Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN: 2468-4376 https://www.jisem-journal.com/ **Research Article**

Why AI Projects in Banks Fail Without the Right Infrastructure Modernization

Swati Kumari
NucleusTeq, USA.

ARTICLE INFO

Received: 23 Jan 2026

Revised: 28 Jan 2026

ABSTRACT

Everywhere across the globe, banks are increasingly viewing artificial intelligence as a revolutionary force that is likely to transform fraud detection, customer service automation, targeted financial advisory services, and risk management operations. Even with huge investments in AI technologies, many banking organizations face great challenges in deploying machine learning systems at scale in production environments. The main constraint often arises not from algorithmic constraints or lack of data science talent but from substandard technological infrastructure unable to accommodate the scale of modern AI workloads. Legacy banking platforms built many years ago run on batch-processing architectures that are inherently incompatible with real-time data streaming demands necessary for today's machine learning use cases. Mainframe environments using inflexible data schemas cannot support heterogeneous unstructured data sources required for training advanced neural networks, while limitations of aging hardware constrain the computation of intricate algorithms at the enterprise level. Data fragmentation within departmental silos hinders accessibility, undermining model precision and predictive power. Infrastructure modernization is an unavoidable requirement for the effective implementation of AI, involving the adoption of cloud computing for computational elasticity, containerization technologies for standardized deployment across environments, single unified data platforms tying together dissimilar information sources, and compliance-aware architectures meeting very stringent regulatory requirements. Banking organizations are required to address systematic infrastructure constraints through in-depth transformation programs combining technical upgrades with organizational change management, skills development initiatives, and cross-functional collaboration models. Only by conscious infrastructure development can financial services companies release the full transformative power of artificial intelligence and realize enduring competitive gains in ever more digital banking environments.

Keywords: Artificial Intelligence Infrastructure, Cloud Computing Modernization, Legacy System Constraints, Data Architecture Integration, Regulatory Compliance Framework, DevOps Transformation

Introduction

Adoption of AI in banking is expanding significantly, with the span of applications ranging from fraud prevention to personalized financial guidance, customer service automation, to risk examination. The evolution is being fueled by sophisticated machine learning methods that are able to handle large volumes of customer interaction data to derive useful insights and forecast behavioral tendencies. Machine learning methods have exhibited impressive performance in classifying customer sentiment and satisfaction based on natural language processing of social media posts, with classification accuracy rates higher than 85% in separating positive, negative, and neutral customer feedback in large-scale datasets with tens of thousands of customer interactions [1]. In customer service applications, AI systems utilize such sentiment analysis features to route queries automatically, distinguish between urgent cases and other instances, and pick out unhappy customers that need immediate attention,

allowing human agents to concentrate on solving tricky problems while minimizing average response times from hours to minutes [1]. The technology holds great value creation for international banking organizations by way of improved operational effectiveness and enhanced customer satisfaction, with predictive analysis offering more precise credit risk determination by using ensemble learning techniques that bring together several decision trees to provide higher prediction performance than individual models [2].

Yet, this promise is often hampered by the technology platform on which AI solutions operate. Most banks continue to run on decades-old legacy systems, with their core banking platforms having an average of 20 to 30 years of run life and being hosted on mainframe architectures that were created for batch processing cycles—typically running overnight processes for transaction reconciliation and account updates—versus the real-time streams of data demanded by current AI use cases. The integration issues are immense since modern AI architectures demand scalable tree boosting systems that can handle billions of training data points and millions of features while maintaining processing efficiency using parallel processing and distributed computing capabilities unavailable to traditional infrastructure [2]. Such contemporary gradient boosting frameworks get their performance from advanced optimization methods such as regularized learning objectives, column block organization for parallel learning, and cache-conscious algorithms with reduced memory access patterns, all of which require computing infrastructures radically different from conventional banking systems [2].

Although banks are spending a lot on AI capabilities, most projects get bogged down or fail due to infrastructural unpreparedness. Legacy systems, aged data structures, and compliance-driven environments pose resistance to the deployment of scalable, real-time AI models that need the capability to manage sparse data representations, weighted dataset processing for imbalanced classification issues prevalent in fraud detection, and out-of-core computation support for datasets larger than available memory capacity [2]. Fragmentation of data across departmental silo systems implies customer details can be dispersed throughout multiple databases and applications, hindering the development of consistent feature engineering pipelines required for successful machine learning model training. Computational needs for training advanced ensemble models—models that derive their state-of-the-art performance using methods such as sparsity-aware split finding algorithms and parallel tree creation—are far beyond the capacities of aging on-premises infrastructure based on transactional processing instead of intense analytical workloads typical of contemporary machine learning operations [2].

