What Happens When a Cloud Goes Dark? Designing Cross-Cloud Disaster Recovery Between AWS and Azure for Saudi Government SAP Systems.

Abstract

Saudi government entities are moving finance, payroll and procurement systems built on SAP into hyperscale clouds, and both major providers are opening regions inside the Kingdom. Placing a mission-critical system with one provider, however, concentrates risk: a region-wide failure, a control-plane fault or a contractual dispute could leave the entity without its core administration for days. This paper asks how a recovery capability that spans two providers, with Amazon Web Services and Microsoft Azure as the case, should be designed for such systems. Following a design science approach, it draws on thirty journal articles from 2020 to 2025 on cloud resilience, availability modelling, replication, multi-cloud interoperability and continuity management, together with national cybersecurity controls and the providers' announcements of in-Kingdom regions. The resulting artefact is a tiered reference architecture that keeps high availability inside the primary provider and places recovery in the second, using database-level asynchronous replication, independent backup copies, code-defined rebuilds of the application tier and provider-neutral identity, naming and monitoring. Three recovery tiers are matched to system criticality, and a responsibility matrix and drill cycle complete the design. An invented comparison of the tiers illustrates the trade-off between recovery speed and standing cost. The evidence supports redundancy and replication as means to raise availability, is mixed on the cost and complexity of multi-cloud operation and remains limited for enterprise resource planning systems recovered across providers. Everything rests on published sources; nothing comes from primary or proprietary data.

1. Introduction

Cloud platforms rarely fail, but they do fail, and when they fail the effects are concentrated. A fault in a provider's network, storage layer or management plane can take down many services in a region at once, and customers who built every layer of their systems with that provider have nowhere to go. For a government entity, the systems most exposed are often the least visible to

UNDER PEER REVIEW IN IJAR

citizens and the most important to the state: the ERP platforms that pay salaries, settle supplier invoices and record the budget. Many of these run on SAP software.

The question is becoming practical in Saudi Arabia. Microsoft has announced that its Saudi Arabia East region, with three availability zones in the Eastern Province, would be available to customers in November 2026, and Amazon Web Services has said that its first region in the Kingdom is on track for December 2026 (Table 1, D3, D4). For the first time, an entity that must keep data in the Kingdom could, in principle, place a primary system with one hyperscale provider and a recovery copy with another without sending data abroad. National controls already require entities to plan for disaster recovery and to test their backups (D1, D2). What they do not provide is a design.

Academic work on cloud resilience is broad but rarely joins up the pieces an entity needs. Two studies anchor this paper. Martinez et al. (2022) analysed the challenges that cloud infrastructure poses for application resilience and presented a microservices platform with mechanisms that limited service interruptions and kept response times acceptable when infrastructure failed. Campêlo et al. (2020) surveyed how geo-replicated cloud storage keeps its replicas consistent, distinguishing fixed, configurable and monitored approaches. One addresses how applications ride out failure; the other addresses the price of keeping distant copies of data in step. A cross-cloud recovery design for an ERP system must solve both problems at once, for a monolithic, stateful application rather than a set of microservices.

This study aims to design a cross-cloud disaster recovery architecture for SAP systems of Saudi government entities, using two hyperscale providers as the case. Its objectives are to identify the failure scenarios and requirements that such an architecture must meet, to evaluate the mechanisms that the literature offers, to assemble those mechanisms into a tiered reference design with governance and to illustrate the cost and recovery trade-offs between tiers. Three research questions follow. RQ1: which failure scenarios and requirements should a cross-cloud recovery design for government SAP systems address? RQ2: which architectural mechanisms meet those requirements, and what support does the evidence give them? RQ3: how should those mechanisms be combined, governed and tested in a design suitable for Saudi entities?

UNDER PEER REVIEW IN IJAP

The paper continues with background on resilience concepts, multi-cloud practice and the national setting. After the method come the findings, then the proposed architecture with an invented tier comparison, and finally the discussion and conclusion.

