© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723551

Zero-Downtime Migration of Legacy Enterprise Systems to Cloud Infrastructure: A Framework for Saudi Arabia's Digital Transformation

NAGA SRAWAN KUMAR PABBA
0009-0006-6030-4825

Abstract- ***Cloud migration is often presented as an infrastructure relocation problem, yet enterprise continuity depends on a more difficult requirement: moving intertwined applications, databases, interfaces, identities, and operating practices without a user-visible service interruption. This review develops an evidence-informed framework for zero-downtime migration of legacy enterprise systems to cloud infrastructure, with specific attention to Saudi Arabia's digital transformation context. An integrative review synthesizes peer-reviewed research on legacy modernization, application portfolio assessment, microservice decomposition, data migration, progressive delivery, infrastructure automation, cybersecurity, and Saudi cloud adoption. The evidence indicates that zero downtime is not produced by a single architecture pattern. It emerges from coordinated controls: dependency discovery, workload characterization, parallel environments, continuous data replication, backward-compatible schema evolution, progressive traffic shifting, observable service-level objectives, transactional reconciliation, and rapid rollback. The review also finds that indiscriminate microservice decomposition can increase migration risk when domain boundaries, data ownership, and operational capabilities are immature. Accordingly, the proposed framework separates continuity controls from modernization depth and uses stage gates to decide when rehosting, replatforming, containerization, or deeper refactoring is justified. For Saudi organizations, the framework adds governance emphasis on trust, cybersecurity, organizational readiness, service continuity, and locally appropriate cloud operating capabilities. The paper contributes a structured synthesis that links technical migration mechanisms with managerial assurance and identifies research priorities for empirical validation in Saudi enterprise and public-service environments.***

Keywords: ***legacy systems; cloud migration; zero downtime; Saudi Arabia; digital transformation; progressive delivery; change data capture; enterprise modernization***

I. INTRODUCTION

Cloud infrastructure has become a central platform for enterprise digital transformation because it can provide elastic capacity, rapid provisioning, managed services, and automation that are difficult to reproduce consistently in aging on-premise estates. However, the migration of a mature enterprise system is not equivalent to deploying a new cloud-native application. Legacy portfolios contain long-lived databases, undocumented dependencies, brittle interfaces, scheduled jobs, shared file stores, embedded security assumptions, and business processes that may have accumulated over decades. Ellison et al. showed that workload and database structure materially influence migration duration, cost, and deployment choices, illustrating why migration planning must begin with evidence about the source system rather than generic cloud sizing [1]. More recent portfolio research similarly argues that organizations need application-level profiling and assessment before selecting cloud adoption paths [2]. Architectural reviews add that legacy-to-cloud migration is constrained by quality attributes, not simply by the target hosting model [3].

The continuity problem becomes sharper when a system cannot be taken offline for a maintenance window. In this paper, zero downtime means that the migration creates no externally observable interruption beyond the service's agreed availability and performance objectives. This definition does not imply zero risk, zero defects, or instantaneous cutover. Instead, it treats continuity as a verifiable engineering objective supported by redundancy, reversible change, data convergence, and controlled traffic movement. The distinction matters because a technically successful data copy can still cause business failure if transactions are lost, interfaces break, authentication

changes unexpectedly, or performance degrades after traffic is redirected. Security modernization is equally relevant: evidence from large public-sector modernization programs shows that modern infrastructure and migration choices can influence organizational cyber risk [4].

Modernization literature frequently recommends microservices as a destination for monolithic systems, but the evidence is more conditional than popular cloud narratives suggest. Assessment frameworks emphasize architectural fitness and decomposition readiness [5], while empirical work on technical debt shows that service extraction may redistribute rather than automatically remove structural problems [6]. Roadmaps for microservice modernization therefore stress staged decisions, domain understanding, and explicit migration activities [7]. Empirical studies of real transitions further demonstrate that systemic and technical migration interact: code structure, teams, operations, and deployment practices evolve together rather than in isolation [8]. Systematic reviews of monolith decomposition report growing automation support, but also persistent difficulties in defining service boundaries and evaluating extracted architectures [9,10]. These findings support a central premise of this review: zero-downtime migration should be designed as a continuity program first and an architectural transformation second.

Saudi Arabia provides a consequential setting for this problem. National organizations are expanding digital services while enterprises are increasing cloud adoption, yet Saudi studies continue to identify trust, organizational readiness, security concerns, and adoption conditions as important determinants of cloud decisions [23-25]. Research on digital transformation and cybersecurity in the Saudi context also highlights the interdependence between technical change, resilience, and security governance [26,27]. Consequently, a migration framework for Saudi organizations must do more than describe data transfer or target architecture. It must connect service continuity, application modernization, cyber controls, operating capability, accountability, and evidence-based cutover decisions.

This review addresses that need by integrating software-engineering and cloud-adoption research into a single operational framework. It uses the attached study only as a structural reference for a staged, evidence-led treatment of migration decisions; that study illustrates the value of modelling source workloads, comparing migration options, and validating assumptions against real migration behavior [1]. The present paper extends the problem from database migration to whole enterprise systems and from cost-duration estimation to continuity assurance. Its contribution is therefore conceptual and integrative: it identifies the control mechanisms that make interruption avoidable, explains where migration strategies differ, and organizes those mechanisms into stage gates that can be adapted to Saudi public and private organizations.