This piece tackles the frequently neglected topic of infrastructure as the bottleneck in enterprise AI success. The subsequent sections delve into how infrastructure modernization serves as an enabler at the foundation of successful AI endeavors in banking, touching on the technical constraints of legacy systems and recommending critical modernization strategies that bring infrastructure roadmaps in line with strategic AI objectives.

The Infrastructure Gap in Banking AI

Legacy System Constraints

Legacy banking infrastructure was built with batch processing and periodic refreshes in mind, not the real-time, continuous flows of data modern AI systems demand. Legacy environments are usually running on mainframe architectures with inflexible data schemas that cannot accommodate the heterogeneous, unstructured data sources AI models require for training and inference. Legacy banking systems carry out transactions based on traditional business process management standards in which activities are grouped into sequential workflows with well-defined start and completion states, control flow patterns, and data dependencies that determine how information flows through the firm [3]. Banking business process management frameworks have traditionally focused on an orchestrated

process where customer onboarding, loan approvals, and account maintenance follow predetermined paths with human decision-making nodes and manual data validation steps involved to create process execution cycles taking hours or days instead of the millisecond response time that current AI systems demand [3]. This design limitation is inherently at odds with current AI needs, in which fraud detection rules must examine patterns of transactions in real-time and recommendation systems must constantly examine streams of customer activity without the strict process delineations present in legacy workflow management systems [3].

The processing limitations of legacy hardware also limit the capacity to execute advanced machine learning models at scale. Legacy banking infrastructure runs on computing paradigms optimized for online transaction processing, in which specific operations finish fast, but the entire system architecture cannot support the large parallel data processing workloads that are typical of contemporary machine learning operations [4]. Contemporary AI model training necessitates distributed computing frameworks that can automatically partition big data across commodity hardware clusters, manipulating data in parallel using map operations that convert input records independently and then combining results using reduce operations that merge intermediate outputs into final computational results [4]. Training advanced neural network models from customer transaction histories involves processing terabytes of historical data using iterative optimization algorithms, workloads that gain enormously from parallel processing architectures in which hundreds or thousands of computing nodes process distinct data partitions in parallel, completing in times measured in hours instead of the weeks or months it may take on conventional sequential processing systems [4]. The MapReduce model illustrates how big data processing can be made easier by hiding the complexity of parallelism, fault tolerance, data partitioning, and load balancing, enabling programmers to concentrate on defining map and reduce operations while the underlying runtime system takes care of executing over distributed clusters automatically, scheduling jobs, coordinating inter-machine communication, and recovering from node failures [4]. Legacy banking infrastructures do not have these distributed computing capabilities, such that data scientists must either experiment with small sample sizes that do not contain critical patterns or wait a very long time for training tasks to finish on suboptimal infrastructure, severely impacting the iterative experimentation cycles necessary for the creation of effective AI models [4].

Data Accessibility Issues

AI models live on thorough, unified data sets, but most banks struggle with data stuck in departmental silos. Customer information, transaction records, and operational data often reside in disconnected systems that lack interoperability, creating organizational and technical challenges rooted in how business processes have historically been designed and managed within financial institutions [3]. Business process management methods commonly structure workflows according to functional departments and particular business targets, resulting in process fragmentation whereby customerfacing activities such as opening an account, loan processing, and service requests each run as separate process instances with their respective data repositories and information handling practices [3]. Lacking a common data access layer, AI models are unable to build end-to-end insights or provide accurate predictions, since the process-centric focus of banking operations brings natural silos within organizational units that impede complete data integration throughout the customer lifecycle [3]. Technical complexities reach beyond mere data integration into capturing basic differences in operational process data generation and consumption, with transactional systems being geared towards high-speed single record updates and analytical AI systems needing batched access to preaggregated historical data spread across multiple process runs and business scenarios [3].

The lack of live data pipelines implies that even if data is available, it comes in too late to support responsive AI programs like real-time fraud detection or customized product offers. Distributed data processing architectures have shown the ability to process terabytes of data on thousands of machines

using fault-tolerant models of execution where failures of individual tasks automatically cause reexecution on different nodes without necessitating the restart of the entire job, but exercising these capabilities necessitates fundamental architectural changes from the traditional banking systems [4]. The challenge also includes ensuring data locality optimizations where computation is scheduled on nodes holding data partitions of concern so as to reduce network traffic, and designing efficient combiners that do local aggregation prior to network transmission to minimize bandwidth usage during shuffle stages, where intermediate map outputs are moved to reduce tasks [4]. Legacy banking infrastructure is not equipped with such advanced data movement and processing orchestration functions, rendering it incapable of reaching the processing throughput and latency characteristics real-time AI applications require for fraud detection patterns across millions of concurrent transactions or personalization engines supporting thousands of simultaneous customer interactions [4].

