CLOUD COMPUTING ARCHITECTURES

Resilient Cloud Computing Architectures for Mission-Critical Financial Systems in Saudi Arabia: Integrating Oracle Cloud Infrastructure, Multi-Domain High Availability, Automated Failover and Business Continuity under Vision 2030

Mohammad Thousif Rahman

Published November 2026

Abstract

Saudi financial institutions require uninterrupted transaction processing despite software defects, regional service disruption, cyberattack and electricity-system disturbances. This critical review evaluates how Oracle Cloud Infrastructure (OCI) can support business continuity through fault domains, availability domains, independent regional recovery, automated failover and verifiable data restoration. The evidence base comprises thirty DOI-identifiable publications from 2020–2025, interpreted alongside official Saudi regulatory and provider documentation. A structured thematic synthesis distinguishes empirically supported reliability mechanisms from vendor capabilities and design recommendations. The proposed reference architecture combines distributed application replicas, controlled database replication, dependency-aware recovery orchestration, immutable backups and independently protected management paths. It treats recovery-time and recovery-point objectives as measurable outcomes rather than configuration labels. The review shows that geographical redundancy does not ensure operational resilience when identity services, transaction integrity, telecommunications, power supply or human authorisation remain common points of failure. Automated switching may shorten interruption yet amplify damage if corrupted data, stale state or compromised credentials propagate. For Saudi Arabia, domestic-region deployment, explicit supervisory requirements, heat-sensitive cooling loads and the reliability of electrical infrastructure change the appropriate balance among availability, sovereignty, energy use and cost. A risk-tiered validation method is proposed using dependency mapping, fault injection, recovery drills and financial integrity tests. The principal contribution is an integrated, auditable architecture and research agenda for Saudi banking and fintech under Vision 2030; the paper reports no new field experiments or measured gains.

Keywords: cloud resilience; Oracle Cloud Infrastructure; banking continuity; availability domains; automated failover; disaster recovery; Saudi Vision 2030; electrical infrastructure

1 Introduction

Digital banking, card authorisation, lending platforms and payment gateways must preserve both service accessibility and an accurate financial record. A conventional uptime measure can conceal rejected payments, duplicated instructions or inconsistent account balances. The consequences of outages therefore extend beyond lost application availability to liquidity management, customer confidence, regulatory obligations and reconciliation workload. Cloud computing promises rapid provisioning and geographically distributed capacity, yet concentration in shared providers can reproduce or enlarge common-cause risk. Research on banking adoption illustrates that operational

risk can rise even where profitability improves [1]. Resilience must consequently be designed as an end-to-end service property rather than inferred from the presence of cloud infrastructure.

Saudi Arabia offers a particularly important setting. The Financial Sector Development Program and Vision 2030 encourage digital financial services, innovation and operational maturity. Banks and finance companies operate under Saudi Central Bank requirements relating to cybersecurity, cloud outsourcing and business continuity. The Saudi Central Bank's cloud controls emphasise domestic location as the normal preference, subject to applicable supervisory approval where services are outside the Kingdom; the precise application depends on the institution and current rulebook. Its continuity framework requires continuity governance, business-impact analysis and tested plans. The architecture explored here therefore uses a Saudi-hosted production and recovery design as the working assumption, but does not treat provider marketing descriptions as evidence of regulatory approval.

Oracle Cloud Infrastructure provides regions, availability domains and fault domains, yet the available number of availability domains is region-specific. A single-availability-domain region may still offer three fault domains, whereas recovery from loss of an entire region requires a separate region and independent operational arrangements. Describing fault-domain placement as multi-availability-domain resilience would misstate the protection obtained. This distinction is critical when architects design for Saudi regions, select backup destinations and estimate achievable recovery objectives. Provider documentation is useful for understanding supported features; it does not demonstrate a particular financial institution's realised availability.

The International Monetary Fund's 2024 financial-system assessment places Saudi financial development within a wider stability and supervisory programme [2]. This provides useful national context, but does not establish that any particular bank can withstand a cloud outage. For design review, the institution should specify which customer services contribute to systemic confidence, which payment rails are time critical and which manual workarounds can operate legally during a digital interruption. A service may recover technically while outstanding payments remain uncertain; therefore, continuity classification should consider settlement obligations, customer harm and dependencies on other regulated participants, not merely revenue associated with the application.

Existing reviews classify reactive, proactive and adaptive tolerance mechanisms [3–7], but research frequently treats compute resilience independently from database semantics, governance, electricity supply or financial correctness. Availability modelling further identifies how network dependencies and shared infrastructure undermine simplistic multiplication of component reliability [8]. This paper connects these strands, interrogates automation risk and specifies an auditable operating model. Its original contribution is conceptual synthesis: no claimed production deployment, simulated outage result or empirical recovery improvement is presented.

1.1 Aim and objectives

The aim is to establish a defensible, Saudi-specific framework for resilient OCI-based financial services. The objectives are to distinguish fault domains, availability domains and regions; evaluate active-active and active-passive alternatives; analyse automated switching, database consistency and cyber-recovery safeguards; integrate business continuity, oversight and electrical infrastructure; and define measurable assurance criteria and unresolved research questions. The review asks which failure modes each protection layer actually covers, what new hazards automation introduces and how evidence should be produced for independent validation.