II. AIM OF THE STUDY

The aim is to develop a rigorous, evidence-informed framework for migrating legacy enterprise systems to cloud infrastructure without user-visible service interruption, while preserving data integrity, security, reversibility, and operational accountability. The study also aims to contextualize migration assurance for Saudi Arabia by linking technical continuity controls with organizational readiness, cybersecurity, trust, and governance factors identified in Saudi cloud and digital-transformation research [23-27].

III. RESEARCH OBJECTIVES

The study has five objectives. First, it synthesizes the principal technical failure modes that threaten continuity during legacy-to-cloud migration. Second, it compares modernization options, including rehosting, replatforming, containerization, and microservice-oriented refactoring, in relation to cutover risk. Third, it identifies evidence-based controls for data synchronization, progressive delivery, observability, and rollback. Fourth, it integrates those controls into a staged zero-downtime migration framework. Fifth, it examines how Saudi organizational, cybersecurity, and cloud-adoption conditions should shape governance and implementation of that framework.

IV. RESEARCH QUESTIONS

The review addresses four questions. RQ1 asks which architectural, data, deployment, and operational mechanisms are most consistently associated with continuity during legacy-to-cloud migration. RQ2 asks how organizations should choose the depth and sequence of modernization without increasing cutover risk. RQ3 asks how service-level evidence, transactional reconciliation, and rollback should govern progressive traffic transfer. RQ4 asks which Saudi-specific organizational and cybersecurity considerations require adaptation of a general zero-downtime migration approach.

V. REVIEW METHODOLOGY

An integrative review design was selected because the research problem spans software architecture, cloud engineering, deployment operations, information systems, and national adoption context. A narrowly defined meta-analysis would be inappropriate: studies use heterogeneous outcomes such as architectural quality, migration effort, decomposition accuracy, adoption intention, deployment reliability, security risk, and organizational readiness. The review instead follows a structured evidence-synthesis process that preserves methodological transparency while allowing comparison across these complementary research traditions.

The principal discovery sources were Scopus, Web of Science, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink. Searches prioritized work published from 2020 through 2025, with older studies retained only when they supplied foundational migration evidence directly relevant to the review, as in the workload-based migration study used as the structural reference [1]. Search strings combined terms in four clusters: legacy system or monolith; cloud migration, modernization, rehosting, replatforming, containerization, or microservices; zero downtime, high availability, blue-green, canary, continuous deployment, change data capture, replication, rollback, or schema evolution; and Saudi Arabia, cloud adoption, digital transformation, cybersecurity, trust, or organizational readiness. Backward and forward citation tracing were used to locate studies that connected otherwise separate themes, especially application decomposition and progressive delivery.

Eligibility required peer-reviewed journal or conference research with a clear relationship to enterprise application migration, cloud modernization, continuity-enabling deployment practices, infrastructure automation, cybersecurity, or Saudi cloud adoption. Studies were excluded when they were vendor marketing material, unreviewed blog guidance, purely educational demonstrations without enterprise relevance, or papers in which cloud terminology was incidental to the research question. Architectural studies were retained when they provided empirical data, systematic synthesis, explicit methods, or well-defined assessment frameworks. Saudi-context papers were included when they examined cloud adoption, trust, cybersecurity, digital transformation, or implementation conditions sufficiently close to enterprise migration governance. The final synthesis was restricted to the thirty peer-reviewed works listed in the References; this is a bibliographic boundary, not a fabricated screening statistic.

Screening was conducted conceptually in three passes: title and abstract relevance, full-text or authoritative metadata assessment, and thematic fit. No numerical flow diagram is reported because a frozen exported search corpus and auditable deduplication log were not created for this commissioned review; inventing identification, exclusion, or screening totals would therefore misrepresent the process. This limitation is made explicit rather than masked by pseudo-precision. Bibliographic details and DOI identifiers were checked against publisher pages, DOI metadata, or authoritative indexing records before inclusion.

Quality assessment focused on six criteria: clarity of research question, transparency of method, description of technical or organizational context, credibility of evidence, acknowledgment of limitations, and direct relevance to migration assurance. Systematic reviews and empirical enterprise studies were treated as stronger synthesis anchors; design proposals without substantial evaluation were used more cautiously to explain mechanisms. This approach avoids converting heterogeneous studies into an artificial numerical quality score while still distinguishing evidentiary

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723551

strength. The coding scheme captured source-system discovery, modernization strategy, service decomposition, data movement, deployment control, infrastructure automation, observability, rollback, cybersecurity, organizational readiness, and Saudi contextual factors.

Synthesis proceeded through constant comparison. Findings were first coded within the four continuity domains of architecture, data, deployment/operations, and governance. Mechanisms were then compared across studies to identify convergence, disagreement, and boundary conditions. For example, microservice research was examined not only for benefits but also for decomposition uncertainty, coordination complexity, and technical-debt effects [5-10]. Deployment evidence was then connected to data and infrastructure evidence rather than treated as an isolated DevOps concern [11,20-22]. Finally, Saudi adoption and cybersecurity studies were used to contextualize governance without claiming that a general software-engineering result automatically represents Saudi institutions [23-30]. Figure 1 summarizes this evidence-to-framework logic.