Infrastructure Dimension	Traditional Banking Systems	Modern AI Requirements	Impact on AI Initiatives
Data Processing Model	Batch processing with overnight cycles and scheduled updates	Real-time streaming analytics with immediate processing	Prevents responsive fraud detection and dynamic customer personalization
Computational Architecture	Centralized mainframe systems with sequential processing	Distributed parallel processing with GPU acceleration	Cannot handle neural network training or complex algorithm execution
Data Schema Design	Rigid relational structures with predefined formats	Flexible schema-onread supporting multiple data types	Blocks integration of unstructured customer communications and behavioral data
System Integration	Siloed departmental applications with manual transfers	Unified platforms with automated integration pipelines	Creates fragmented customer profiles and prevents holistic AI insights
Scalability Mechanism	Fixed capacity with lengthy procurement processes	Dynamic allocation with instant provisioning	Causes resource contention and inefficient infrastructure utilization

Table 1. Legacy Infrastructure Limitations Impacting AI Deployment in Banking [3, 4].

Key Modernization Strategies

Cloud and Hybrid Architecture Adoption

Transition from on-premises infrastructure to cloud or hybrid environments offers the computational elasticity that AI workloads require. Cloud platforms provide scalable storage of enormous training datasets and elastic compute assets that can be dynamically provisioned according to model needs, allowing banks to take advantage of the five fundamental characteristics that are necessary for cloud computing: on-demand self-service in which customers can unilaterally provision computing resources without human intervention from service providers, broad network access via standardized mechanisms with an emphasis on use across various heterogeneous client platforms, resource pooling in which provider computing resources are used by multiple consumers based on a multi-tenant model using varying physical and virtual resources dynamically allocated based on demand, rapid elasticity in

which capabilities are elastically provisioned and de-provisioned to quickly scale with demand, and measured service in which cloud systems automatically monitor and manage use of resources with metering capabilities fitted to the type of service [5]. Advanced cloud architectures use advanced resource management systems based on service models such as Infrastructure as a Service in which consumers can provision processing, storage, networks, and other basic computing resources and deploy any software such as operating systems and applications, Platform as a Service in which consumers can deploy consumer-developed applications using programming languages, libraries, services, and tools supported by the provider, and Software as a Service in which consumers employ provider applications running on cloud infrastructure [5]. The deployment models that are accessible to banking organizations are private clouds wherein infrastructure is provisioned for singleorganization exclusive use by multiple consumers, community clouds wherein infrastructure is shared by multiple organizations serving a particular community with common interests, public clouds wherein infrastructure is provisioned for general use by the public, and hybrid clouds that consist of two or more different infrastructures which are standalone entities but are connected through standardized or proprietary technology that provides data and application portability [5].

Hybrid infrastructures enable banks to keep sensitive information locally on-premises while taking advantage of cloud resources for demanding processing workloads, meeting performance requirements as well as data residency regulatory concerns. This architectural style deploys secure connectivity protocols between public and private cloud infrastructures, facilitating workload distribution models in which sensitive customer data processing is accomplished within private cloud infrastructures while computationally expensive model training processes take advantage of the virtually boundless capabilities of public cloud infrastructures [5]. Hybrid deployments allow banks to adopt tiered data management plans that align with cloud deployment models, in which sensitive production systems run in private cloud setups with exclusive infrastructure and added security controls, development and testing workloads use partner-shared community cloud infrastructure, and non-sensitive analytics processing uses public cloud scale for cost-saving purposes [5]. The fundamental attribute of rapid elasticity becomes especially useful for AI applications that have highly fluctuating patterns of resource utilization, since model training stages may consume enormous computational power for hours or days and then experience long periods of low resource utilization in the case of model testing and validation, enabling banks to provision resources based on real demand instead of having to pre-provision permanent infrastructure to meet peak workloads [5]. Security designs in hybrid environments need to respond to the extensive network access feature of cloud computing by employing end-to-end encryption, authentication, and authorization systems that secure information crossing public networks while allowing for the smooth resource pooling and metered service attributes that provide cloud computing economic value [5].

Containerization and Microservices

Containerized designs allow AI models to be deployed with dependencies and run reliably across dev, test, and production environments. Container technologies descended from Linux Containers as operating-system-level virtualization technologies that execute several isolated systems on a single control host with kernel-level namespaces and cgroups to manage resources, with Docker being a revolutionary platform that unified and streamlined container creation and deployment through the use of layered file systems, image registries, and standardized runtime interfaces [6]. This approach resolves compatibility issues that plague legacy systems where AI components must interface with decades-old applications, as containers encapsulate entire runtime environments in portable images that execute identically across different infrastructure platforms, eliminating the dependency conflicts and environmental inconsistencies that traditionally required extensive configuration management and troubleshooting during application deployment [6]. Container orchestration tools such as Kubernetes offer auto-deployment, scaling, and management features through advanced cluster management tools that abstract the infrastructure, schedule the containers on available nodes according to resource needs

and placement rules, maintain desired state with automatic restart of crashed containers, and roll out updates where the old versions of containers are replaced by new ones in phases while keeping the service up and running [6]. The isolation of resources afforded by containerization enables several AI models with different dependency demands to be hosted on common infrastructure using kernel-level isolation techniques that make every container believe it has a dedicated execution environment with a personal file system, process address space, and network stack, while the container runtime optimally shares the underlying operating system kernel and hardware resources among all containers executed on a host [6].