2 Review methodology

This study adopts a structured critical narrative review rather than presenting itself as an exhaustive systematic review. Searches undertaken for this manuscript used combinations of cloud fault tolerance, microservice resilience, cloud-native databases, financial operational risk, business continuity, disaster recovery, zero trust, data-centre reliability and electrical flexibility. Evidence was identified through scholarly publisher and institutional discovery pages, journal records and DOI-linked publication metadata. Eligible scholarly items were published or assigned to 2020–2025 journal or conference volumes, were in English and directly informed at least one architectural or governance claim. Thirty DOI-identifiable items were retained; provider documentation and Saudi regulatory texts were consulted separately as normative or product-specific context, not counted as peer-reviewed outcome studies.

Screening privileged technical specificity, transparent research purpose and relevance to availability, replication, recovery, governance or power-system dependencies. Studies outside Saudi Arabia were retained when mechanisms were transferable, while their local measurements were not extrapolated to Saudi conditions. Research designs were categorised as conceptual review, analytical model, experiment, simulation, empirical banking analysis or regulatory assessment. This classification matters because a successful laboratory fault injection cannot establish financial-sector readiness, and a legal critique cannot validate failover latency. Searches were not claimed to cover every indexed database record; no numerical identification or exclusion flow is invented.

Extraction used five questions: what component or process fails; how failure is detected; what state and service dependencies must recover; which integrity and recovery measurements are reported; and which assumptions constrain transfer to Saudi financial institutions. Evidence was then compared across four layers: runtime availability, transactional recoverability, organisational control and underlying electrical-telecommunications infrastructure. Contradictory findings were retained rather than averaged. When source publications test unrelated workloads, their percentages or response times are treated as context-specific, not predicted OCI outcomes.

Methodological limitations include heterogeneous workload definitions, limited availability of bank-specific operational data and rapid changes in commercial cloud services. DOI resolution and bibliographic metadata were checked against accessible publisher, repository or indexed discovery records where possible; such checks do not establish study quality. The thirty sources represent a bounded thematic sample, not a meta-analysis. Vendor capability statements are distinguished from empirical evidence in the analysis, and all proposed evaluation thresholds are illustrative until an institution's business-impact assessment approves them.

3 Architectural foundations and failure boundaries

Cloud resilience is the capacity to absorb faults while sustaining acceptable financial outcomes, not simply the ability to restart servers. Reviews distinguish redundancy, checkpointing, migration and self-healing [3–6]. These methods protect different boundaries. Application replicas within separate fault domains address host or rack-level disruption; multiple availability domains, where a region has them, address a broader physical boundary; independent regions protect against some regional events. None alone addresses malicious database writes, incorrect payment logic or an expired identity credential. A realistic architecture maps each critical business service to the weakest dependency on its execution path.

Figure 1 presents an original two-region reference model, with regional and fault-domain isolation shown separately. Production traffic enters an independently controlled routing tier before reaching replicated application services. Stateless components can scale horizontally, but stateful databases require explicit roles and transaction-aware replication. A cross-region standby receives approved data changes and configuration packages. Backup copies remain logically protected from the production administrator path. During a regional event, the recovery runbook must restore identity, networking, encryption keys, payment integrations and monitoring before exposing the standby as a transactional system.

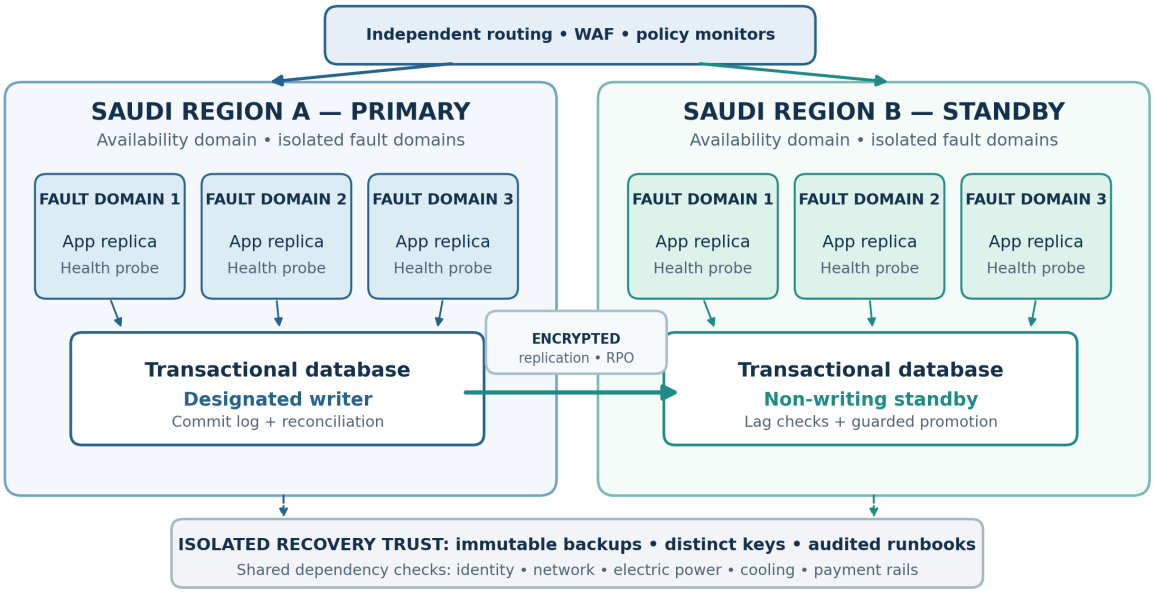

Figure 1. Conceptual Saudi-hosted OCI architecture showing local fault-domain protection, cross-region standby and independent recovery controls (original diagram).

