International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Avoiding Engineering Failures in Healthcare Cloud Migrations: Ten Lessons from Platform Modernization at Scale

Sandipan Biswas

Director of Engineering; Fortune 20 Healthcare Company, Richmond, Virginia, USA
Email: *sandipan.ece[at]gmaill.com*

Abstract: ***Cloud migration in the healthcare industry involves high-stakes initiatives impacting not only technical infrastructure but crucially data governance, compliance, and patient care. While public cloud platforms offer significant scalability and flexibility, healthcare organizations frequently encounter systemic challenges during large-scale migrations, including regulatory misalignment, architectural inefficiencies, and operational gaps. Unlike previous literature that primarily focuses on general technical considerations, this paper provides novel insights drawn from hands-on leadership experience in large-scale healthcare data platform modernization efforts. Specifically, it identifies and analyzes ten recurring engineering pitfalls, emphasizing often-overlooked data management issues such as metadata cataloging, data governance, and compliance monitoring. Each identified mistake is supported by concrete, real-world examples, recent academic findings, and explicit practical mitigation strategies utilizing advanced data management tools (e.g., AWS Glue, Lake Formation, and Google Data Catalog). By highlighting these critical data-centric considerations, this work aims to guide engineering leaders, architects, and compliance officers toward building healthcare cloud-native systems that are compliant, cost-effective, resilient, and fully optimized for data reuse, transparency, and adaptability within evolving digital health ecosystems.***

Keywords: Healthcare Data Engineering, Cloud Migration, Data Platform Modernization, Data Governance, Cloud Cost Optimization

1. Introduction

Healthcare is undergoing a significant shift toward cloud-based infrastructure, driven by the need for enhanced scalability, interoperability, and real-time analytics. However, cloud adoption in this sector is more than just a technical upgrade: It represents a complex transformation directly tied to patient privacy, stringent regulatory compliance, and uninterrupted operational continuity.

Unlike other industries, healthcare organizations face a uniquely challenging environment due to the following:

- **Regulatory constraints**: Compliance with HIPAA, HITRUST, SOC2, and related standards is mandatory, significantly impacting all architectural and operational decisions.
- **Complex data ecosystems**: Healthcare data spans electronic health records (EHRs), claims processing, medical imaging, IoT-generated patient vitals, and provider directories, each adhering to different standards (HL7, FHIR, X12) and requiring specialized handling.
- **Persistent legacy systems**: Healthcare providers often rely on legacy systems designed decades ago, which are inherently unsuitable for modern, distributed, cloud native architectures.
- **Organizational inertia and stakeholder fragmentation**: Healthcare IT teams frequently navigate a landscape of diverse stakeholders—clinical, administrative, and legal—each bringing competing priorities, limited cloud expertise, and varied expectations.

These unique constraints significantly amplify the complexity and risks associated with healthcare cloud migrations. Despite this, many organizations mistakenly approach migrations as simple "lift-and-shift" initiatives, neglecting strategic considerations vital for sustainable, compliant, and scalable data infrastructure.

This paper fills an important gap in existing research by explicitly focusing on critical but often overlooked data-centric issues inherent in healthcare cloud migrations. Specifically, it identifies and explores ten recurring engineering pitfalls drawn from first-hand leadership experience in national-scale healthcare data platform modernizations. Unlike prior studies, this work emphasizes novel insights into advanced data management methodologies and governance tools (such as AWS Glue, Google Data Catalog, AWS Lake Formation), crucial for meeting healthcare-specific regulatory requirements and maximizing data re-usability, observability, and auditability.

The practical lessons and structured recommendations presented here will assist engineering leaders, architects, compliance officers, and healthcare executives in designing cloud-native solutions that are robust, compliant, and strategically optimized to adapt seamlessly within the rapidly evolving digital health ecosystem.

2. The Cost of Getting Cloud Migration Wrong in Healthcare

Cloud migration is often framed as a routine IT modernization step — a necessary move toward agility, elasticity, and cost efficiency. But in healthcare, the consequences of getting it wrong go far beyond missed deadlines or budget overruns.

Healthcare systems are deeply interwoven with real-world outcomes. Poorly executed cloud migrations can ripple

across the organization, disrupting not only operations but also patient care, legal standing, and public trust. Here is why avoiding engineering missteps in this space is mission-critical:

2.1 Patient Safety and Care Continuity

In 2020, a ransomware attack on Universal Health Services forced 400+ hospitals and care sites throughout the US to revert to paper processes for weeks, directly affecting diagnostics, scheduling, and medication administration [1]. While this was a security incident, it illustrates a broader truth: any disruption to healthcare IT — including from flawed migrations — risks compromising patient safety.