Microservices architecture enables AI features to be independently developed and updated without affecting core banking processes, speeding up innovation cycles without compromising system stability. Such an architectural style converges well with containerization since microservices are often deployed as container images that can be replicated across cluster nodes for horizontal scaling, updated separately by way of rolling deployment patterns, and governed by orchestration platforms that manage service discovery, load balancing, and health checking [6]. The transition from monolithic, legacy applications to containerized microservices is a fundamental change in the way software is architected and deployed, with the packaging mechanism being containers and the management layer being provided by orchestration platforms such as Kubernetes [6]. Kubernetes targets the operational issues of managing containerized applications in production with features such as pods as minimal deployment units that collect related containers sharing storage and network resources, services that offer stable network endpoints for reaching sets of pods, replica sets that ensure desired counts of pod instances for high availability, and deployments that handle the declarative update process for pods and replica sets [6]. Banks that use containerized microservices architectures take advantage of the portability offered by containers because applications bundled as container images are free to run on any infrastructure that hosts the container runtime whether in onpremises data centers, private clouds, or public cloud platforms, and allow workload mobility strategies that optimize for cost, performance, or regulatory needs without applications having to change [6].

Unified Data Infrastructure

Creating data lakes or data mesh architectures builds centralized data repositories where raw and processed data exist in consumable forms. Such contemporary data platforms are capable of housing both structured transactional data as well as unstructured sources such as customer communications and market sentiment feeds, using schema-on-read patterns where data is stored in native forms without transformation upfront, enabling varied analytical workloads to read the same underlying data based on their individual needs. Cloud computing platforms support the infrastructure attributes necessary for contemporary data lakes via on-demand self-service provision of scalable storage resources between gigabytes and petabytes without human intervention, wide network access supporting data ingestion from a wide variety of source systems within the organization, resource pooling that cost-effectively stores data from different business units on common infrastructure, rapid elasticity facilitating the addition of storage capacity seamlessly as new data sources become available, and measured service allowing detailed monitoring of storage consumption and costs within various data domains and organizational units [5]. Cloud-based data lakes' elasticity provides for automatic scaling of storage as new sources of data are consumed, bypassing the capacity planning and procurement latency that puts a stranglehold on traditional data warehouses, and utilizing Infrastructure as a Service cloud models that offer the raw storage and compute resources required to construct advanced data processing pipelines [5].

By having strong data governance frameworks in place with these platforms, banks can maintain data quality and lineage traceability and democratize access for AI teams throughout the organization. Data governance deployments need to cater to the multi-tenancy nature of cloud infrastructure pooling, where multiple organizational units utilize underlying infrastructure yet require logical partitioning and

access controls, necessitating advanced identity management and authorization systems capable of working across hybrid cloud deployments that span private and public infrastructure [5]. The containerization technologies of today's data platforms allow for the deployment of specialized data governance tools as microservices that can be scaled separately according to demand, with metadata management tools, data quality monitoring tools, and lineage tracking tools, each running in its own containers orchestrated by Kubernetes clusters that provide high availability and performance [6]. Automated data quality monitoring systems take advantage of the cloud's rapid elasticity to dynamically provision processing capacity as quality validation jobs run against just-introduced datasets and release resources when validation finishes, minimizing operational expenses while maintaining full quality coverage [5].

Technology Component	Core Capabilities	Banking AI Applications	Operational Benefits
Cloud Computing Characteristics	Self-service provisioning, elastic scaling, resource pooling, and measured usage	Scalable storage for training data, dynamic compute for model training, and distributed inference serving	Eliminates procurement delays, optimizes costs, and enables global accessibility
Infrastructure as a Service	Virtual machine provisioning, storage services, and network configuration	GPU allocation for deep learning, high-performance storage, secure processing environments	Flexible resource sizing, geographic distribution, and infrastructure automation
Platform as a Service	Managed environments, automatic scaling, integrated monitoring	Machine learning platforms, automated training pipelines, serverless endpoints	Reduced operational overhead, built-in scaling, simplified deployments
Container Technologies	Lightweight virtualization, dependency packaging, portable environments	AI model encapsulation, consistent deployment, and microservices isolation	High deployment density eliminates conflicts, and rapid startup
Container Orchestration	Automated deployment, health monitoring, load balancing, rolling updates	Multi-model serving, canary deployments, and automatic scaling	High availability, zerodowntime updates, efficient resource use

Table 2. Cloud and Containerization Technologies Enabling AI Infrastructure Modernization [5, 6].