2. Background

2.1 Continuity, recovery and resilience

The vocabulary is contested. Sawalha (2021) showed that business continuity planning and disaster recovery planning are often treated as the same thing, although the first concerns keeping essential functions going and the second concerns restoring technology after disruption. Stamenkov (2022) reviewed the many plans organisations are asked to maintain, from critical infrastructure protection to continuity of operations, and proposed a layered model that places each in a coherent hierarchy. Steen et al. (2024) went further, arguing that standard continuity management assumes systems can be decomposed and predicted, and that genuine uncertainty calls for adaptive capacity rather than plans alone. For this paper, disaster recovery means the technical ability to bring an SAP system back into service at another location within agreed limits for lost data and downtime, nested inside the entity's wider continuity arrangements.

Two measures express those limits. The recovery point objective (RPO) is the maximum tolerable data loss, measured as time; the recovery time objective (RTO) is the maximum tolerable time to restore service. Mendonça et al. (2020) combined experiments on real recovery solutions with analytic models of availability, both objectives, downtime and cost, and showed that suitable recovery solutions can raise availability while containing cost, with a sensitivity analysis identifying the parameters that most affect each objective. That pairing of measurement and modelling is the logic followed here.

2.2 Why one provider is a single point of failure

Resilience research has catalogued techniques at every layer of a cloud system. Welsh and Benkhelifa (2021) arranged resilience techniques by the layers of both established and newer forms of cloud computing and assessed how far each actor can shape resilience at each layer. Its implication for a customer is sobering: many failure modes sit in layers that only the provider controls. Architectural surveys for cloud, fog and edge systems reached a similar view and highlighted trust between components as a resilience issue in its own right (Prokhorenko & Ali

UNDER PEER REVIEW IN IJAR

Babar, 2020). Fault-tolerance research distinguishes proactive approaches, which act on predicted failures, from reactive ones, which recover after the event (Rehman et al., 2022).

Redundancy inside one provider addresses many failures but not all. Availability models of redundant private clouds showed that redundancy raises availability and cuts downtime, with a cost that must be weighed scenario by scenario (Faraji Shoyari et al., 2021). Bayesian network models of large replicated services added cascading and common-cause failures from shared infrastructure and networks (Bibartiu et al., 2024), which are precisely the failures that defeat redundancy within a single provider. Insurance research treats provider outages as a financial risk large enough to warrant catastrophe bonds, modelling outage frequency and loss severity explicitly (Mastroeni et al., 2022). The common thread is that correlated failures, not isolated ones, determine the worst case.

Multi-cloud designs promise independence but impose costs. Using several providers can avoid lock-in, but it adds management overhead and a trade-off between cost and availability (Hari Kumar & Mala, 2022). Portability and interoperability remain hampered by proprietary interfaces and limited standardisation (Ramalingam & Mohan, 2021), and a recent review of multi-cloud integration found consistent service delivery across providers still difficult in practice (Singh et al., 2025).

SAP systems sharpen these difficulties. An ERP installation is not a single server but a chain of components: the database, the central services that manage locks and messages, several application servers, file shares for interfaces and print, and connections to banks, tax portals, national identity services and other government platforms. Each link must be present at the recovery site, configured for its new location and reachable by users. Because the database holds the system of record, the consistency of its copy decides whether recovery is possible at all, while the other components decide how long it takes.

2.3 The national setting

The Essential Cybersecurity Controls require entities to define, implement and periodically review requirements for backup and recovery, covering critical assets, quick recovery after incidents and periodic testing of restores, and to include cybersecurity in business continuity management, including disaster recovery plans (Table 1, D1). The Cloud Cybersecurity Controls

UNDER PEER REVIEW IN IJAP

extend those duties to cloud tenants, asking them to develop and implement secure disaster recovery and continuity procedures for cloud computing, and since a 2025 update they refer localisation to the national data office (D2). The provider announcements establish that in-Kingdom regions from both providers are imminent but, at the time of writing, that one was in its announced availability month and the other was scheduled for the following month (D3, D4). Any design must therefore treat region availability and service catalogues as parameters to be confirmed.