Imagine migrating a real-time eligibility check system to the cloud without appropriate failover or regional redundancy. If it goes down during peak hours, it delays authorizations and disrupts urgent procedures, particularly in EDs and surgical departments. These aren't just IT problems — they're care delivery failures.

2.2 Regulatory Exposure and Legal Liability

Healthcare is one of the most tightly regulated industries when it comes to data — with good reason. Protected Health Information (PHI) must meet strict standards for confidentiality, integrity, and availability under HIPAA, HITECH, and GDPR (where applicable).

Misconfiguring an S3 bucket, failing to encrypt data at rest, or deploying a pipeline without access control logging could lead to a data breach. According to IBM's 2023 Cost of a Data Breach Report, the average healthcare data breach cost is now $10.93 million, the highest across all industries [2].

In 2021, a major cloud-based telehealth platform was fined over $7 million for failing to secure PHI during a system migration — a direct consequence of improper IAM design and a lack of real-time auditability during the transition [3].

2.3 Financial Waste and Cost Overruns

Without planning for cloud-specific design patterns (e.g., autoscaling, event-driven triggers, storage tiering), many organizations "lift and shift" their legacy architecture — and carry all their inefficiencies with them. This results in exploding compute bills, data egress fees, and idle resources.

A payer organization I consulted with migrated over 100 TB of claims data into AWS but failed to implement intelligent lifecycle management. They racked up six-figure monthly storage bills before a tagging and cost control strategy was enforced. In another case, data warehouse jobs were re-hosted without parallel optimization — nightly jobs missed SLA windows due to under-provisioned compute, despite being on powerful cloud infrastructure.

2.4 Technical Debt and Team Frustration

A poorly executed migration often swaps old problems for new ones. Engineers face unfamiliar services, unclear dependencies, and insufficient documentation. Analysts lose trust in reporting when data is missing or inconsistent. Governance teams can't trace lineage or monitor data quality. And leadership loses patience when promised ROI doesn't materialize.

This friction snowballs into attrition, low morale, and platform stagnation — a common fate for cloud systems that weren't built with scale, governance, and usability in mind.

2.5 Strategic Setbacks and Lost Competitive Edge

In a value-based care world, cloud data infrastructure isn't just plumbing — it's strategic. Cloud-native capabilities are essential for powering real-time analytics, risk scoring, patient segmentation, and predictive modeling. Migrate poorly, and you don't just lose money — you lose the ability to innovate.

Organizations that fail to modernize effectively fall behind peers who are already running serverless ETL, streaming FHIR pipelines, and ML-powered care interventions.

3. The Mistakes

Cloud migration is often framed as a routine IT modernization step — a necessary move toward agility, elasticity, and cost efficiency. But in healthcare, the consequences of getting it wrong go far beyond missed deadlines or budget overruns.

Healthcare systems are deeply interwoven with real-world outcomes. Poorly executed cloud migrations can ripple across the organization, disrupting not only operations but also patient care, legal standing, and public trust. Here is why avoiding engineering missteps in this space is mission-critical:

3.1 Lifting and Shifting Legacy Systems Without Refactoring

One of the most common — and costly — mistakes in healthcare cloud migrations is the direct lift-and-shift of legacy systems into cloud infrastructure without architectural refactoring. "Successful cloud migration is not just 'lift and shift,' which alone will not deliver the strategic benefits achieved through cloud. The critical success factor is the organization's ability to utilize cloud migration as an opportunity to transform" [4]. While this approach may promise speed and simplicity, it almost always leads to poor performance, rising costs, and failure to realize the core benefits of the cloud.

In legacy on-premises environments, systems were often designed for vertical scaling, static capacity provisioning, and batch processing. These designs are incompatible with cloud-native capabilities such as serverless execution, auto-scaling, event-driven architectures, and elastic storage. Simply re-hosting these systems in virtual machines (e.g., EC2, Compute Engine) replicates inefficiencies and technical debt — and in regulated environments like healthcare, can also introduce compliance risks if monitoring, encryption, and access models aren't reconfigured appropriately.

In a healthcare data modernization effort involving hundreds of downstream applications, we encountered a recurring problem: legacy batch jobs designed for on-premises systems were moved directly to cloud-based virtual machines with no change in structure or logic. These jobs continued to run on fixed nightly schedules, used static resource allocation, and lacked any support for parallel execution. As a result, workloads consumed excessive compute time, caused frequent SLA breaches for data freshness, and failed to scale reliably during peak load times.

This experience is consistent with broader industry observations. Organizations that adopt lift-and-shift migrations without rearchitecting often find that their systems inherit the same inefficiencies that existed in their previous environments — only now with higher cloud bills and more fragile operations. These legacy systems struggle to leverage the elasticity, observability, and distributed processing capabilities of the cloud. Moreover, the failure to modularize monolithic workflows leads to poor fault isolation, making maintenance and debugging much harder in production environments.