Balancing Innovation with Compliance

Regulated financial entities have to operate under very tight regulatory regimes that require data safeguarding, model explainability, and audit trails. Infrastructure modernization has to include security and compliance by design and not as an afterthought, and deploy end-to-end cryptographic security architectures that use the mathematical constructs developed through public-key cryptography in which mathematically related pairs of keys allow secure data exchange without any need for prenegotiation of keys [7]. Contemporary architectures must incorporate encryption in transit and at rest, fine-grained access controls, and comprehensive logging features utilizing cryptographic

primitives such as symmetric encryption algorithms like Advanced Encryption Standard with 128-bit block size and key sizes of 128, 192, or 256 bits offering computational security estimates of 2^126, 2^190, and 2^254 operations respectively to resist brute force attacks, asymmetric encryption methods like RSA with suggested key sizes of 2048 bits or more and Elliptic Curve Cryptography with equivalent security but much shorter 256-bit keys, and cryptographic hash functions like SHA-256 yielding 256-bit digests offering collision resistance necessary for digital signatures and integrity checking [7]. API gateways and service meshes offer the monitoring and governance layers required to prove regulatory compliance while facilitating integration of AI systems, applying digital signature schemes in which the signer digitally signs messages or transactions using their private key that anyone can validate using the respective public key for confirming authentication of the signer's identity, non-repudiation such that the signer cannot deny having signed the message, and integrity to ensure signed content was not tampered with after signature generation [7]. The cryptographic basis of such security mechanisms is based on mathematical problems that are thought to be computationally intractable, such as the integer factorization problem for RSA security in which factoring the product of two large primes continues to be computationally infeasible with present computational power, the discrete logarithm problem in elliptic curve groups or finite fields offering security for Diffie-Hellman key exchange and Elliptic Curve Digital Signature Algorithm, and the collision resistance property of hash functions in which computation proportional to the square root of the size of the hash output space is needed to find two distinct inputs that produce the same hash output [7].

Secure sandboxing environments enable data scientists to try out AI models with production-like data without breaching live systems or exposing sensitive data. Sandbox environments utilize cryptographic access controls in which user authentication is based on digital certificates holding public keys that have been signed by trusted certificate authorities, establishing chains of trust that verify user identities without centralized password databases that can be breached [7]. Versioning of code as well as data makes reproducibility and regulatory audit possible, using Merkle tree data structures where cryptographic hashes of blocks of data are recursively hashed together to generate a single root hash that represents the entire dataset in a unique way such that it allows verification of the fact that any given element of data is part of the committed dataset without needing to transmit the entire data [7]. Infrastructure that is programmed to automate compliance checks and has extensive audit logs lowers the operational burden of compliance with regulatory requirements while enabling rapid AI development, making use of cryptographic timestamping services offering verifiable evidence of when particular events took place by adding timestamps to digital signatures and blocking backdating of documents or logs and the creation of tamper-evident audit trails where any attempt at altering historical records would make cryptographic signatures invalid [7]. The authentication and authorization infrastructures behind these compliance features use challenge-response protocols in which servers challenge clients to demonstrate possession of private keys without actually having to transmit the keys themselves, zero-knowledge proof systems through which a party can demonstrate knowledge of secret information without disclosing the secret itself, and threshold cryptography schemes in which cryptographic functions involve collaboration of multiple parties such that individual parties are unable to perform sensitive operations like signing for a transaction or decryption on their own [7]. Banking AI systems require regulatory compliance to have full audit capabilities deployed via blockchain-inspired cryptographic methods where every entry in the audit log is included in a cryptographic hash chain of the prior entry, forming an unbreakable chain whereby any attempt to alter past records would destroy the cryptographic connection and be instantly visible, allowing forensic traces to aid regulators in reconstructing entire decision-making procedures with mathematical certainty regarding log integrity [7].