Table 1. Documentary sources consulted (not counted among the thirty reviewed articles)

Code	Document	Issuer and year	Provisions relevant to this study
D1	Essential Cybersecurity Controls (ECC-2:2024)	National Cybersecurity Authority, 2024	Subdomain 2-9 on backup and recovery management: scope covering critical assets, quick recovery after incidents and periodic testing of restores; subdomain 3-1 on cybersecurity resilience in business continuity management, including disaster recovery plans, with periodic review
D2	Cloud Cybersecurity Controls (CCC-2:2024)	National Cybersecurity Authority, 2024 (updated 2025)	Tenants must develop and implement secure disaster recovery and continuity procedures for cloud computing (3-1-T-1-1); providers must secure tenant backups (2-8-P-1-1); localisation referred to the National Data Management Office
D3	Announcement of the Saudi Arabia East data centre region	Microsoft, 31 August 2026	Region with three availability zones in the Eastern Province, announced as available to customers in November 2026 for hosting eligible workloads and data locally
D4	Announcement of the first infrastructure region in the Kingdom	Amazon Web Services, 31 August 2026	First region in the Kingdom stated to be on track to launch in December 2026

3. Method

3.1 Design science process

The output sought is an artefact, a reference architecture, so the work follows a design science process. The problem was framed from the concentration risk of single-provider hosting and the national requirements for recovery. Solution objectives were set as bounded data loss and downtime, independence from any single provider's failure domain, compliance with national controls, affordable standing cost and testability. Its design drew on the coded evidence and was then demonstrated through an invented comparison of recovery tiers. Evaluation is analytical at this stage, checking the design against the objectives and against the failure scenarios identified in the findings.

UNDER PEER REVIEW IN IJAR

3.2 Evidence base

Literature for this study came from an openly available scholarly index, consulted in November 2026; for every candidate, the DOI, year of publication and any correction or retraction notice were then checked in the publishers' own registration data. Searches combined disaster recovery, business continuity, cloud resilience, fault tolerance, availability modelling, backup, geo-replication, consistency, multi-cloud, interoperability, vendor lock-in and failure prediction. To qualify, an item had to be a refereed journal paper dated 2020 to 2025 and relevant to recovering or protecting stateful systems in the cloud. Preprints, conference proceedings and records with broken identifiers fell outside scope. The final set holds thirty articles. Martinez et al. (2022) and Campêlo et al. (2020) were picked before the others as anchors, since they address application resilience under infrastructure failure and consistency of distant replicas respectively.

3.3 Synthesis

Each article was coded for the failure it addresses, the mechanism proposed, the layer at which it acts, the evaluation approach and any reported effect on availability, data loss, recovery time or cost. Mechanisms were grouped by layer and mapped to requirements, and each requirement was checked to make sure at least one mechanism with supporting evidence addressed it. Where none did, the gap was noted as a limitation instead of being filled by assumption. A rating of supportive meant that several measured or modelled results agreed; mixed meant results hinged on setting or cost assumptions; limited meant few studies, or none dealing with stateful enterprise systems. No primary data were gathered, and every value in the tier comparison is invented for illustration.

4. Findings

4.1 Failure scenarios and requirements (RQ1)

Five scenarios define the design space. The first is the loss of a single host, zone or component, which the primary provider's own redundancy should absorb without invoking disaster recovery. The second is the loss of a whole region through power, network or facility failure. The third is a provider-wide fault in shared services, such as identity, networking or the management plane, that affects several regions at once; correlated failures of this kind defeat redundancy inside one provider (Bibartiu et al., 2024). The fourth is logical damage, for example ransomware,

UNDER PEER REVIEW IN IJAR

corruption or an erroneous mass update, which replication copies faithfully to every replica. The fifth is non-technical loss of access, such as a contractual or legal dispute, which cuts the entity off from an otherwise working platform.