3.1.1 Recommended Fix

To fully realize the benefits of cloud infrastructure — including elasticity, observability, resilience, and compliance — healthcare systems must move beyond "lift-and-shift" and embrace refactor-first strategies. This means rethinking legacy architecture not just as a technical migration, but as an opportunity to align data workflows with modern, cloud-native principles. Below are five actionable strategies for doing so effectively:

- Rather than lifting monolithic ETL pipelines into the cloud as-is, healthcare organizations should modularize them into discrete stages (e.g., ingestion, validation, transformation). This improves observability, error handling, and reusability while enabling orchestration through tools like Apache Airflow or AWS Step Functions.
- Replacing cron-based batch triggers with event-driven logic is another key improvement. Serverless frameworks such as AWS Lambda or Google Cloud Functions allow data workflows to execute in response to actual events — like a new file arrival or a patient record update — instead of fixed intervals, reducing cost and improving responsiveness.
- Cloud-native systems should be built for elasticity. Instead of provisioning static compute resources, teams can deploy auto-scaling clusters (e.g., EMR with spot instances or GKE-based Spark jobs) and decouple storage from compute using S3, BigQuery, or similar technologies, ensuring resilience and cost-efficiency.
- Legacy jobs often reload entire datasets each night, but cloud-native patterns favor real-time ingestion. By adopting change data capture (CDC) tools such as Debezium or AWS DMS and streaming platforms like Kafka or Dataflow, teams can build low-latency pipelines that continuously update dashboards, directories, and downstream systems.
- Finally, it is critical to benchmark before and after migration. Teams should compare SLA adherence, resource usage, and job performance using both historical production workloads and synthetic stress tests. This validation step ensures that refactoring translates into measurable gains — not just architectural elegance.

3.2 Ignoring Data Governance in Cloud Design

A critical mistake often observed in healthcare cloud migrations is the failure to embed robust data governance into the architecture from the outset. While cloud platforms offer sophisticated governance tooling, many organizations mistakenly treat these as post-deployment add-ons rather than foundational design principles. The result is fragmented access control, unclear data lineage, and an inability to consistently enforce compliance—all of which jeopardize trust, significantly increase operational risk, and hinder regulatory readiness.

In the highly regulated healthcare sector, where sensitive patient data flows across payers, providers, and regulators, poor governance can lead to serious and costly complications, including data inconsistencies, redundant data ingestion pipelines, unauthorized access to Protected Health Information (PHI), and failures during critical compliance audits. Without clear data tagging, defined ownership, and automated lineage tracking, it becomes nearly impossible to answer fundamental questions crucial for HIPAA and HITRUST compliance [5].

During the modernization of a multi-tenant provider data platform serving over 200 applications, our team encountered critical gaps caused by the absence of a comprehensive metadata strategy and clear data ownership definitions. Data engineers were frequently forced to manually reverse-engineer table provenance for compliance audits, often leading to conflicting interpretations and delayed responses. This experience is consistent with broader industry observations, where neglecting proper architectural and governance considerations can lead to higher operational costs and risks in healthcare cloud environments [6].

3.2.1 Recommended Fix

- Establish a formal data governance framework before any migration begins. This includes defining data ownership, classification levels, access roles, and stewardship responsibilities. Cloud-native tools such as AWS Lake Formation or Google Cloud Data Catalog should be leveraged to automate and enforce these policies across datasets.
- Implement metadata tagging and lineage capture from day one. Apply uniform tagging strategies across tables, storage buckets, and processing jobs to enable traceability and auditability. Use schema registries and pipeline metadata stores to track transformations, inputs, and outputs throughout the data lifecycle.
- Automate access control and audits through policy-as-code. Leverage role-based and attribute-based access models with services like AWS IAM, GCP IAM, or Azure RBAC. Ensure access requests, grants, and revocations are logged and reviewed periodically, ideally with integrated dashboards for compliance teams.

3.3 Underestimating Compliance Requirements in Cloud

Another frequent and dangerous mistake during healthcare cloud migrations is the assumption that cloud platforms are "secure by default" and automatically compliant with healthcare regulations like HIPAA, HITECH, or SOC 2. While public cloud providers offer robust security features, it is the responsibility of the healthcare organization to correctly configure and enforce them—a shared responsibility model that is often misunderstood or neglected.

Inadequate encryption, misconfigured storage buckets, overly permissive access controls, and insufficient logging are among the most common violations observed in healthcare cloud environments. These issues often emerge when legacy applications are ported to the cloud without reevaluating the security posture in the context of dynamic, distributed environments. In healthcare, such oversights can lead not only to breaches of protected health information (PHI) but also to multi-million-dollar regulatory penalties and lasting reputational damage.