Where OCI offers only one availability domain in a particular region, placing three application replicas across distinct fault domains reduces exposure to an individual host-group incident but does not create three independent geographical sites. A regional pair may provide stronger disaster separation, subject to supported service availability, legal restrictions and live network capacity. The architecture must explicitly test routing convergence, DNS caching, firewall policy drift and inter-region latency. A standby that runs in a separate location but shares the same provisioning credentials or privileged directory is geographically separated yet operationally coupled.

Active-active operation can improve utilisation and local fault response, but distributing writes across financial ledgers introduces ordering, conflict resolution and settlement risks. Active-passive operation simplifies a single-writer invariant and may cost more in underutilised capacity. Intelligent placement of microservice replicas improves resilience under certain analysed workloads [9], although optimisation studies do not override bank-specific dependency and consistency constraints. Selection should begin with a classified service catalogue: customer enquiry may tolerate brief stale reads; payment posting and account balances ordinarily demand stronger invariants and explicit duplicate prevention.

3.1 Observability and failure detection

Automated action is only as dependable as its evidence. Health checks based solely on process liveness can promote a standby while payment APIs remain unable to post transactions. Service-level indicators should therefore include authorised payment completion, reconciliation lag, database replication delay, queue age, error budgets, third-party connectivity and customer-visible transaction status. Self-healing literature highlights the potential of proactive diagnosis but also uncertainty surrounding prediction quality [4]. An isolated symptom must be correlated with dependent services before a disruptive recovery step is executed.

A resilient control plane should not rely on the failing workload to authorise its own recovery. Independent telemetry collection, signed configuration, separate privileged access and tamper-evident audit trails reduce correlated compromise. Microservice fault-injection studies demonstrate the value of request-level experiments for exposing hidden dependencies [10]. However, synthetic failures must be scoped to avoid unintended payment authorisations, disclosure of real customer data or propagation into production settlement processes. Safe tests use masked data, controlled fault budgets and documented rollback responsibilities.

4 Automated failover and orchestration

Failover encompasses detection, diagnosis, decision, fencing, state reconciliation, capacity readiness, routing and controlled resumption. An automated trigger that equates a brief network partition with destruction of the primary site may create two writable database leaders. Fencing must revoke or isolate the old writer before promoting the new one, with quorum and authoritative state checks fitted to the chosen database technology. The cross-region procedure should also verify the last committed transaction boundary and outstanding messages. Availability research based on dependency-aware Bayesian models shows why independence assumptions require scrutiny [8].

Figure 2 presents an original decision sequence. It requires an operator-approved policy envelope and explicit stop conditions, even where individual recovery operations are automated. Observed application failure first triggers correlation with network, identity and database signals. When local remediation is safe, traffic stays regional; when a site event is credible, an orchestrator validates standby state and protected credentials. If replication lag exceeds the approved recovery-point objective, the automation holds promotion for adjudication rather than silently sacrificing ledger correctness. Following promotion, immutable evidence and reconciliation outputs support a controlled return to primary service.

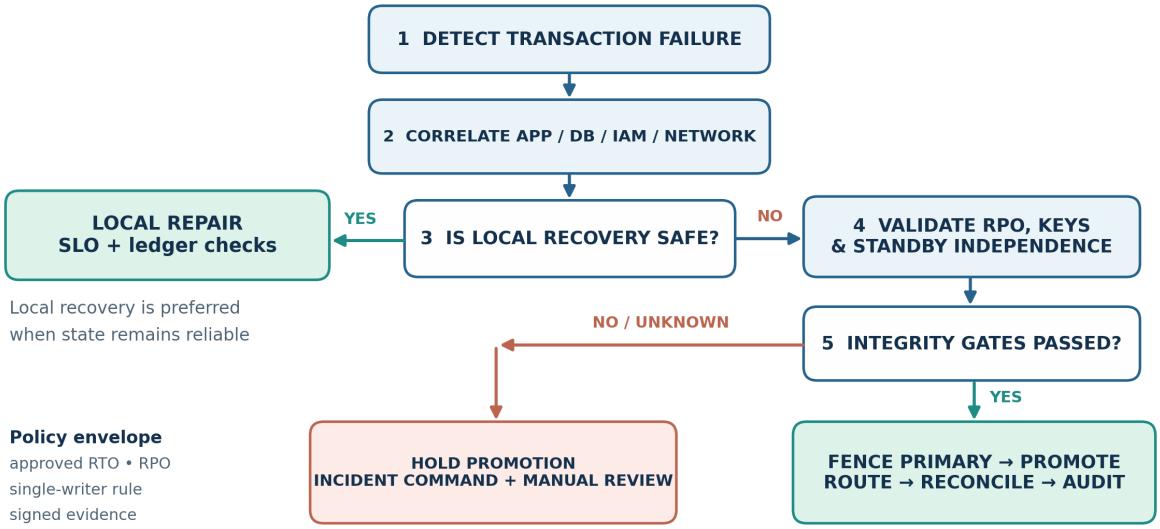

Figure 2. Safety-gated automatic failover decision path prioritising transactional integrity and bounded recovery objectives (original workflow).