From these scenarios, six requirements follow. Recovery objectives must be explicit for each system and agreed with the business owners. The recovery site must not share failure domains with the primary, including identity, naming and monitoring. Point-in-time copies must be protected against logical damage, which argues for immutable or isolated backups in addition to replication (Maher & Nasr, 2021). Data must stay within permitted locations (Table 1, D2). Responsibilities must be clear between the entity, its providers and its integrator (Stamenkov, 2025). And the whole arrangement must be tested regularly (D1).

A further requirement is easy to miss: the recovery site must also reach everything the ERP system depends on. Interfaces to payment networks, government service buses and document management systems, as well as certificates, encryption keys and allow-listed addresses held by partners, often assume a fixed network location. If these dependencies are not reproduced or redirected, a technically successful database failover still leaves the entity unable to pay or post. Recovery design therefore has to begin with an inventory of dependencies, not with the database alone.

4.2 Mechanisms and their evidence (RQ2)

Mechanisms fall into five groups (Table 2). The first is in-provider high availability. Malhotra et al. (2023) proposed a cloud-native template for enterprise applications and evaluated automatic database failover for near-zero downtime during planned maintenance and outages, noting that redundancy at the database layer is harder to achieve than at the application layer. Availability-aware placement of virtual machines, which accounts for both hardware availability and the behaviour of co-located tenants, can further protect service levels within a provider (Liu et al., 2024). These measures handle the first scenario and should remain inside the primary provider, where latency allows synchronous replication.

The second group is cross-site data replication. Campêlo et al. (2020) showed that geo-replicated stores must choose between fixed, configurable and monitored consistency, and that strong consistency over long distances is expensive. Hamrouni et al. (2023) reviewed replication

UNDER PEER REVIEW IN IJAP

strategies for single and interconnected clouds and found that few of them consider service agreements, cost and data correlation together. Work on multi-cloud consistency protocols proposed middleware that orders writes globally across providers (Kozina et al., 2024) and adapted write latency to network conditions, cutting latency by 13% in one scenario (Kozina et al., 2025). Multi-master designs can keep strong consistency across regions with light coordination (Zhou et al., 2023), but an ERP database normally has a single writer. For a primary and recovery site in different providers, asynchronous replication with a bounded lag is the realistic choice, and the lag defines the recovery point.

The third group is backup and data placement. Secure multi-cloud backup that splits encrypted data across providers improved reliability, security and privacy (Maher & Nasr, 2021), and adaptive placement that predicts access frequency can lower storage cost while keeping availability (Wang et al., 2020). Erasure-coded placement across providers, tuned by optimisation, can balance cost against availability (Hari Kumar & Mala, 2022). Cloud backup and recovery studies stress security of copies in transit and at rest (Swagatika & Panda, 2022), which national controls also require of providers (Table 1, D2).

The fourth group concerns compute capacity for recovery. Keeping a full duplicate environment running is costly. Robust optimisation of shared backup capacity, which protects virtual machines against multiple host failures, saved about a third of total capacity in the cases examined (He et al., 2022). Across providers, the equivalent is to keep only the database replica warm and to rebuild application servers from code when needed. Rebuilding depends on portability, which proprietary interfaces limit (Ramalingam & Mohan, 2021; Singh et al., 2025).

The fifth group is detection and operations. Comparisons of classical and deep learning models on public cloud traces showed that job and task failures can be predicted with useful accuracy (Tengku Asmawi et al., 2022). Experience with node-failure prediction on an ultra-large platform showed that such tools must also be trustworthy, interpretable, maintainable and evaluated in context (Li et al., 2020). Root cause analysis that combines traces, metrics and logs outperformed earlier methods on public datasets (Han et al., 2024). These techniques shorten the time to decide that a failover is needed, which is often a larger share of real recovery time than the technical switch itself.

UNDER PEER REVIEW IN IJAR

Networking links all five groups. Continuous replication between providers needs a private, encrypted path with enough bandwidth to keep lag inside the recovery point during peak posting periods, such as month-end close or payroll runs. Data leaving a provider usually carries transfer charges, and those charges grow with change volume rather than database size. Designers should therefore estimate change rates, not only storage, before choosing between continuous replication and periodic log shipping, and should size the interconnect for the busiest days of the financial calendar.