In one enterprise migration effort involving claim data systems, our audit revealed that several Amazon S3 buckets used for staging data were left unencrypted and open to public internet access due to misapplied policies during initial setup. Although no breach occurred, this finding triggered an internal investigation and a full review of our cloud configuration practices. Such misconfigurations are a well-documented risk, with numerous instances of patient data exposure resulting from improperly secured cloud storage, highlighting the critical need for vigilant compliance and robust configuration management [7].

3.3.1 Recommended Fix

- Understand and implement the cloud provider's shared responsibility model. While AWS, Azure, and GCP secure the infrastructure layer, customers must configure encryption, access, and auditing at the application and data levels. Compliance checklists from cloud vendors should be used as mandatory baselines during architecture design.
- Enforce encryption for all data at rest and in transit. Services like AWS KMS or Google Cloud Key Management should be used to manage keys centrally. Ensure storage buckets and data lakes are encrypted by default, and APIs use TLS for transmission.
- Apply least privilege and role-based access control. Avoid default or overly broad permissions. Tools such as AWS IAM Access Analyzer or GCP Policy Analyzer can detect unnecessary access paths. Regularly audit user roles, service accounts, and permission grants.
- Implement centralized logging and alerting for all sensitive operations. Use services like AWS CloudTrail or Google Cloud Audit Logs to track access to PHI-related resources. Integrate with SIEM platforms to flag anomalies and meet HIPAA audit trail requirements.

3.4 Poor Identity and Access Management (IAM)

Cloud environments offer powerful mechanisms for managing user identities and securing resource access. However, when poorly configured, they transform into one of the most significant attack surfaces in healthcare IT. A prevalent mistake during migration is to either replicate the same monolithic role structures from legacy on-premises environments or to over-provision access in the name of expediency. Both approaches fundamentally violate the principles of least privilege and zero trust, creating critical vulnerabilities that are easily exploited by malicious actors [8].

In the healthcare sector, where Protected Health Information (PHI) is accessed by diverse teams—including engineering, data science, and operations—IAM missteps can escalate rapidly. Overly permissive service accounts, hard-coded credentials embedded in applications, and unmanaged role sprawl lead to dangerous lateral movement opportunities in the event of a breach. These inherent risks are further magnified in multi-tenant or federated systems where responsibility for defining and enforcing IAM boundaries is distributed across various organizational units [9].

During a recent audit of a federated payer system with multiple upstream data contributors, our team discovered that over 60% of IAM roles had never undergone review since their initial assignment. These roles included broad, wildcard permissions to highly sensitive objects such as provider directories and claims extracts. Furthermore, many users retained excessive access privileges even after their roles within the organization had transitioned. Such oversights are a well-documented concern in cloud security, particularly within healthcare, where inadequate IAM hygiene remains a leading cause of data exposures and regulatory non-compliance [8], [9].

3.4.1 Recommended Fix

Design IAM policies using the principle of least privilege. Break down access by function, data classification, and sensitivity. Instead of broad administrative roles, define narrowly scoped roles (e.g., read-only analytics, write-access ingestion) and assign them via group-based policies.

Regularly audit and rotate credentials, tokens, and role assignments. Implement identity lifecycle management workflows that de-provision access when users change roles or exit the organization. Enable MFA for all accounts with access to PHI or cloud administration.

Use federated identity and single sign-on (SSO) to centralize access management. Leverage tools like AWS SSO, Azure AD, or GCP Identity Federation to reduce reliance on manual IAM rule assignments and hard-coded credentials in pipelines or applications.

Continuously monitor IAM policies for drift or anomalies. Use services like AWS IAM Access Analyzer, Google Policy Analyzer, or third-party tools (e.g., Wiz, Lacework) to identify over-privileged users and detect violations of your intended access control strategy.

3.5 Inadequate Observability and Monitoring Setup

A major oversight in many healthcare cloud migrations is the lack of investment in observability. While data movement and infrastructure provisioning often take center stage, logging, tracing, and monitoring are treated as afterthoughts. This leaves systems vulnerable to undetected failures, data quality degradation, SLA breaches, and potential mishandling of protected health information (PHI).

Legacy systems typically rely on rigid, centralized logging tools that are not equipped for today's distributed, cloud-native environments. In contrast, cloud platforms require real-time, context-rich telemetry across multiple layers — from APIs to storage to job orchestration — to enable timely diagnosis and response.