Patterns such as retries, circuit breakers and connection limits reduce cascading failures only when configured against realistic time budgets. Controlled experiments show that retries and circuit breakers can worsen latency at higher failure rates [11]. Excessive retries multiply pressure on constrained database writers, while incorrectly configured breakers may block healthy dependencies. Every payment API should implement idempotency keys, transaction outcome inquiry and bounded back-off. Message replay must use persistent deduplication and exactly-defined business effects; transport-level delivery guarantees should never be confused with exactly-once financial posting.

Recovery orchestration should use infrastructure and policy as code, versioned within protected repositories and tested before release. Simulation studies of microservice resilience offer a way to compare candidate controls without assuming that simulated recovery reproduces all physical faults [12]. Drift detection should flag missing firewall rules, expired certificates, inaccessible key vaults and inadequate spare capacity. Critical steps need a manual override with independent approval, because an attacker who controls the automation engine can transform high-availability tooling into an efficient disruption mechanism.

5 Data integrity, replication and recoverability

Recovery-time objective measures the maximum acceptable duration to restore a defined business process; recovery-point objective expresses the maximum tolerable loss of recoverable data. Neither is a universal property of a cloud subscription. The acceptable values depend on transaction class, settlement deadlines, payment-network agreements and business-impact findings. Zero-data-loss aspirations require careful examination of synchronous commitment, latency, write availability and failure boundaries. Cross-region synchronous replication may incur delay or reduce availability during connectivity failure; asynchronous replication accepts a bounded potential loss window that must be documented and monitored.

Distributed database studies describe the trade-offs between replica placement, commit coordination and performance [13–16]. Epoch-based commit research demonstrates alternatives for balancing

throughput and reliable replication [14], but its experimental architecture cannot be treated as an OCI configuration benchmark. Database consistency experiments show that stronger guarantees may increase latency depending on the technology and workload [15]. A bank should test actual ledger semantics, audit ordering and reconciliation, rather than adopting generic throughput claims from non-financial benchmarks.

A practical design separates operational replication from immutable recovery copies. Replicas reproduce changes rapidly, including accidental deletion or malicious encryption; backups preserve earlier recoverable states when properly isolated. The recovery catalogue should retain encrypted point-in-time copies, transaction logs, restoration manifests and tested key retrieval. Recovery validation should confirm journal continuity, customer balances, duplicate suppression, standing instructions and regulatory record retention. The data owner must certify the restored state, not merely that a database instance has reached an online status.

A three-stage cutover reduces ambiguity: first establish an authoritative data checkpoint; then fence obsolete writers and validate the standby; finally expose customer traffic with staged release and reconciliation. Read-only access may resume before financial posting when integrity remains uncertain, provided customers receive an accurate service notice. An explicit transaction-freeze protocol and exception queue are often safer than an unqualified immediate cutover. This prioritises correct settlement over cosmetic uptime and creates a defensible audit sequence.

6 Cyber resilience and business continuity

Cloud redundancy does not prevent a privileged user, compromised supply-chain package or ransomware operation from corrupting both primary and standby environments. Business continuity must therefore include cyber-isolated recovery, verified clean images, independent identity paths and separate backup credentials. NIST guidance on destructive-event recovery emphasises restoration of trustworthy data [17], while service-mesh control publications describe mutual authentication, attribute-based access and policy-enforced microservice boundaries [18, 19]. These are necessary controls but not proof against compromised administrators or shared certificate authorities.

Financial-sector research argues that cloud-native DevSecOps demands security decisions throughout the software lifecycle [20]. Workload identity should be short-lived, narrowly authorised and traceable to an approved deployment. Production and recovery regions should be evaluated for common dependence on image registries, secrets services, domain-name systems and orchestration pipelines. Security monitoring requires independent retention, and recovery playbooks should assume that the primary management environment may be untrusted after an attack. The organisation must retain a viable path for restoring certificates, keys and privileged access.

Cyber-recovery economics adds another dimension. Baker and Shortland [21] show why post-incident loss mitigation shapes organisational choices in ransomware response; insurance may finance some losses but cannot restore a trusted ledger or prevent correlated operational harm. Financial institutions should quantify the cost of prolonged service degradation, investigative downtime, customer remediation and transaction reconstruction. These costs are asymmetric: a premature restart that processes inaccurate balances may create greater obligations than a short, transparent suspension. Recovery exercises should therefore include a simulated declaration that data integrity is uncertain, require independent evidence before resuming posting and test whether senior decision-makers can withstand pressure to restore visible uptime prematurely.