Table 2. Mechanisms for cross-cloud recovery, requirements served and strength of evidence

Mechanism group	Requirement served	Placement in the design	Key sources	Evidence
In-provider high availability	Absorb host and zone failures	Primary provider, synchronous	Malhotra et al. (2023); Liu et al. (2024)	Supportive
Cross-site data replication	Bounded data loss after regional or provider failure	Asynchronous to second provider	Campêlo et al. (2020); Hamrouni et al. (2023); Kozina et al. (2024)	Supportive
Isolated backup and placement	Protection against logical damage	Object storage in second provider	Maher & Nasr (2021); Wang et al. (2020)	Mixed
Recovery compute and portability	Affordable standing cost	Rebuild from code; small warm replica	He et al. (2022); Ramalingam & Mohan (2021)	Mixed
Detection and operations	Shorter time to decide and recover	Independent of both providers	Tengku Asmawi et al. (2022); Han et al. (2024)	Limited

4.3 How firm is the evidence?

Support is strongest for redundancy and replication as means to raise availability, with consistent results from analytic models, simulations and testbeds. It is mixed on the cost and operational burden of multi-cloud designs, because savings and overheads depend heavily on workload and pricing assumptions. It is limited for the specific case of an ERP system recovered across two providers: no reviewed study measured recovery objectives for such a system, and the continuity literature that addresses government is largely conceptual. Studies of continuity in practice, such as cloud-enabled continuity management in South African ports (Ochara et al., 2022), suggest that organisational fit matters as much as technical design. Few of the reviewed studies report recovery drills, real outages or the cost of operating a second provider over time, so claims about the affordability of cross-cloud designs rest more on models than on observation.

UNDER PEER REVIEW IN IJAR

5. A Cross-Cloud Recovery Architecture

5.1 Reference design

The architecture separates two jobs (Figure 1). High availability stays inside the primary provider: the SAP database runs with a synchronous standby in a second availability zone, and application servers are spread across zones, so that host and zone failures are absorbed locally (Malhotra et al., 2023). Disaster recovery goes to the second provider: an asynchronous database replica runs in that provider's in-Kingdom region, receiving changes over a private interconnect, while application servers are defined as code and built only when needed. Either provider can play either role; the choice of primary depends on where each entity's other services run and which region and services are available when the design is implemented (Table 1, D3, D4).

Four supporting elements prevent hidden dependencies. Backups are written to object storage in the recovery provider with immutability or isolation, so that logical damage in the primary cannot reach them. Identity for administrators and integration accounts is federated from a source that remains available if the primary fails. Naming and traffic steering are controlled outside both providers or duplicated in each. Monitoring and alerting run independently of the primary, so that the entity learns of a failure from its own tools. Together they keep the recovery environment out of the primary's failure domains.

A failover follows a fixed sequence recorded in the runbook. Detection and confirmation come first: monitoring raises the alarm and a named decision-maker, supported by root cause information, declares a disaster. The database replica is then promoted to primary and checked for its last applied change, which fixes the actual data loss. Application servers and central services are built from code in the recovery region, interfaces are redirected and partners are notified where addresses or certificates change. Users are steered to the new site only after functional checks on posting, payroll and procurement. Finally, the entity records the timeline so that achieved recovery time can be compared with the target.