Compliance-conscious infrastructure deployment involves careful consideration of important management practices in which hierarchical deterministic key derivation protocols produce several cryptographic keys from a single master seed employing one-way hash functions such that key

hierarchies in their entirety can be backed up by securely storing the master seed, while derived keys can be produced on demand without securing individual key storage infrastructure [7]. Cloud service providers hosting banking AI system infrastructures are required to prove conformance with cryptographic standards such as the utilization of approved algorithms, minimum key sizes that are according to current recommendations for security, adequate random number generation with cryptographically secure pseudorandom number generators seeded with real entropy sources, and secure key lifecycle management through generation, distribution, storage, rotation, and destruction stages [8]. The distributed architecture of cloud computing presents novel compliance issues as sensitive data processing can be done over infrastructure owned by multiple parties, such as cloud providers, third-party service providers, and the banking institutions themselves, requiring entitycentric privacy and identity management strategies that preserve user control over personal data regardless of where processing actually happens [8]. Cloud architectures today solve these problems by using privacy-protecting frameworks that keep user identity information separate from the service providers, using protocols in which users authenticate to trusted identity providers who provide cryptographic credentials with minimal required attributes, presenting these credentials to cloud services without the services having to learn the user's full identity or having to interact with the identity provider [8].

The incorporation of AI systems into current banking infrastructure demands privacy-preserving authorization mechanisms that allow for fine-grained access control decisions without revealing unnecessary user attributes to service providers [8]. Entity-centric security models deploy sticky policies whereby privacy choices are cryptographically attached to data items themselves instead of being applied only at perimeter access controls to guarantee that privacy policies accompany data when it is transmitted between systems and get applied at every stage of processing [8]. Service mesh designs enforce attribute-based access control in which authorization decisions compare encrypted user attributes without service provider decryption, utilizing cryptographic methods like predicate encryption that enable conditional decryption based on whether the encrypted attributes fulfill given predicates [8]. The observability features needed to support regulatory compliance need to balance complete monitoring with the protection of privacy by introducing differential privacy mechanisms that inject calibrated noise into aggregate metrics so that individual records cannot be identified, but statistical utility is retained for compliance reporting and anomaly detection [8]. Container security within such architectures necessitates privacy-sensitive logging systems that collect enough detail to enable security incident investigation and regulatory audit without logging sensitive personal data, applying data minimization practices where logs carry only pseudonymized identifiers and aggregated metrics instead of personal-level data [8].

Security Dimension	Cryptographic Mechanisms	Implementation in AI Infrastructure	Regulatory Compliance Support
Data Protection at Rest	Symmetric encryption, secure key storage, hierarchical key derivation	Encrypted dataset storage, protected model repositories, secure credential vaults	Satisfies data protection mandates, enables audit integrity
Data Protection in Transit	TLS encryption, mutual authentication, forward secrecy	Encrypted API communications, secure pipeline transfers, protected inference requests	Prevents data interception, ensures communication authenticity

Authentication Mechanisms	Digital signatures, multi-factor authentication, challenge-response	User identity verification, service authentication, API validation	Provides nonrepudiation, supports strong authentication requirements
	protocols		
Integrity Verification	Cryptographic hashing, Merkle trees, blockchain-inspired chains	Model versioning with tamperevident logs, dataset lineage tracking	Enables decision reconstruction, proves data integrity
Authorization and Access Control	Attribute-based policies, threshold cryptography, zeroknowledge proofs	Fine-grained data permissions, multi-party approvals, privacypreserving checks	Implements least privilege, enforces separation of duties
Privacy Preservation	Predicate encryption, differential privacy, pseudonymization	Protected analytics, compliant training, secure experimentation	Satisfies privacy regulations, enables compliant data sharing

Table 3. Cryptographic Security and Compliance Frameworks for Banking AI Systems [7, 8].

Organizational and Operational Considerations

Infrastructure transformation success hinges on technology teams, business stakeholders, and compliance functions working in harmony. Technology leaders will need to create phased modernization roadmaps that balance high-impact uses of AI with disciplined tackling of infrastructure debt, adopting DevOps practices that are key success factors such as cultural transformation where organizations move away from siloed functional groups to collaborative crossfunctional units, automation of manual operations via continuous integration and continuous deployment pipelines, improvement driven through metrics where comprehensive sets of metrics are collected and analyzed, and knowledge sharing mechanisms that eliminate information barriers between development and operations teams [9]. Developing skills is necessary as legacy operations teams need to cope with cloud-native technologies and DevOps processes, necessitating organizational practices that enable learning through focused training programs, test and development environments in which teams are able to test new technologies in a risk-free manner without endangering production systems, and communities of practice enabling knowledge transfer across organizational boundaries [9]. The cultural shift with the modernization of infrastructure is as important as technical developments, with studies finding effective adoption of DevOps relies significantly on the development of common purposes and shared responsibility with development and operations teams sharing responsibility for system dependability and business results in preference to optimizing for opposing local goals, adopting blameless postmortem processes aimed at systemic improvement over attributing faults to individual persons when things go wrong, and enhancing psychological safety where team members are able to speak up, acknowledge errors, and question default ways of doing things without fear of retribution [9].