A continuity plan connects technical recovery to incident command, communication and downstream institutions. Table 1 compares layers by failure coverage, residual risk and audit evidence. Contractual arrangements must identify the provider's infrastructure responsibilities and the institution's duties for configuration, backup policy, access governance and application integrity. Responsibility-model research cautions that service models change who is accountable for particular continuity actions [22]. Contract clauses cannot replace live recovery exercises, and a supplier's general certification does not establish readiness for a particular payment workflow.

Table 1. Comparative resilience controls, coverage boundaries and audit evidence (author-developed synthesis).

Resilience control	Failure boundary addressed	Residual vulnerability	Audit evidence
Fault-domain application replicas	Host, local network or power-group faults	Regional outages; shared identity dependencies	Replica placement, isolation tests, service error budget
Cross-region warm standby	Regional unavailability and site loss	Replication lag; cross-region configuration drift	RPO measurements, recovery logs, latency profiles
Single-writer fencing	Duplicate leaders during partition or promotion	Stale sessions and delayed downstream settlement	Fencing proof, journal continuity, reconciliation results
Immutable independent backups	Destructive writes and production credential compromise	Shared encryption keys or untested restoration	Restore rehearsal, key custody, backup access logs
Manual continuity fallback	Unsafe automation or unresolved financial state	Reduced throughput; operational delay	Incident authority, customer notice, settlement exception register

7 Saudi governance, sovereignty and concentration risk

The Saudi Central Bank's cloud-control and business-continuity provisions require a conservative architectural interpretation. Domestic regions can support localisation strategies, but location alone is insufficient: telemetry exports, overseas support access, backup replication, key custody and contractor administration may constitute additional data flows requiring assessment. Personal-data processing also engages Saudi privacy requirements. Institutions should map each service, dataset and subcontractor to the applicable rules, obtain approvals where required and document residual third-party risks before enabling replication routes.

Cloud concentration presents a governance issue even where every replica meets technical specifications. An interruption affecting shared authentication, regional connectivity or a widely used provider service may affect many financial institutions simultaneously. Banking research links cloud adoption with changes in operational risk [1], while European financial-regulation analyses illustrate the transition from institution-level outsourcing risk towards systemic oversight of critical providers [23, 24]. These foreign legal frameworks are comparative evidence only; they must not be presented as Saudi legal requirements.

A board-approved resilience appetite should identify critical services, maximum disruption tolerances, integrity thresholds, reportable incidents and accountable executives. Risk committees need evidence of failover testing and recovery controls, not just service-level agreements. An exit strategy should address data export, encryption-key ownership, application portability and replacement capacity. Multi-cloud failover may reduce some vendor dependencies while introducing operational complexity, inconsistent security policy and new synchronisation risks. The correct

target is demonstrable recoverability within the governing Saudi constraints, not multi-cloud deployment for its own sake.

8 Electrical infrastructure and physical continuity

Every digital financial transaction ultimately relies on electricity, power conversion, cooling and telecommunications. Data-centre reliability studies show that server redundancy cannot compensate for poorly modelled power distribution or cooling components [25]. For Saudi deployments, summer thermal stress, cooling electricity demand, backup generation, dual utility feeds and fibre-route diversity should be included in continuity scenarios. This does not imply that a given Saudi cloud region has a known weakness; it identifies dependencies that institutions should verify through contractual evidence and independent assurance.

Energy reviews document the performance measures and trade-offs of efficient data-centre designs [26]. Research on battery storage suggests that flexibility services may be economically attractive but require safeguards against compromising data-centre reliability [27]. Grid-reliability models explore how adjusting computing loads may support power-system operation [28]. Such studies cannot justify moving real-time payment workloads simply to follow electricity prices. Financial workloads should be ranked by latency, criticality and data-sovereignty restrictions; only non-critical batch processing may offer discretionary scheduling flexibility under tightly defined operating constraints.

Distributionally robust data-centre planning provides an additional analytical perspective on electricity and workload uncertainty [29]. However, optimisation models often assume the operator can shift computational demand without breaching externally imposed transaction deadlines. In Saudi banking, that assumption should be constrained by payment cut-off times, the hours of overseas correspondent institutions, audit retention and approved data locations. Engineering teams can schedule backup verification, archival analytics and certain non-interactive jobs to reduce peak demand, but the protected capacity for real-time payment authorisation must remain reserved. Energy resilience should be assessed using worst-case coupled outages rather than only expected operating cost.

A cross-region continuity strategy must test correlated heat events and grid disturbances, not presume geographical separation guarantees independent electricity exposure. Evidence should cover uninterruptible power supply autonomy, generator refuelling, cooling redundancy, maintenance windows, fuel logistics and black-start priorities, subject to the provider's disclosure boundaries. Table 2 defines an assurance matrix connecting failure events with recovery objectives and verification artefacts. The proposed approach integrates electrical engineering into financial ICT risk management, consistent with Vision 2030's emphasis on dependable digital and physical infrastructure.

Table 2. Risk-based exercise and verification matrix for Saudi financial workloads (proposed assurance protocol).