UNDER PEER REVIEW IN IJAR

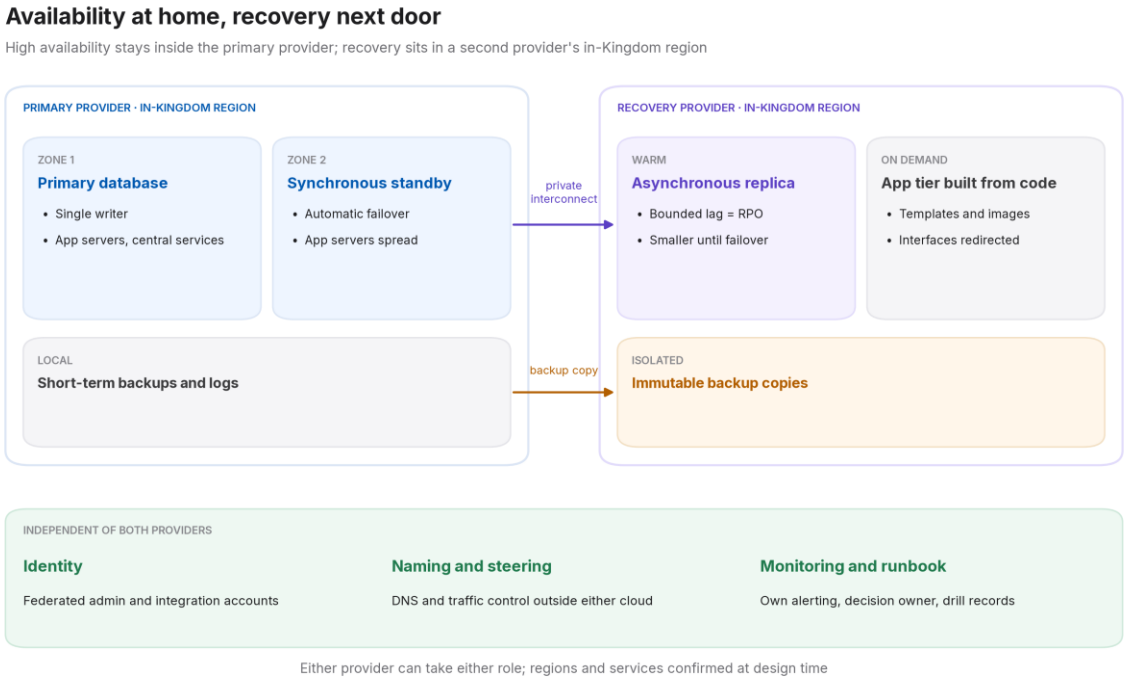


Figure 1. Cross-cloud recovery reference architecture for a government SAP system

5.2 Recovery tiers and governance

Not every system justifies the same investment. The design therefore offers three tiers. Tier 1, a warm standby, keeps a running database replica and a minimal application footprint in the second provider and targets recovery within hours with data loss of minutes. Tier 2, a pilot light, keeps a small replica or frequent log shipping and rebuilds most capacity on demand, accepting longer recovery. Tier 3 relies on backups copied to the second provider and full rebuild, with recovery measured in days. Payroll, financial posting and procurement for large entities would normally sit in Tier 1, reporting systems in Tier 2 and development or archive systems in Tier 3.

Governance makes the design work. Stamenkov (2025) proposed responsibility matrices for continuity and recovery across cloud service models and recovery service models, and showed that accountability shifts with the model chosen. In a cross-cloud design the entity remains accountable for the recovery plan and its decisions, each provider is responsible for its platform, and the integrator is usually responsible for executing the runbook. Recovery should be tested at least annually for Tier 1 systems with a full failover and failback, and more often with partial

UNDER PEER REVIEW

tests of replication lag, backup restore and application rebuild (Table 1, D1). Each test should record achieved recovery point and time against target.

Failback deserves the same care as failover. Once the primary provider recovers, the entity must decide whether to return, re-establish replication in the reverse direction and schedule a controlled switch back, usually in a maintenance window. Running for weeks in the recovery region is acceptable only if that region has been sized and secured for production; the design should state how long the entity can remain there and what capacity it will add if the outage is prolonged.

Security at the recovery site must match production from the first day. A dormant environment is easily neglected: patches lag, accounts accumulate and logging is switched off to save cost. National controls require tenants to implement secure recovery procedures and providers to protect tenant backups against damage, alteration and unauthorised access (Table 1, D2). In practice this means the same hardening baseline, privileged access controls, key management and log collection in both providers, with the recovery environment included in vulnerability scanning and compliance assessment even while it carries no users.