Cross-functional teams make sure that investment in infrastructure serves business goals directly and not for its own sake. Periodic comparison of infrastructure capacity with AI project needs ensures early identification of bottlenecks that can sabotage initiatives, establishing feedback loops that are a key DevOps success driver in the sense that they ensure operational wisdom informs development agendas, production metrics influence infrastructure capacity planning, and business impact monitoring

confirms that technical enhancements translate to real value creation [9]. Change management procedures need to improve in order to accommodate the accelerated release cycles enabled by newer infrastructure without compromising the risk mitigation banking necessitates, embracing continuous delivery techniques recognized as critical DevOps capabilities that facilitate frequent, low-risk releases through end-to-end automated testing, incremental deployment approaches, and quick rollback processes that limit the business effect of failures [9]. The automation building blocks of contemporary infrastructure operations cater to key success factors such as tool standardization in which firms consolidate disparate toolchains into unified platforms that enable the entire software delivery pipeline, infrastructure as code methodologies that substitute manual configuration with versioned declarative statements for reproducible environment provisioning, and automated testing mechanisms for ensuring functionality, performance, and security attributes prior to production deployment [9]. Banking organizations that have instituted these practices note that DevOps adoption is highly correlated with enhanced deployment frequency, lower lead times from code development to production deployment, reduced mean time to recovery when there are incidents, and fewer change failures, with organizational practices such as executive sponsorship, dedicated transformation teams, and phased adoption approaches being more strongly predictive of success than technology selection [9].

The organizational designs facilitating infrastructure change are now increasingly recognizing that operations teams and system administrators do advanced cognitive work that involves deep technical knowledge, understanding of context, and refined problem-solving ability [10]. System administrators perform varied work such as routine monitoring routines where they continually evaluate system health by analyzing log files, performance data, and alert messages, troubleshooting work that requires systematic examination of unusual behaviors by hypothesis creation and testing, preventive maintenance operations that foresee possible failures and fix vulnerabilities before they affect users, and capacity planning activities that forecast future resource needs based on growth patterns and usage rates [10]. System administrators' information needs cover a range of dimensions such as realtime status data regarding current system conditions and active processes, historical trend data that displays patterns over long periods of time, comparative baselines that are able to differentiate usual variations from true anomalies warranting intervention, and contextual metadata regarding system configurations, dependencies, and recent changes which could explain what is observed [10]. System administrators depend a great deal on informal knowledge bases such as personal documentation, team wikis, and institutional knowledge resident in experienced practitioners and not on formal procedures since the nature of production systems is constantly changing, and thus written-down procedures are soon obsolete while tacit knowledge of system idiosyncrasies and failure modes is built up through direct experience of running systems [10]. The interruption-driven nature of operations work creates significant challenges as administrators must constantly context-switch between planned tasks and urgent incidents, maintaining mental models of multiple concurrent issues while responding to alerts that demand immediate attention, leading to fragmented work patterns where extended periods of focused effort on complex problems become rare [10]. Tool design for contemporary infrastructure management has to account for these realities of work practice by offering integrated perspectives that connect information between different system components, intelligent alerting that winnows and prioritizes truly important events, automation features that perform routine tasks in the background, and collaboration tools that facilitate knowledge sharing in geographically distributed teams working asynchronously on common infrastructure [10].

Transformation Area	Critical Success Factors	Implementation Practices	Organizational Impact

Cultural Evolution	Cross-functional collaboration, shared accountability, blameless postmortems, psychological safety	Joint ownership of systems, collective incident resolution, and transparent communication	Breaks information barriers, accelerates problem-solving, fosters innovation
Skills Development	Cloud-native proficiency, infrastructure-as-code, container orchestration, and distributed systems knowledge	Training programs, strategic recruitment, vendor partnerships, experimentation environments	Builds internal capabilities, reduces external dependency, and attracts modern talent
Automation Implementation	CI/CD pipelines, automated testing, infrastructure provisioning, and configuration management	Standardized toolchains, automated validation, version-controlled infrastructure	Increases deployment frequency, reduces lead times, minimizes errors
Measurement and Feedback	Comprehensive metrics, production telemetry, operational data analysis, and impact validation	Service level objectives, error budgets, continuous monitoring, and capability assessments	Enables evidence-based decisions, proactive issue identification, and value alignment
Leadership Commitment	Executive sponsorship, organizational support, sustained investment, resource allocation	Dedicated transformation teams, comprehensive budgeting, and incremental adoption	Overcomes resistance, sustains momentum, legitimizes new practices

Table 4. Organizational Transformation Factors for AI Infrastructure Modernization Success [9, 10].