For example, in a claims enrichment pipeline using managed Spark clusters, our team struggled to trace intermittent data loss due to insufficient logging granularity and no unified tracing. Once we introduced OpenTelemetry for end-to-end traces and Prometheus for metrics, we isolated a retry timeout issue in a connector. This not only improved reliability but also accelerated root-cause analysis and boosted confidence in operational readiness.

Without a robust observability strategy, even well-architected systems can become unmanageable, especially under regulatory pressure or during audit situations. Cloud systems demand visibility by design — not as a post-migration patch [10].

3.5.1 Recommended Fix

- Deploy centralized observability platforms early in the migration lifecycle. Solutions such as AWS CloudWatch, Google Cloud Operations Suite (formerly Stackdriver), or third-party tools like Datadog and New Relic should be used to collect logs, metrics, and traces across all layers — storage, compute, APIs, and orchestration.
- Instrument pipelines with structured, context-aware logging. Ensure all critical stages in the data lifecycle emit logs with trace IDs, error codes, and metadata tags. Correlate logs with monitoring dashboards to track job status, performance metrics, and system health in real time.
- Enable real-time alerting and anomaly detection. Use rule-based alerts for critical failures (e.g., job crashes, PHI access spikes) and ML-based anomaly detection for identifying unexpected behavior over time. Integrate alerts with incident management tools like PagerDuty or Opsgenie.
- Maintain audit-grade telemetry. Ensure logs are immutable, timestamped, and retained according to HIPAA and SOC 2 guidelines. Regularly test traceability for access, transformation, and deletion events involving PHI.

3.6 Skipping Cost Optimization Planning

A common but dangerous assumption in healthcare cloud migration is that cost savings will occur automatically once infrastructure moves off-premises. Without proactive cost management strategies, organizations often find themselves facing higher operational expenses than before. This is especially true in healthcare, where large volumes of structured and unstructured data are constantly ingested, stored, processed, and retained for compliance.

The cloud's flexibility enables rapid experimentation and scaling, but it also introduces the risk of unchecked sprawl — unused compute nodes, idle services, unmonitored staging environments, and redundant data copies can silently inflate bills. Because cloud pricing is usage-based, rather than fixed, it requires continuous visibility, tagging, and governance to prevent runaway costs. Without deliberate cost optimization, healthcare organizations risk burning resources that could otherwise be allocated toward innovation — such as clinical AI tools, patient-facing applications, or analytics platforms that drive operational value. Planning for efficiency must be built into the architecture from the start, not left as an afterthought [11].

3.6.1 Recommended Fix

- Embed FinOps practices into your migration strategy. Use cloud-native tools like AWS Cost Explorer, GCP Billing, or Azure Cost Management to monitor usage, set budgets, and identify high-cost services. Establish cost accountability across engineering, finance, and product teams from day one.
- Enforce tagging policies and resource ownership. Apply consistent tags for environment, application, data sensitivity, and cost center to all resources. Use tagging-based automation to terminate idle jobs, archive cold data, and trigger alerts for overages.
- Optimize compute and storage configurations. Use autoscaling, serverless, or spot instances for variable workloads. Apply intelligent storage tiering with lifecycle rules that move unused data to infrequent access or archival tiers after predefined thresholds.
- Run periodic cost audits and cleanup campaigns. Schedule reviews of idle resources, redundant datasets, and oversized VM instances. Align cost optimization with compliance requirements to avoid retention violations while maximizing savings.

3.7 Not Planning for Multi-Region or Disaster Recovery Early

A critical but often overlooked element in healthcare cloud migration is early planning for disaster recovery (DR) and multi-region architecture. Legacy systems typically relied on periodic backups or secondary data centers for redundancy — methods that are insufficient in the dynamic, distributed nature of cloud-native infrastructure.

In healthcare, the cost of downtime extends beyond financial loss. Service interruptions in systems supporting Electronic Health Records (EHRs), claims processing, or provider networks can directly compromise patient care and violate regulatory expectations. Despite this, many organizations defer DR planning until late in the deployment process, treating it as an operational concern rather than a core architectural responsibility, a common challenge in cloud disaster recovery planning [12].

In one real-world example, a regional cloud service disruption rendered several health plan APIs inaccessible for hours due to the lack of cross-region failover and automated restoration. The incident forced clinical and administrative systems into manual fallback modes. Only after the outage were measures such as DNS-based routing, storage replication, and infrastructure-as-code recovery implemented. This case illustrates the importance of building high availability and resilience into the initial system design — not as a response to failure, but as a safeguard against it.