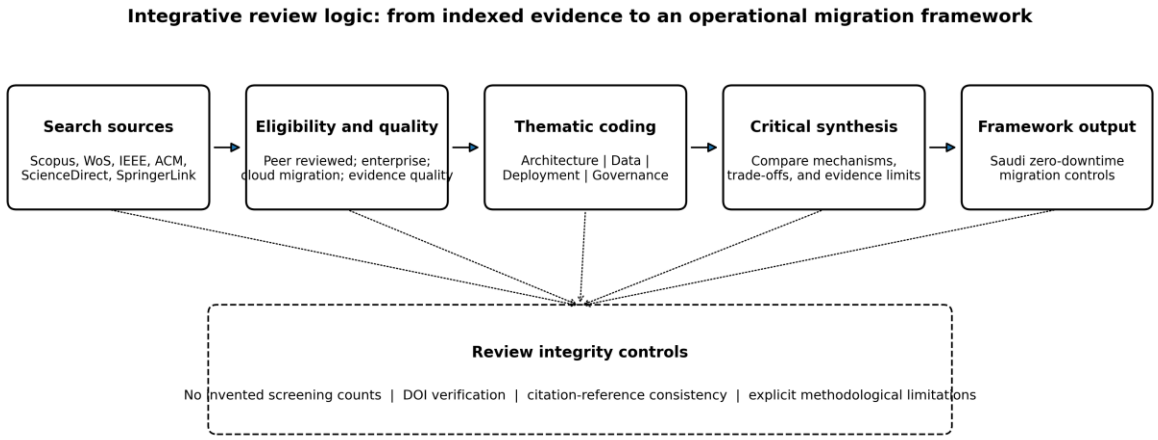

Figure 1. Integrative review logic and evidence-to-framework synthesis.

VI. THEMATIC LITERATURE REVIEW AND CRITICAL ANALYSIS

Architecture and portfolio segmentation. The first continuity decision is not how to copy a system but what should change before, during, and after migration. Application portfolio assessment research recommends profiling business criticality, dependencies, technical characteristics, and cloud suitability before assigning a migration path [2]. This is important because modernization depth changes the number of failure modes introduced at cutover. Architectural reviews of legacy-to-cloud migration show that rearchitecting can unlock cloud quality benefits, but they also identify weak treatment of quality evaluation in many proposed migration approaches [3]. A zero-downtime program should therefore segment the portfolio. Stable, poorly understood systems may be safer candidates for rehosting or limited replatforming, while systems with well-understood domains and chronic scalability or maintainability constraints may justify staged refactoring.

Microservice decomposition illustrates the trade-off. Assessment frameworks can help determine whether a monolith is structurally ready for decomposition [5], yet technical debt does not disappear automatically when code is divided into services [6]. Roadmap research emphasizes preparatory architecture work, migration sequencing, and organizational coordination [7]. Empirical evidence further shows that migration toward microservices is both systemic and technical: service extraction changes deployment

ownership, monitoring needs, testing boundaries, and team interactions [8]. Systematic reviews find many decomposition techniques but continuing disagreement about how to identify optimal service boundaries and how to validate the quality of resulting architectures [9,10]. Refactoring-oriented research [12], cloud-native SaaS migration evidence [13], and systematic mapping of migration problems [14] reinforce the need to match decomposition technique to system context rather than treating service extraction as a uniform recipe. Execution-trace approaches [15] and evolutionary search methods [16] improve candidate identification, but they do not remove business-semantic judgment. Quality-attribute research likewise shows that microservices trade centralized simplicity for distributed concerns such as reliability, security, performance, and operability [17]. The implication is clear: service decomposition can support independent deployment, but premature decomposition may create more coordination points precisely when continuity risk is highest.

Sequencing and dependency control. Legacy portfolios rarely migrate independently. Shared databases, identity systems, message brokers, batch feeds, and synchronous interfaces create graph-like dependencies that determine feasible migration order. Graph-theoretic research on phase-wise migration demonstrates the value of sequencing multiple legacy applications rather than treating each move as isolated [18]. The most recent mapping research continues to identify migration strategy, decomposition, and orchestration as central concerns in monolith-to-microservice modernization [19]. For zero downtime, dependency mapping must therefore include both technical and business edges: which callers tolerate version skew, which consumers can process duplicate messages, which jobs assume local file paths, and which external partners cannot change simultaneously. An anti-corruption or API facade can reduce coupling during transition by preserving legacy contracts while internals change behind them.

Data continuity and transactional integrity. Database movement is commonly the critical path because data changes while users remain active. Workload-based migration research demonstrates that database size alone is insufficient for planning; read/write patterns, growth, source capacity, network performance, and target resources affect duration and cost [1]. For zero downtime, the important extension is continuous convergence. A baseline copy must be followed by change data capture or equivalent replication so that new writes flow toward the cloud target while the legacy system remains authoritative. Replication lag then becomes a cutover metric rather than a background statistic. The target should not receive production traffic merely because copying has completed; it should receive traffic only when lag, reconciliation, and error thresholds satisfy predefined gates.

Schema evolution is equally important. A big-bang schema replacement creates a hidden outage risk when old and new application versions overlap. The safer pattern is expand-contract evolution: additive schema changes are introduced first, both versions operate against compatible structures, data is backfilled and validated, traffic is shifted, and destructive cleanup occurs only after rollback is no longer required. This is consistent with the broader evidence that modernization should be phased and reversible [7,18]. Where a monolith owns cross-domain transactions, decomposition also requires explicit decisions about transaction boundaries, idempotency, ordering, and compensating behavior. Service extraction that leaves uncontrolled shared-table writes may create the appearance of distribution without real data ownership.

Progressive delivery and reversible cutover. Zero downtime is ultimately established at the traffic layer. Blue-green deployment maintains an old and a new environment so that routing can switch without rebuilding infrastructure; canary delivery moves a small proportion of traffic first, enabling evidence to accumulate before full exposure. Experimental work with Kubernetes, Istio, and Liquibase demonstrates how canary techniques, database change practices, and automated orchestration can be combined to support zero-downtime continuous deployment in enterprise applications [11]. These practices matter for migration because the final cutover should be a controlled series of routing decisions, not a single irreversible event.

Progressive traffic movement requires observability that is meaningful to the business. Infrastructure health alone is insufficient. A migration gate should combine

technical service-level indicators such as latency, error rate, saturation, queue depth, and replication lag with business invariants such as successful order completion, payment reconciliation, entitlement accuracy, or case-processing throughput. If either class of indicator breaches tolerance, routing must pause or reverse. This makes rollback an ordinary branch of the migration workflow rather than an emergency improvisation. The ability to reverse traffic also justifies keeping the legacy environment synchronized for an agreed stabilization period after the cloud target becomes primary.

Containerization and infrastructure reproducibility. Containerization can reduce environmental drift and make legacy deployment units more portable, although it does not by itself resolve application coupling. Advisory frameworks for application containerization show that modernization decisions can be systematized around application characteristics rather than made uniformly [22]. Infrastructure as code further supports repeatable cloud environments, auditable configuration, and rapid recreation of migration components. However, systematic research warns that infrastructure code has its own quality pitfalls and requires disciplined practices [20]. Modelling-oriented reviews similarly show continuing efforts to generate and manage infrastructure definitions more reliably [21]. In a zero-downtime program, this evidence supports treating the target landing zone, connectivity, identity configuration, observability, and rollback infrastructure as versioned products that are tested before production traffic arrives.

Security and identity continuity. Security controls can become an outage mechanism if identity, secrets, certificates, network policies, or authorization semantics change at the same moment as hosting. Research on IT modernization in the U.S. federal context links modernization and cloud migration choices with cybersecurity risk reduction, but also implies that benefits depend on how modernization is executed [4]. For migration assurance, identity and access should therefore be established early in the target environment and tested in parallel. Security controls need functional equivalence before optimization: a stricter target policy that unexpectedly blocks a business-critical integration is a continuity failure even if its security intent is valid. After cutover, modernization can strengthen segmentation, secret management, logging, and automated policy enforcement without coupling all improvements to one risky migration event.

Saudi organizational and adoption context. Saudi cloud research consistently shows that adoption is shaped by more than technical capability. Studies of Saudi enterprises and SMEs identify organizational, trust, and adoption conditions that influence movement to cloud services [23,24]. Research in Saudi higher education also demonstrates that cloud adoption decisions emerge from interacting technological and institutional determinants rather than a single factor [25]. These findings matter for zero downtime because a continuity framework depends on operating confidence: teams must trust monitoring evidence, know who owns rollback authority, understand provider responsibilities, and possess skills to run parallel estates during migration.

Saudi cybersecurity and digital-transformation studies reinforce this socio-technical interpretation. Work on business resilience connects digital transformation with cybersecurity challenges and recommends coordinated organizational responses [26]. Research on Saudi governmental digital services highlights the importance of usable security perceptions in public-service transformation [27]. Wider cloud transformation research identifies security challenges and opportunities as organizational design issues, not only platform features [28]. Mixed-methods enterprise research likewise presents cloud computing as an enabler whose benefits depend on transformation capability and information-management practices [29]. Even sector-specific Saudi work, such as cloud adoption research in construction, reports interacting political, economic, social, technological, environmental, and legal influences [30]. The practical conclusion is not that one migration blueprint fits every Saudi organization; rather, the technical control loop must sit inside explicit governance for risk acceptance, provider coordination, skills, communication, and service ownership.

Table 1 consolidates the main evidence themes and shows why no single mechanism is sufficient. Architecture determines change scope; data controls

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723551

preserve state; progressive delivery limits blast radius; infrastructure automation reduces configuration variance; observability supplies decision evidence; and governance defines authority for proceeding or reversing. A claim of zero downtime is credible only when these mechanisms operate together and are tied to measurable acceptance criteria.

Table 1. Evidence themes, continuity risks, and zero-downtime controls.