Conclusion

Infrastructure modernization is not so much a technical upgrade program as a strategic necessity for banking organizations in pursuit of unlocking artificial intelligence's production-environmenttransforming potential. Legacy computing infrastructures pose essential obstacles to success with AI by way of limited computational resources, dispersed data access patterns, and architecturally entrenched incompatibilities with today's machine learning practices involving real-time processing and dynamic scalability. Legacy mainframe systems built for batch transactional processing are incapable of handling real-time data flows critical for dynamic fraud detection models or targeted recommendation systems reacting to customer interactions in milliseconds. Information siloed across disparate departmental systems does not allow for extensive feature engineering and global understanding of customers needed for effective predictive modeling. Infrastructure modernization programs must then holistically solve these underlying constraints through strategic deployment of cloud computing platforms offering on-demand provisioning of resources and near-infinite storage capacity, containerization technologies that enable uniform deployment and avoid conflicts between dependencies, integrated data architectures bringing together structured transactional data with unstructured communications and behavioral indicators, and security models embracing cryptographic defenses and privacy-preserving controls meeting rigorous regulatory requirements. Successful change requires more than technology procurement, with organizational development involving cross-functional cooperation among technology groups and business customers, training programs equipping

operations staff with cloud-native technology and DevOps practices skills, cultural transformation towards collaborative responsibility and continuous improvement, and leadership support maintaining multi-year efforts in the face of inevitable implementation difficulties. Financial institutions making systematic investments in infrastructure modernization establish enabling foundations for production-scale AI systems, yielding quantifiable business value through operational efficiency improvements, improved risk management capabilities, and better customer experiences. Banking organizations tackling infrastructure constraints intentionally can move beyond pilot project boundaries and realize enterprise-wide AI deployment, creating enduring competitive differentiation in quickly changing digital financial services markets.

References

[1] Sachin Kumar & Mikhail Zymbler, "A machine learning approach to analyze customer satisfaction from airline tweets," Springer Nature Link, 2019. [Online]. Available: https://link.springer.com/article/10.1186/s40537-019-0224-1

[2] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," ACM, 2016. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/2939672.2939785

[3] Marlon Dumas et al., "Introduction to Business Process Management," Springer Nature Link, 2018. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-56509-4_1

[4] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified data processing on large clusters," Communications of the ACM, 2008. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1327452.1327492

[5] Peter Mell and Timothy Grance, "The NIST Definition of Cloud Computing," NIST, 2011. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[6] DAVID BERNSTEIN, "Containers and Cloud: From LXC to Docker to Kubernetes," IEEE Explore, 2021. [Online]. Available: https://sweet.ua.pt/andre.zuquete/Aulas/AES/20-21/extras/Bernstein14.pdf

[7] Víctor Gayoso Martínez et al., "Analysis of the Cryptographic Tools for Blockchain and Bitcoin," MDPI, 2020. [Online]. Available: https://www.mdpi.com/2227-7390/8/1/131

[8] Pelin Angin et al., "An Entity-centric Approach for Privacy and Identity Management in Cloud Computing," ResearchGate. [Online]. Available: https://www.researchgate.net/profile/RohitRanchal/publication/224189965_An_Entity-Centric_Approach_for_Privacy_and_Identity_Management_in_Cloud_Computing/links/0912f50c64cafc64e1000000/An-Entity-Centric-Approach-for-Privacy-and-Identity-Management-in-CloudComputing.pdf

[9] Nasreen Azad, "Understanding DevOps critical success factors and organizational practices," IEEE/ACM International Workshop on Software-Intensive Business, 2022. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3524614.3528627

[10] Nicole F. Velasquez and Suzanne P. Weisband, "Work Practices of System Administrators: Implications for Tool Design," ResearchGate, 2008. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/51207249/Work_practices_of_system_administrators_20170105-10190-pdbic-libre.pdf?1483641817=&response-contentdisposition=inline%3B+filename%3DWork_practices_of_system_administrators.pdf&Expires=17616
38979&Signature=IpoSbfybOrSuxkGc-

Journal of Information Systems Engineering and Management
2026, 11(1s) e-ISSN: 2468-4376 https://www.jisem-journal.com/ **Research Article**

b2TodBBNmczYISAnZUjNj~FRrs1yjoHLM~GvgHfwMKBm3q73jxRA1WSaIjR8~gr21mRXRTCWkRdp15YTWunHiz6POFugM-hz0mnOPJHfiXWfE-ag0TQXp3hOg2nGDIl2x9lisLDqw1yLQUaFKiWcpUS0kht6F8KGFXGOTBGHBmJY~UF6MHUOk90ALak~e-HsJztZDL-ZRHFHzgbQVALw-UfrRz5zXfbup9jE20HCjFf6nafwGYV5cUIF42SfW6wxEj8PNH4B1L14U~aLn1pweb1F9BYqaYTFRw4n8TllM79QZqQD0Yu6i292o8gakvxmF1~Q__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA