed, modernized reservation functionality, based on the established business logic.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Adelina Z (2024) From Traditional to Digital: The Evolution of Business Models in Hospitality Through Platforms, MDPI, 2024. https://www.mdpi.com/2813-4176/2/4/15
[2] Adya M, (2020) Legacy System Modernization: Effective Strategies and Best Practices, IJLRP, 2020. https://www.ijlrp.com/papers/2020/8/1245.pdf
[3] Andrew S, (2021) Legacy Systems Modernization for Travel Enterprises, DataArt, 2021. https://www.dataart.com/blog/legacy-systems-modernization-for-travel-enterprises
[4] KFA, (2018) Application Modernisation – Characteristics of Modern IBM i Applications (Part 3), 2018. https://www.kfa.co.uk/blog/application-modernisation-characteristics-modern-ibmi-applications-part-3/
[5] Kowsick V, (2024) From Monolith To Microservices: A Practical Guide To Legacy Application Modernization, IJRCAIT, 2024. https://iaeme.com/MasterAdmin/Journal_uploads/IJRCAIT/VOLUME_7_ISSUE_2/IJRCAIT_07_02_107.pdf
[6] Nadir K. A et al., (2014) Modernizing IBM i Applications from the Database up to the User Interface and Everything in Between, IBM, 2014. https://www.redbooks.ibm.com/redbooks/pdfs/sg248185.pdf
[7] Olufunmilayo O et al., (2023) Modernizing Legacy Systems: A Scalable Approach to Next-Generation Data Architectures and Seamless Integration, *International Journal of Multidisciplinary Research and Growth Evaluation*, 2023. https://www.allmultidisciplinaryjournal.com/uploads/archives/20250306182550_MGE-2025-2-018.1.pdf
[8] Rajkumar C, (2025) Systematic ROI Assessment Framework for Legacy System Modernization, SSRN, 2025. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5181205
[9] Ricardo P et al., (2015) Understanding Legacy Architecture Patterns, ENASE, 2015. https://www.scitepress.org/papers/2015/54673/54673.pdf
[10] Zahir I et al., (2023) The impact of legacy systems on digital transformation in European public administration: Lessons learned from a multi-case analysis, ScienceDirect, 2023. https://www.sciencedirect.com/science/article/pii/S0740624X22001204