Theme	Main literature insight	Continuity risk	Zero-downtime control	Principal evidence
Portfolio and architecture	Migration path should reflect criticality, coupling, quality goals, and modernization readiness.	Too much simultaneous change expands the cutover failure surface.	Portfolio segmentation; stable interfaces; modernization depth chosen per system.	[2,3,5-10,17-19]
Data and schema	Workload, write patterns, target capacity, and data ownership shape migration behavior.	Lost, duplicated, stale, or incompatible state during active writes.	Baseline copy plus CDC; lag thresholds; reconciliation; expand-contract schemas.	[1,7,18]
Progressive delivery	Canary and blue-green approaches permit controlled exposure and rollback.	Big-bang routing makes defects immediately systemic.	Small traffic increments; SLO gates; automated pause and rollback.	[11]
Infrastructure reproducibility	Containers and infrastructure code reduce environmental variance when governed well.	Configuration drift prevents predictable cutover or recovery.	Versioned target environment; repeatable builds; tested rollback infrastructure.	[20-22]
Security and observability	Modernization benefits depend on operational evidence and integrated cyber controls.	Identity or policy changes can block legitimate workloads; blind spots delay detection.	Parallel identity validation; telemetry; technical and business indicators.	[4,26,28]
Saudi governance context	Trust, readiness, institutional conditions, and security perceptions shape cloud adoption.	Unclear responsibility or weak skills undermine dual-running decisions.	Named gate owners; provider escalation; skills assurance; evidence-based acceptance.	[23-30]

VII. DISCUSSION

The synthesis supports a framework in which zero downtime is treated as a gated control system rather than a migration label. Figure 2 represents the core operating loop. The legacy production environment remains available while a compatible cloud target is built in parallel. Continuous data capture keeps the target converging with the source. A traffic controller then exposes the cloud system progressively, while observability and business-invariant reconciliation determine whether the next increment is safe. A failed gate leads to rollback, repair, and renewed testing; a passed gate permits further traffic transfer. This loop continues until the cloud environment is primary and the legacy fallback can be retired after a defined stabilization period.

The framework differs from modernization approaches that implicitly equate cloud migration with microservice conversion. Evidence from assessment, technical-debt, roadmap, and decomposition studies

shows that microservices can improve independent deployability when boundaries are mature, but they can also introduce distributed failure modes and operational complexity [5-10,17]. Accordingly, modernization depth is a variable in the framework, not a prerequisite. A tightly coupled but stable system can first be containerized or replatformed behind a stable interface, moved with replication and progressive routing, and decomposed later under lower continuity pressure. Conversely, a monolith whose scaling bottleneck and ownership boundaries are well understood may benefit from extracting selected services before full migration.

This separation resolves an important tension in the literature. Cloud-native architecture aims to increase adaptability, whereas migration assurance aims to reduce simultaneous change. The proposed framework reconciles the two by limiting the number of irreversible transformations within any one cutover wave. Reversibility becomes the organizing design principle. Backward-compatible interfaces, expand-contract schemas, dual-running environments, idempotent data movement, and retained fallback capacity all increase the set of safe recovery actions. Infrastructure as code and containerization strengthen this principle by making target environments reproducible [20-22]. Progressive delivery provides the traffic mechanism that converts reversibility into operational practice [11].

For Saudi organizations, governance is not an outer administrative layer added after technical design. It is part of the control loop. Trust in providers, organizational readiness, cybersecurity, and institutional conditions influence cloud adoption in Saudi studies [23-27]. These factors translate directly into migration decisions: who can authorize a canary increase, which evidence is sufficient for a cutover gate, how provider incidents are escalated, what skills are available during dual running, and how business owners validate transactional outcomes. Table 2 therefore expresses the framework as technical actions plus acceptance gates and Saudi-oriented managerial emphasis. The intent is to make continuity auditable: each stage should end with evidence that can be reviewed by engineering, security, operations, and business stakeholders before the next stage begins.

The framework also clarifies what zero downtime does not mean. It does not mean keeping two systems active indefinitely, eliminating all transient errors, or guaranteeing that every migration will succeed on the first attempt. It means designing the transition so that service remains within agreed objectives and so that failures can be contained without a prolonged interruption. This is a more defensible and testable standard than an absolute promise. It shifts executive attention from a ceremonial cutover date to measurable readiness, controlled exposure, and demonstrated recovery capability.

Additional validation should prioritize realistic production dependencies, measurable business outcomes, cross-team coordination, transparent decision records, and rehearsed recovery procedures.

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723551

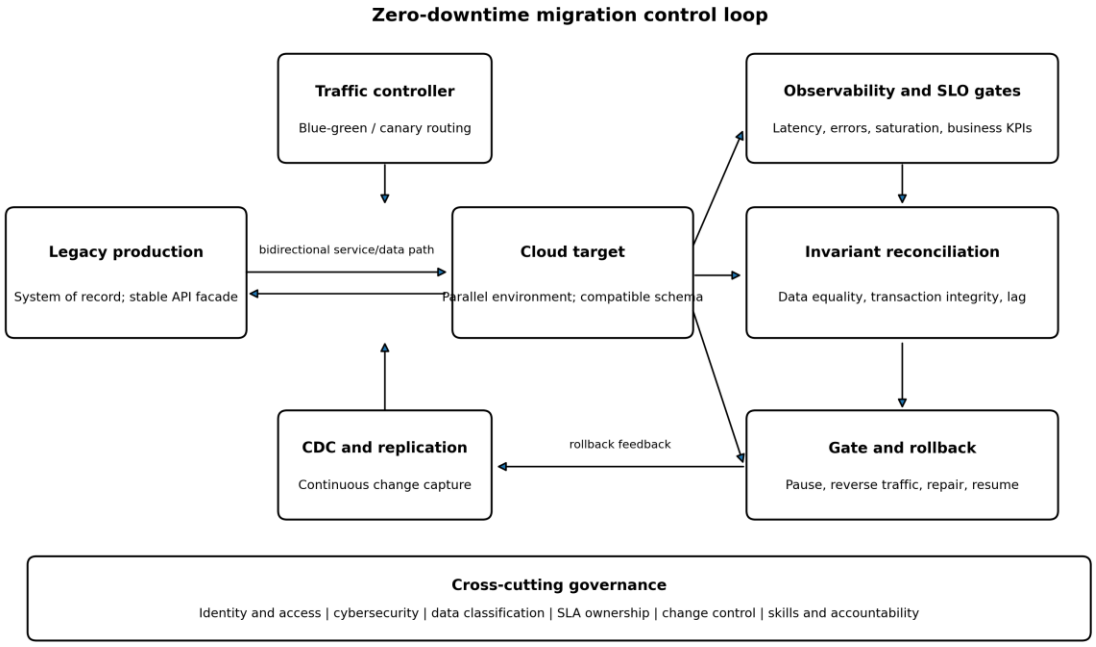

Figure 2. Zero-downtime migration control loop for legacy-to-cloud transformation.

Table 2. Saudi-oriented zero-downtime migration framework and stage gates.

Stage	Technical actions	Acceptance gate	Saudi managerial/policy emphasis
1. Discover	Inventory dependencies, workloads, critical transactions, identity paths, and recovery constraints.	Critical dependencies and business owners are identified; migration path is justified.	Readiness ownership; provider and internal responsibility mapped.
2. Prepare target	Create landing zone, connectivity, identity, observability, deployment automation, and security baseline.	Target can be reproduced and monitored; access and connectivity are validated.	Cybersecurity accountability; operational skills and escalation tested.
3. Establish data convergence	Baseline copy, CDC/replication, compatible schema evolution, reconciliation tooling.	Lag and reconciliation remain within defined tolerance under representative load.	Data stewardship and decision authority documented.
4. Parallel run	Shadow or duplicate suitable workloads; compare outputs; rehearse rollback.	Technical SLOs and business invariants match agreed baseline.	Business owner participates in evidence review.
5. Progressive cutover	Canary/blue-green traffic increments with automated gates and rollback.	Each increment passes latency, errors, saturation, security, and business checks.	Named gate approver; provider escalation path remains active.

6. Stabilize	Cloud target becomes primary; legacy remains synchronized for a controlled fallback period.	No unresolved critical divergence; recovery drill completed.	Service continuity evidence retained for audit and assurance.
7. Decommission and optimize	Archive evidence; retire legacy components; optimize cost, resilience, and architecture.	Rollback window formally closed; data retention and ownership confirmed.	Benefits realization, skills transfer, and provider performance reviewed.

VIII. RESEARCH GAPS AND FUTURE RESEARCH

The literature remains fragmented across architecture, deployment, data engineering, and adoption research. First, there is a shortage of longitudinal enterprise studies that measure end-to-end continuity across discovery, dual running, cutover, stabilization, and decommissioning. Future research should report service-level indicators, replication behavior, rollback events, and business-transaction outcomes rather than only migration completion. Second, microservice decomposition studies need stronger linkage to cutover assurance: candidate boundaries should be evaluated not only for cohesion and coupling but also for data ownership, version compatibility, observability, and reversible deployment [9,10,19].

Third, research on data migration should move beyond transfer duration toward formal and practical reconciliation of transactional invariants under continuous writes. Work is needed on cross-database consistency, event ordering, duplicate suppression, and measurable conditions for declaring a target authoritative. Fourth, infrastructure-as-code research should examine how configuration quality affects migration rollback and disaster recovery, extending current work on infrastructure-code practices and generation [20,21]. Fifth, progressive delivery research should be evaluated in heterogeneous legacy environments that include batch workloads, third-party interfaces, and shared databases, not only cloud-native service stacks [11].

Saudi-specific evidence is especially limited. Future studies should conduct anonymized case research across government, finance, healthcare, industrial, and large-enterprise settings to test whether the proposed gates are practical under local organizational and security constraints. Comparative studies could examine public cloud, hybrid cloud, and locally operated cloud models without assuming that one topology is inherently superior. Research should also investigate skills, provider trust, incident accountability, and usable security during dual running [23-30]. Finally, AI-assisted dependency discovery, anomaly detection, and reconciliation deserve evaluation, but future work should measure false assurance as carefully as automation gains; an opaque confidence score must not replace auditable migration evidence.

IX. PRACTICAL, MANAGERIAL AND POLICY IMPLICATIONS

For chief information officers and migration leaders, the principal implication is to fund continuity controls as first-class deliverables rather than as contingency work. Portfolio discovery, replication, observability, automated rollback, and dual-running capacity may appear to increase short-term cost, but they create the evidence and recovery options required for a safe transition. Architecture teams should explicitly separate the hosting decision from the refactoring decision and justify modernization depth using application criticality, coupling, data ownership, and operational readiness [2,5,7].