3.7.1 Recommended Fix

- Integrate disaster recovery planning into the core architecture. Use multi-region and multi-zone deployments for critical services, and ensure that databases, file stores, and message queues support cross-region replication and failover readiness.
- Automate infrastructure recovery using infrastructure-as-code (IaC). Tools like Terraform or AWS CloudFormation should be used to define and rapidly redeploy environments in a secondary region if primary systems become unavailable.
- Regularly test failover and recovery processes. Conduct controlled DR drills in staging environments to validate that RTO (Recovery Time Objective) and RPO (Recovery Point Objective) targets are being met. Include observability checks and rollback mechanisms in these rehearsals.
- Balance redundancy with cost by classifying workloads. Not all components require active-active designs. Segment services by criticality (e.g., clinical APIs vs. batch analytics) and assign appropriate availability strategies to optimize spend without compromising compliance.

3.8 Overengineering Data Lake Architectures

In their pursuit of modernization, many healthcare organizations overengineer cloud-based data lakes—introducing excessive complexity without proportional benefit. While modular, layered designs are encouraged, overuse of architectural patterns and toolchains often results in bloated systems that are costly, slow, and difficult to evolve [13].

This complexity usually arises when teams try to apply every best practice simultaneously, rather than designing with clear use cases and data consumers in mind. Redundant ingestion zones, overlapping metadata catalogs, or multiple transformation engines are frequently implemented before stakeholder needs or access patterns are well understood. In healthcare, where accuracy and transparency are paramount, such designs can severely impair usability. When pipelines are opaque and data lineage unclear, trust in the platform erodes—affecting clinical insights, compliance efforts, and operational decisions.

In one provider data consolidation initiative, the initial architecture spanned three ingestion layers, dual metadata catalogs, and four separate compute frameworks. This setup created ongoing maintenance burdens and version mismatches between layers. Eventually, delivery timelines suffered, and engineering teams were diverted into managing architecture instead of delivering insights. The experience underscored a common lesson in cloud data engineering: simpler systems, aligned with real stakeholder needs, deliver more value than technically elaborate ones.

3.8.1 Recommended Fix

- Start with a minimal viable data lake. Focus on ingesting high-value datasets and supporting critical use cases before scaling horizontally. Avoid implementing all layers (bronze, silver, gold) upfront unless each has a defined role and measurable benefit.
- Reduce tool sprawl and standardize key components. Choose one metadata catalog, one orchestration engine, and one transformation framework unless a multi-tool strategy is justified by clear workload segmentation. Simplicity enables better observability, training, and governance.
- Define access and retention policies early. Ensure that data lifecycle rules, row-level permissions, and audit logging are configured during initial development to avoid costly retrofitting later. Align this design with your security and compliance frameworks.
- Iterate based on user feedback. Monitor how analysts, clinicians, and engineers interact with the data lake, and prioritize architectural evolution based on real pain points—whether it's schema evolution, access speed, or lineage traceability.

3.9 Lack of Data Cataloging and Metadata Strategy

A frequently overlooked component in healthcare cloud migrations is the implementation of a robust data cataloging and metadata management strategy. Without it, organizations quickly lose visibility into what data exists, where it resides, who owns it, and how it can be used—undermining the very purpose of modernizing the data platform.

In healthcare, this is more than a usability issue. Transparency, auditability, and traceability are critical for maintaining compliance and operational integrity. When cataloging is skipped or deferred, data lakes often degrade into opaque swamps: datasets are duplicated or misused, analysts reprocess similar data unknowingly, and compliance teams are left uncertain about where PHI is stored or how long it’s retained. These challenges underscore the critical role of data governance in healthcare information systems, which aims to address data problems and ensure data is treated as a valuable asset [14].

In one clinical quality analytics platform migration, the absence of standardized dataset documentation led to frequent inconsistencies in dashboards. Different teams used similarly named tables that represented different logic or time frames, resulting in mismatched reports, delayed releases, and eroded stakeholder trust. These challenges underscore the importance of treating metadata not as a nice-to-have, but as a foundational layer of any data-driven healthcare platform.

3.9.1 Recommended Fix

- Adopt a centralized metadata catalog from the beginning. Tools like AWS Glue Data Catalog, Google Cloud Data Catalog, or Apache Atlas allow you to automatically

register datasets, schemas, and ownership metadata. Enforce mandatory documentation and tagging at the time of ingestion.
- Standardize dataset naming conventions, classifications, and stewardship assignments. Use taxonomy models tailored to healthcare (e.g., FHIR resource types, claims dimensions, patient risk categories) to ensure metadata aligns with both technical and business use.
- Integrate the catalog into data workflows. Enable search, discovery, and impact analysis features directly within the tools used by analysts, engineers, and compliance officers. Ensure that catalog changes are version-controlled and traceable.
- Track lineage from source to consumption. Use pipeline metadata or observability tools to build automated lineage graphs. This transparency supports debugging, SLA tracing, and HIPAA-mandated data tracking in case of breaches or subject access requests.