Injected event	Expected service response	Financial integrity gate	Verification artefact
One fault-domain outage	Local failover; no regional promotion	No duplicate posting; bounded error rate	Per-transaction trace and domain-event timeline
Saudi primary-region isolation	Guarded promotion after independent checks	Old writer fenced; approved RPO maintained	Promotion log, route change and reconciled ledger

Injected event	Expected service response	Financial integrity gate	Verification artefact
Replication delay above approved RPO	Halt automatic promotion; raise incident command	Reject unverifiable commitment point	Lag snapshot, approval and decision record
Corrupt payment event or credential abuse	Quarantine; restore verified state	Preserve audit trail and settlement ordering	Signed logs, key-access record, restore test
Joint power/cooling and carrier disturbance	Exercise independent capacity and manual response	No premature ledger restart	Facility dependency map and recovery rehearsal

9 Assurance model and evaluation protocol

The framework should be evaluated through a staged, institution-approved test programme. First, teams construct a dependency graph linking the customer journey to applications, queues, databases, identity, DNS, networks, electricity and vendors. Second, they select representative transaction classes and record acceptable recovery objectives through business-impact analysis. Third, controlled tests inject faults at increasing scope: container termination, host loss, fault-domain outage, replication interruption, compromised credentials, full regional isolation and data corruption. Fourth, independent observers verify balances, audit trails, logs, queued instructions and customer communications after recovery.

Performance indicators should include detection latency, time to fence a failed writer, decision latency, traffic restoration, the age of the last valid commit, reconciliation exceptions, failed customer journeys and time to complete manual approval. An illustrative test could assign a thirty-minute service recovery target and a five-minute recoverable-data bound to a non-settlement function; these figures are examples, not regulatory thresholds or claims of demonstrated performance. For a payment ledger, the institution may set stricter requirements and reject any switch that risks an unverified committed transaction.

All tests need documented preconditions, failure injection scope, independent timestamps and reproducible success criteria. Results should distinguish partial API recovery from complete business restoration. Fault-injection frameworks and simulation research inform safe test design [10, 12]; cloud-service Bayesian models warn that common-cause events can invalidate simplified availability estimates [8]. Post-test review should record latent dependencies and change corrective action ownership. Passing an exercise does not establish permanent resilience; new software releases, network routes and provider configurations can invalidate assumptions.

The evidence package should make negative results useful. For every unsuccessful exercise, the team should classify whether failure arose from a false alert, hidden dependency, inaccessible credential, inadequate standby capacity, inconsistent data or delayed human authorisation. Corrective actions should change the architecture, playbook or decision rule and be retested before closure. Audit reports must retain the scope and date of each test, because validation of one software release or power scenario does not validate subsequent configuration changes. This supports comparison without unsupported service-wide availability claims.

Assurance also needs cost analysis. Spare capacity, inter-region replication traffic, backup retention, software licensing, security isolation and continuous rehearsals produce recurrent expense. A complete business case compares expected disruption exposure and risk reduction with total lifecycle cost, not merely cloud hosting price. More replicas may offer little additional value where the

dominant loss scenario concerns a common identity service or corrupted primary data. Portfolio design should direct investment towards verified bottlenecks and protect recovery funding even during cost-cutting cycles.

10 Critical discussion and research gaps

The literature supports layered fault tolerance, but transferability remains constrained. First, many evaluations treat synthetic microservice latency as a proxy for resilience, overlooking settlement finality, regulatory notifications and transaction reconciliation. Second, database studies examine consistency and throughput without measuring the operational complexity of credential restoration, application promotion and downstream payment interfaces. Third, cloud availability models identify common-cause failure but often omit electricity procurement, cooling stress and institutional escalation delays. Fourth, banking studies observe cloud-related operational risk yet rarely reveal the granular outage traces needed to identify architectural mechanisms.

A further unresolved tension concerns autonomy. Automated failover may reduce response time for clean infrastructure failures while increasing loss after a cyber compromise or undetected state divergence. Research on proactive self-healing rarely establishes how a financial institution should quantify the cost of a false positive [4]. Equally, zero-trust approaches can restrict lateral movement but introduce dependency on identity and policy distribution. Reliability engineering must therefore examine negative interactions between controls, including retries during database saturation, service-mesh policy errors and backups replicated through compromised trust boundaries.

The geographical boundary of resilience also deserves sharper treatment. Provider regions, fault domains and availability domains are not interchangeable, and some services have region-specific limitations. Statements that an OCI workload is 'multi-domain' must name the exact domain level, deployment topology and failure excluded. A reference design can be internally consistent yet infeasible if a selected financial service or security feature lacks support in the intended region. The inability to inspect provider-internal power and network dependencies further limits independently calculated service availability.

Finally, literature rarely integrates national localisation, financial transaction correctness and power-system reliability into one measurable architecture. Recent work on cloud-provider viability demonstrates that sustainability, security and resilience are competing selection criteria [30]. However, multicriteria rankings alone cannot show how a particular institution performs during a prolonged Saudi-region outage. The crucial research gap is an experimentally validated, regulator-compatible recovery protocol with transparent failure taxonomy, common-cause assumptions and verifiable financial records.