Program managers should organize migration waves around dependency graphs and acceptance gates, not calendar milestones alone [18]. Each gate should have named technical and business owners, measurable criteria, and a rehearsed rollback path. Security teams should test identity, secrets, certificates, logging, and policy behavior in the target environment before traffic migration rather than combining security redesign with final cutover [4,26,28]. Operations teams should own service-level dashboards and reconciliation evidence during canary exposure.

For Saudi organizations, policy and governance should encourage auditable responsibility across

enterprise teams and cloud providers. Studies on trust, readiness, and digital-service security suggest that adoption confidence depends on clear expectations and usable controls [23-27]. Procurement and service-management arrangements should therefore define telemetry access, incident escalation, recovery responsibilities, and evidence required for continuity assurance. Capability development is equally important: a technically strong cloud platform cannot compensate for unclear rollback authority or insufficient operational skills during dual running.

X. LIMITATIONS

This review has four limitations. It integrates heterogeneous software-engineering and information-systems evidence rather than estimating a pooled effect size. The bibliography is deliberately selective and centered on peer-reviewed work most relevant to the proposed framework; it is not a census of all cloud-migration publications. No numerical screening flow is reported because a frozen search export and deduplication log were not maintained. Finally, Saudi-specific studies largely address adoption, trust, cybersecurity, and digital transformation rather than observed zero-downtime migrations, so the national adaptation remains a framework to be empirically validated rather than a demonstrated causal model.

XI. CONCLUSION

Zero-downtime migration of legacy enterprise systems is best understood as a continuity-assurance problem that spans architecture, data, deployment, operations, security, and governance. The evidence reviewed here does not support a universal prescription to convert every legacy system into microservices before cloud adoption. It supports a more disciplined sequence: understand the portfolio and dependencies; create a reproducible and secure target environment; establish continuous data convergence; preserve backward compatibility; run old and new environments in parallel; shift traffic progressively; reconcile technical and business outcomes; and retain a tested rollback path until stability is demonstrated.

The proposed Saudi-oriented framework converts these principles into stage gates. Its defining feature is reversibility: each migration step should increase confidence without removing the previous safe state prematurely. This allows organizations to choose different modernization depths while applying the same continuity logic. For Saudi digital transformation, the framework also makes trust, cybersecurity, readiness, provider coordination, and accountable decision rights part of the technical migration system rather than peripheral management concerns. Future empirical work should test the framework against real Saudi migrations and report service-level, data-integrity, rollback, and business-process outcomes. Until such evidence expands, the most defensible route to zero downtime is not a single cloud technology but a measurable, progressive, and reversible transition process.

REFERENCES

[1] M. Ellison, R. Calinescu, and R. F. Paige, “Evaluating cloud database migration options using workload models,” *Journal of Cloud Computing*, vol. 7, Art. no. 6, 2018. Springer

[2] K. Ramchand, M. Baruwal Chhetri, and R. Kowalczyk, “Enterprise adoption of cloud computing with application portfolio profiling and application portfolio assessment,” *Journal of Cloud Computing*, vol. 10, Art. no. 1, 2021. Springer

[3] M. H. Hasan, M. H. Osman, N. I. Admodisastro, and M. S. Muhammad, “Legacy systems to cloud migration: A review from the architectural perspective,” *Journal of Systems and Software*, vol. 202, Art. no. 111702, 2023. ScienceDirect

[4] M.-S. Pang and H. Tanriverdi, “Strategic roles of IT modernization and cloud migration in reducing cybersecurity risks of organizations: The case of U.S. federal government,” *Journal of Strategic Information Systems*, vol. 31, no. 1, Art. no. 101707, 2022. ScienceDirect

[5] F. Auer, V. Lenarduzzi, M. Felderer, and D. Taibi, “From monolithic systems to microservices: An assessment framework,” *Information and Software Technology*, vol. 137, Art. no. 106600, 2021. ScienceDirect

[6] V. Lenarduzzi, F. Lomio, N. Saarimäki, and D. Taibi, “Does migrating a monolithic system to microservices decrease the technical debt?”

Journal of Systems and Software, vol. 169, Art. no. 110710, 2020. ScienceDirect

[7] D. Wolfart, W. K. G. Assunção, I. F. da Silva, D. C. P. Domingos, E. Schmeing, G. L. D. Villaca, and D. N. Paza, "Modernizing legacy systems with microservices: A roadmap," in *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering*, pp. 149–159, 2021. ACM

[8] H. M. Ayas, P. Leitner, and R. Hebig, "An empirical study of the systemic and technical migration towards microservices," *Empirical Software Engineering*, vol. 28, Art. no. 85, 2023. Springer

[9] Y. Abgaz, A. McCarren, P. Elger, D. Solan, N. Lapuz, M. Bivol, G. Jackson, M. Yilmaz, J. Buckley, and P. Clarke, "Decomposition of monolith applications into microservices architectures: A systematic review," *IEEE Transactions on Software Engineering*, vol. 49, no. 8, pp. 4213–4242, 2023. IEEE

[10] I. Oumoussa and R. Saidi, "Evolution of microservices identification in monolith decomposition: A systematic review," *IEEE Access*, vol. 12, pp. 23389–23405, 2024. IEEE

[11] A. Malhotra, A. Elsayed, R. Torres, and S. Venkatraman, "Evaluate canary deployment techniques using Kubernetes, Istio, and Liquibase for cloud native enterprise applications to achieve zero downtime for continuous deployments," *IEEE Access*, vol. 12, pp. 87883–87899, 2024. IEEE

[12] J. Correia and A. R. Silva, "Identification of monolith functionality refactorings for microservices migration," *Software: Practice and Experience*, 2022. Wiley

[13] E. T. Nordli, S. G. Haugeland, P. H. Nguyen, H. Song, and F. Chauvel, "Migrating monoliths to cloud-native microservices for customizable SaaS," *Information and Software Technology*, vol. 160, Art. no. 107230, 2023. ScienceDirect

[14] V. Velepucha and P. Flores, "Monoliths to microservices - Migration problems and challenges: A systematic mapping study," in *2021 International Conference on Information Systems and Technologies*, pp. 1–7, 2021. IEEE

[15] W. Jin, T. Liu, Y. Cai, R. Kazman, R. Mo, and Q. Zheng, "Service candidate identification from monolithic systems based on execution traces," *IEEE Transactions on Software Engineering*, vol. 47, no. 5, pp. 987–1007, 2021. IEEE

[16] K. Sellami, A. Ouni, M. A. Saied, S. Bouktif, and M. W. Mkaouer, "Improving microservices extraction using evolutionary search," *Information and Software Technology*, vol. 151, Art. no. 106996, 2022. ScienceDirect

[17] S. Li, H. Zhang, Z. Jia, C. Zhong, C. Zhang, Z. Shan, J. Shen, and M. A. Babar, "Understanding and addressing quality attributes of microservices architecture: A systematic literature review," *Information and Software Technology*, vol. 131, Art. no. 106449, 2021. ScienceDirect

[18] R. Punnoose and S. K. De, "Phase-wise migration of multiple legacy applications: A graph-theoretic approach," *Information and Software Technology*, vol. 137, Art. no. 106606, 2021. ScienceDirect

[19] A. Martínez Saucedo, G. H. Rodríguez, F. Rocha, and R. Pereira dos Santos, "Migration of monolithic systems to microservices: A systematic mapping study," *Information and Software Technology*, vol. 177, Art. no. 107590, 2025. ScienceDirect

[20] Kumara, M. Garriga, A. U. Romeu, D. Di Nucci, D. A. Tamburri, W.-J. van den Heuvel, and F. Palomba, "The do's and don'ts of infrastructure code: A systematic gray literature review," *Information and Software Technology*, vol. 137, Art. no. 106593, 2021. ScienceDirect

[21] G. Novakova Nedeltcheva, A. De La Fuente Ruiz, L. Orue-Echevarria Arrieta, N. Bat, and L. Blasi, "Towards supporting the generation of infrastructure as code through modelling approaches: Systematic literature review," in *2022 IEEE International Conference on Software Architecture Companion*, pp. 210–217, 2022. IEEE

[22] A. Kalia, R. Batta, J. Xiao, and M. Choudhury, "ACA: Application containerization advisory framework for modernizing legacy applications," in *2021 IEEE 14th International*

Conference on Cloud Computing, pp. 700–707, 2021. IEEE

[23] N. Alamri and S. Alzahrani, "Navigating the clouds: An exploratory research of cloud computing adoption in Saudi Arabia's small and medium enterprises," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17859–17869, 2024. ETASR

[24] W. Alghaith, "The role of trust in cloud service providers on the adoption of cloud computing in Saudi Arabia: An empirical investigation," *Journal of Computer Science*, vol. 19, no. 9, pp. 1107–1123, 2023. Science Publications

[25] N. S. Aldahwan and M. S. Ramzan, "Quadruple theories based determinants and their causal relationships affecting the adoption of community cloud in Saudi HEI," *BioMed Research International*, vol. 2022, Art. no. 2382535, 2022. Wiley

[26] S. Saeed, S. A. Altamimi, N. A. Alkayyal, E. Alshehri, and D. A. Alabbad, "Digital transformation and cybersecurity challenges for businesses resilience: Issues and recommendations," *Sensors*, vol. 23, no. 15, Art. no. 6666, 2023. MDPI

[27] S. Saeed, "Digital transformation in governmental public service provision and usable security perception in Saudi Arabia," *Information*, vol. 16, no. 3, Art. no. 247, 2025. MDPI

[28] X. Liang and Y. Xu, "A novel framework to identify cybersecurity challenges and opportunities for organizational digital transformation in the cloud," *Computers & Security*, vol. 151, Art. no. 104339, 2025. ScienceDirect

[29] T. R. Merlo, F. Fard, and S. Hawamdeh, "Cloud computing's impact on the digital transformation of the enterprise: A mixed-methods approach," *Sustainability*, vol. 17, no. 13, Art. no. 5755, 2025. MDPI

[30] N. S. Alaboud, A. Elmansoury, and P. Jia, "Implementation of cloud computing in Saudi construction: An exploration of political, economic, social, technological, environmental, and legal factors using fuzzy DEMATEL," *Advances in Civil Engineering*, vol. 2025, Art. no. 1825108, 2025.