3.10 Treating the Migration as a One-Time Project

A fundamental error in many healthcare cloud efforts is viewing migration as a one-time project rather than a continuous transformation. Once data and systems are lifted into the cloud, teams often disband or reassign resources, leaving optimization, governance tuning, and user-driven improvements neglected. But in a sector shaped by evolving regulations, emerging technologies, and dynamic clinical workflows, static systems quickly fall out of alignment, underscoring the need for continuous data governance best practices throughout cloud migration projects and beyond [15].

When migration is treated as a finish line, technical debt accumulates, data pipelines drift from business needs, and governance frameworks become outdated. Without post-migration investment, organizations struggle to adapt to new standards, audit requirements, or analytical demands. Modern healthcare data ecosystems require constant refinement to remain compliant, performant, and user centric.

In one cloud-based provider data migration, the initial deployment was considered complete once ingestion and access pipelines were operational. However, within a year, accuracy issues surfaced due to unmonitored schema changes, delayed credential updates, and misaligned analytics outputs. A formal post-go-live team had to be retrofitted after service issues emerged during open enrollment. This experience highlights the need to treat migration as a phase in a broader digital transformation, supported by sustained monitoring, stakeholder feedback, and governance evolution.

3.10.1 Recommended Fix

- Establish a long-term migration operations team. Post-migration teams should include engineers, analysts, and compliance partners responsible for monitoring, validating, and evolving the system. Success should be measured not by go-live milestones but by sustained platform adoption and reliability.
- Build feedback loops into system governance. Regularly engage clinical, operational, and regulatory stakeholders to identify evolving data needs, access issues, or pipeline gaps. Use this input to prioritize enhancements and maintain alignment with organizational goals.
- Implement ongoing audit and improvement cycles. Use observability tools and metadata audits to continuously check system integrity, cost efficiency, and security posture. Schedule periodic retrospectives to assess whether the architecture is adapting to new data sources, user demands, and regulatory updates.
- Allocate budget and roadmap space for refactoring. Recognize that cloud architecture is not “set and forget.” Plan for regular codebase cleanups, performance tuning, and deprecation of legacy patterns to ensure your system remains agile and compliant over time.

4. Conclusion

Cloud migration in healthcare is more than a technical upgrade—it is a transformation of how data is governed, how systems are operated, and how care is delivered. Yet too often, migrations are rushed, narrowly scoped, or treated as isolated infrastructure changes rather than ecosystem-wide shifts. The ten mistakes explored in this paper reflect recurring engineering pitfalls that have deep operational, financial, and regulatory consequences in healthcare settings.

Taken individually, these mistakes—such as lifting and shifting legacy architectures, ignoring metadata strategies, or neglecting disaster recovery—may appear solvable. But when they co-occur, which is often the case in large-scale cloud programs, their effects become systemic. Data becomes harder to trace, costs become unpredictable, platform confidence erodes, and regulatory compliance is placed at risk. Healthcare’s unique requirements for interoperability, auditability, and patient privacy make it especially vulnerable to poor cloud design decisions [7][5].

Moreover, the complexity of healthcare data—ranging from EHRs and claims to IoT streams and unstructured notes—requires architectures that are not only scalable, but also transparent and adaptable. The shift to value-based care and AI-driven personalization further amplifies the need for trusted, well-governed data infrastructure. As research by Zhang et al. (2023) emphasizes, organizations that invest in continuous post-migration improvement, metadata fidelity, and cloud-native governance report significantly higher satisfaction and performance outcomes [13].

To build resilient, future-proof cloud systems, healthcare IT leaders must reframe migration as a long-term, iterative journey. This requires sustained investment in cross-functional teams, proactive governance, modern architectural thinking, and ongoing feedback from real users—not just compliance officers, but also analysts, clinicians, and engineers. Mistake avoidance alone is not enough; excellence in healthcare cloud engineering depends on intentional design, relentless iteration, and an unwavering focus on enabling safe, efficient, and meaningful data use.

As more organizations undertake this transformation, sharing lessons learned—especially those grounded in real-world

engineering practice—becomes essential. By documenting these common mistakes and their practical remedies, this paper aims to serve not only as a warning, but as a guide for building cloud systems worthy of healthcare's mission.

5. Abbreviations

The following abbreviations are used in this manuscript:

- PHI Protected Health Information
- HIPAA Health Insurance Portability and Accountability Act
- HITRUST Health Information Trust Alliance
- EHR Electronic Health Record
- ETL Extract, Transform, Load
- IAM Identity and Access Management
- DR Disaster Recovery
- SLA Service Level Agreement
- CDC Change Data Capture
- SSO Single Sign-On
- MFA Multi-Factor Authentication
- IaC Infrastructure as Code
- SIEM Security Information and Event Management
- VM Virtual Machine
- GCP Google Cloud Platform
- AWS Amazon Web Services
- RBAC Role-Based Access Control
- SSOT Single Source of Truth
- FHIR Fast Healthcare Interoperability Resources
- NIST National Institute of Standards and Technology.