11 Future research directions

First, researchers should establish anonymised, reproducible financial transaction benchmarks that report business-level correctness alongside application latency and recovery time. Tests should cover account transfers, card authorisation, transaction reversal and reconciliation with deterministic fault injection. Second, future studies should evaluate policy-constrained recovery automation with formal state-machine verification, explicit fencing and approval gates. Rather than optimise switching speed alone, reward functions should penalise inconsistent settlement, lost audit evidence and unsafe propagation of corrupted state.

Third, empirical work should measure recovery under combined regional network failure, electrical supply disturbance and cooling degradation, using confidential provider evidence where access permits. Scenarios should account for realistic Saudi weather, telecom diversity and dependence on standby capacity. Fourth, research should explore confidential but independently auditable evidence-sharing between banks, hyperscalers and supervisors, avoiding disclosure of sensitive topology. Fifth, lifecycle energy assessment should compare the electrical cost of idle standby capacity with the disruption exposure it mitigates. Designs must balance infrastructure efficiency with business-critical availability rather than assume these objectives always reinforce one another.

Finally, comparative experiments should test whether domestically separated regions or carefully governed hybrid configurations offer better validated recoverability for different financial-service tiers. Studies should make failure assumptions, regulatory constraints and costs explicit. A useful output would be a Saudi-specific resilience evidence schema covering technical tests, business continuity, data integrity, sovereignty and energy-system dependency. This would translate broad Vision 2030 aspirations into repeatable assurance practices without treating vendor certifications as proof of transaction resilience.

12 Conclusion

Resilient cloud computing for Saudi financial systems requires more than replicated machines and automatic routing. A defensible OCI architecture must distinguish the boundaries of fault domains, availability domains and regions; protect single-writer financial integrity; maintain isolated clean recovery copies; and orchestrate switching against measurable business objectives. The analysis finds that shared identity, configuration, electrical supply and payment dependencies can defeat otherwise convincing redundant designs. Saudi supervisory controls and domestic data handling should be built into the architecture from inception, while energy and cooling dependencies deserve explicit assurance alongside cybersecurity.

The proposed framework contributes an end-to-end reference design, a decision-controlled failover sequence and auditable evaluation criteria. It is a critical synthesis, not a report of operational trials, and its recommendations require region-specific service checks and supervised testing before implementation. Progress under Vision 2030 will be better served by reproducible evidence of safe transaction recovery than by unqualified availability claims. Future research should connect fault injection, financial reconciliation, regulatory oversight and the Kingdom's physical infrastructure in realistic, reviewable experiments.

References

[1] Cheng, M., Qu, Y., Jiang, C. and Zhao, C. (2022). Is cloud computing the digital solution to the future of banking?. *Journal of Financial Stability*, 63, 101073. https://doi.org/10.1016/j.jfs.2022.101073

[2] International Monetary Fund (2024). Saudi Arabia: Financial Sector Assessment Program—Financial System Stability Assessment. *IMF Country Report* 2024/281. https://doi.org/10.5089/9798400288074.002

[3] Kirti, M., Maurya, A.K. and Yadav, R.S. (2024). Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions. *Concurrency and Computation: Practice and Experience*, 36(13), e8081. https://doi.org/10.1002/cpe.8081

[4] Rouholamini, S.R., Mirabi, M., Farazkish, R. and Sahafi, A. (2024). Proactive self-healing techniques for cloud computing: A systematic review. *Concurrency and Computation: Practice and Experience*, 36(24), e8246. https://doi.org/10.1002/cpe.8246

[5] Rehman, A.U., Aguiar, R.L. and Barraca, J.P. (2022). Fault-Tolerance in the Scope of Cloud Computing. *IEEE Access*, 10, 63422–63441. https://doi.org/10.1109/ACCESS.2022.3182211

[6] Shahid, M.A., Islam, N., Alam, M.M., Mazliham, M. and Musa, S. (2021). Towards Resilient Method: An exhaustive survey of fault tolerance methods in the cloud computing environment. *Computer Science Review*, 40, 100398. https://doi.org/10.1016/j.cosrev.2021.100398

[7] Kumari, P. and Kaur, P. (2021). A survey of fault tolerance in cloud computing. *Journal of King Saud University – Computer and Information Sciences*, 33(10), 1159–1176. https://doi.org/10.1016/j.jksuci.2018.09.021

[8] Bibartiu, O., Dürr, F., Rothermel, K., Ottenwälder, B. and Grau, A. (2024). Availability analysis of redundant and replicated cloud services with Bayesian networks. *Quality and Reliability Engineering International*, 40(1), 561–584. https://doi.org/10.1002/qre.3414

[9] Alelyani, A., Datta, A. and Hassan, M. (2024). Optimizing Cloud Performance: A Microservice Scheduling Strategy for Enhanced Fault-Tolerance, Reduced Network Traffic, and Lower Latency. *IEEE Access*, 12, 35135–35153. https://doi.org/10.1109/ACCESS.2024.3373316

[10] Chen, H., Chen, P., Yu, G., Li, X. and He, Z. (2024). MicroFI: Non-Intrusive and Prioritized Request-Level Fault Injection for Microservice Applications. *IEEE Transactions on Dependable and Secure Computing*, 21(5), 4921–4938. https://doi.org/10.1109/TDSC.2024.3363902

[11] Aderaldo, C., Costa, T.M., Vasconcelos, D.M., Mendonça, N.C., Cámara, J. and Garlan, D. (2024). A declarative approach and benchmark tool for controlled evaluation of microservice resiliency patterns. *Software: Practice and Experience*. https://doi.org/10.1002/spe.3368

[12] Frank, S., Wagner, L., Hakamian, A., Straesser, M. and van Hoorn, A. (2022). MiSim: A Simulator for Resilience Assessment of Microservice-Based Architectures. *Proceedings of the International Conference on Software Quality*, Reliability and Security, 1014–1025. https://doi.org/10.1109/QRS57517.2022.00105

[13] Dong, H., Zhang, C., Li, G. and Zhang, H. (2024). Cloud-Native Databases: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(12), 7772–7791. https://doi.org/10.1109/TKDE.2024.3397508

[14] Lu, Y., Yu, X., Cao, L. and Madden, S. (2021). Epoch-based commit and replication in distributed OLTP databases. *Proceedings of the VLDB Endowment*, 14(5), 743–756. https://doi.org/10.14778/3446095.3446098

[15] Ferreira, S., Mendonça, J. and Andrade, E. (2025). Experimental performance analysis of data consistency levels in NoSQL databases. *Software: Practice and Experience*, 55(6), 1059–1070. https://doi.org/10.1002/spe.3412

[16] Ferreira, S. et al. (2025). Benchmarking consistency levels of cloud-distributed NoSQL databases using YCSB. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3558923

[17] National Institute of Standards and Technology (2020). Data Integrity: Recovering from Ransomware and Other Destructive Events. *NIST Special Publication* 1800-11. https://doi.org/10.6028/NIST.SP.1800-11

[18] National Institute of Standards and Technology (2021). Attribute-based Access Control for Microservices-based Applications Using a Service Mesh. *NIST Special Publication* 800-204B. https://doi.org/10.6028/NIST.SP.800-204B

[19] National Institute of Standards and Technology (2022). Implementation of DevSecOps for a Microservices-based Application with Service Mesh. *NIST Special Publication* 800-204C. https://doi.org/10.6028/NIST.SP.800-204C

[20] Shin, D., Kim, J., Pawana, I.W.A.J. and You, I. (2025). Enhancing cloud-native DevSecOps: A Zero Trust approach for the financial sector. *Computer Standards & Interfaces*. https://doi.org/10.1016/j.csi.2025.103975

[21] Baker, T. and Shortland, A. (2023). Insurance and enterprise: cyber insurance for ransomware. *The Geneva Papers on Risk and Insurance – Issues and Practice*, 48, 275–299. https://doi.org/10.1057/s41288-022-00281-7

[22] Stamenkov, G. (2024). Cloud service models, business continuity and disaster recovery plans, and responsibilities. *International Journal of Organizational Analysis*, 33(3), 437–451 (2025 issue; 2024 online). https://doi.org/10.1108/IJOA-12-2023-4127

[23] Kun, E. (2024). Challenges in regulating cloud service providers in EU financial regulation: From operational to systemic risks, and examining challenges of the new oversight regime for critical cloud service providers under the Digital Operational Resilience Act. *Computer Law & Security Review*, 52, 105931. https://doi.org/10.1016/j.clsr.2023.105931

[24] Parchimowicz, K. (2024). Do not get lost in the cloud: how EU financial institutions could avoid problems with cloud services arising under DORA. *Law*, Innovation and Technology, 16(2). https://doi.org/10.1080/17579961.2024.2392935

[25] Ahmed, K.M.U., Bollen, M.H.J. and Alvarez, M. (2021). A Review of Data Centers Energy Consumption and Reliability Modeling. *IEEE Access*, 9, 152536–152563. https://doi.org/10.1109/ACCESS.2021.3125092

[26] Isazadeh, A., Ziviani, D. and Claridge, D.E. (2023). Global trends, performance metrics, and energy reduction measures in datacom facilities. *Renewable and Sustainable Energy Reviews*, 174, 113149. https://doi.org/10.1016/j.rser.2023.113149

[27] Yu, Y., Shan, K., Tang, H. and Wang, S. (2025). Reliability and economic impacts of utilizing battery energy storage in data centers for energy flexibility services in smart grids. *Energy Conversion and Management*, article 119951. https://doi.org/10.1016/j.enconman.2025.119951

[28] Zhang, S., Gao, J., Qian, H., Yu, J., Luo, S. and Guo, Q. (2024). Power system operational reliability assessment based on the data center energy consumption elastic space. *Frontiers in Energy Research*, 12, 1370861. https://doi.org/10.3389/fenrg.2024.1370861

[29] Wang, Z., Wang, Y., Ji, H., Hasanien, H.M., Zhao, J., Yu, L., He, J., Yu, H. and Li, P. (2024). Distributionally robust planning for data center park considering operational economy and reliability. *Energy*, 290, 130185. https://doi.org/10.1016/j.energy.2023.130185

[30] Amoujavadi, S. and Nemati, A. (2024). Developing sustainability, resiliency, agility, and security criteria for cloud service providers' viability assessment: A comprehensive hierarchical structure. *Sustainable Futures*, 7, 100219. https://doi.org/10.1016/j.sftr.2024.100219