References

[1] Jercich, K. (2021, March 5). *Universal Health Services faces $67 million loss after cyberattack*. Healthcare IT News. https://www.healthcareitnews.com/news/universal-health-services-faces-67-million-loss-after-cyberattack.

[2] IBM. (2024). *Cost of a Data Breach Report 2024*. https://www.ibm.com/reports/data-breach.

[3] U.S. Department of Health and Human Services. (2021). *Resolution agreements*. https://www.hhs.gov/hipaa/for-professionals/compliance-enforcement/agreements/index.html

[4] Black, M., & Veroff, D. (2022). *Unlocking enterprise innovation in the cloud: Strategy and blueprinting for health care organizations*. Deloitte Consulting LLP.

[5] N. K. M. Pulikonda, "Real-Time Clinical Data Governance Architecture: Financial Compliance-Inspired Model for HIPAA/HITECH Compliance," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 4, pp. 712–719, 2025.

[6] F. Sadoughi, S. Ahmadzadegan, and N. Bahri, "How the Health Information Systems Can Overcome the Challenges of Migrating to the Cloud? A Framework Based on a Mix Method Approach," *Front. Health Informatics*, vol. 11, no. 1, p. 107, 2022.

[7] Sadoughi F, El-Gazzar RF, Erfannia L, Sheikhtaheri A. How the health information systems can overcome the challenges of migrating to the cloud? A framework based on a mix method approach. *Front Health Inform.* 2022; 11: 73.

[8] M. Zhou, R. Zhang, W. Xie, W. Qian and A. Zhou, "Security and Privacy in Cloud Computing: A Survey," 2010 Sixth International Conference on Semantics, Knowledge and Grids, Beijing, China, 2010, pp. 105-112, doi: 10.1109/SKG.2010.19.

[9] Mehrtak, M., SeyedAlinaghi, S., MohsseniPour, M., Noori, T., Karimi, A., Shamsabadi, A., Heydari, M., Barzegary, A., Mirzapour, P., Soleymanzadeh, M., Vahedi, F., Mehraeen, E., & Dadras, O. (2021). Security challenges and solutions using healthcare cloud computing. Journal of medicine and life, 14(4), 448–461. https://doi.org/10.25122/jml-2021-0100.

[10] Mehrtak, M., SeyedAlinaghi, S., MohsseniPour, M., Noori, T., Karimi, A., Shamsabadi, A., Heydari, M., Barzegary, A., Mirzapour, P., Soleymanzadeh, M., Vahedi, F., Mehraeen, E., & Dadras, O. (2021). Security challenges and solutions using healthcare cloud computing. Journal of medicine and life, 14(4), 448–461. https://doi.org/10.25122/jml-2021-0100.

[11] Deochake, S. (2023). Cloud cost optimization: A comprehensive review of strategies and case studies. arXiv preprint arXiv:2307.12479.

[12] A. A. Tamimi, R. Dawood and L. Sadaqa, "Disaster Recovery Techniques in Cloud Computing," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019, pp. 845-850, doi: 10.1109/JEEIT.2019.8717450.

[13] Azzabi, S., Alfughi, Z., & Ouda, A. (2024). Data Lakes: A Survey of Concepts and Architectures. Computers, 13(7), 183. https://doi.org/10.3390/computers13070183

[14] Ngesimani, N. L., Ruhode, E., & Harpur, P. A. (2022). Data governance in healthcare information systems: A systematic literature review. South African Journal of Information Management, 24(1), 1475.

[15] Singh, K., & Singh, A. (2024). Data Governance Best Practices in Cloud Migration Projects. International Journal of Research Radicals in Multidisciplinary Fields, ISSN, 821-836.

Author Profile

Sandipan Biswas received his B.Tech. degree in Electronics and Communication Engineering from West Bengal University of Technology and later earned his M.B.A. in Data Analytics from the University of Southern Indiana. He is currently pursuing an M.S. in Computer Science from the Georgia Institute of Technology. Over the past 18+ years, he has worked extensively in enterprise-scale data and cloud engineering, with a specialized focus on modernizing healthcare IT infrastructure. He currently serves as Director of Engineering at a Fortune 20 healthcare company, where he leads national cloud migration programs, including the design and execution of enterprise data lakes and secure big data platforms on AWS and GCP. His research interests include cloud-native architecture, data governance, healthcare interoperability, and scalable distributed systems.