5.3 Illustrative comparison of tiers

To show the trade-off, the three tiers were compared with invented values (Figure 2). Suppose the warm standby restores service in 4 hours and loses at most 15 minutes of data, at a standing cost index of 45, where 100 stands for a full duplicate environment. The pilot light is assumed to need 12 hours, lose up to an hour and cost 20 on the same index, while backup-only recovery takes 72 hours, loses up to a day and costs 6. None of these figures is measured; they serve only to make the shape of the trade-off visible.

The example shows two things. First, the cost of recovery capacity falls steeply as recovery time is relaxed, so tier assignment by criticality matters more than any single technical choice. Second, even the most expensive tier costs well under a full duplicate, because the application tier is rebuilt rather than kept running, and the database replica can run on a smaller instance until failover. An entity choosing tiers should replace these invented numbers with its own measured objectives and quoted prices, and should note that recovery time includes the time to decide, which detection and root cause tools can shorten (Han et al., 2024).

UNDER PEER REVIEW IN IJAP

Tier choice can then be framed as a simple comparison. For each system, the entity estimates what an hour of downtime and an hour of lost transactions would cost, including penalties, manual rework and the effect on salaries or supplier payments, and multiplies by the downtime and data loss each tier allows in a plausible disaster. A higher tier is justified when the reduction in expected loss exceeds its additional standing cost over the planning period. Because the probability of a provider-wide failure is low and hard to estimate, the comparison is best run for a range of probabilities, and political or reputational consequences may justify a higher tier even when the arithmetic does not.

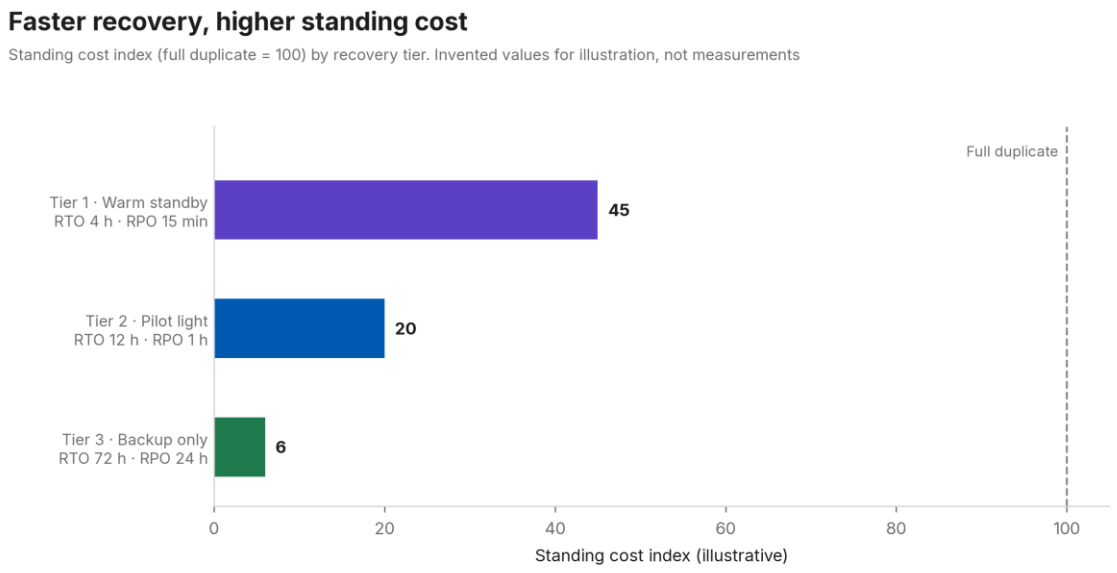


Figure 2. Illustrative recovery time and standing cost of three recovery tiers (invented values, not measurements of any real system)

6. Discussion

6.1 Contribution

Together, the two anchor studies set the two problems that the design joins. Martinez et al. (2022) showed how a platform can keep applications available through infrastructure failure, and Campêlo et al. (2020) showed the consistency choices that distant replicas force. The architecture brings these together for a stateful ERP system: it keeps strong consistency where latency allows, inside a provider, and accepts bounded asynchronous lag across providers,

UNDER PEER REVIEW - IJAR

turning that lag into an explicit recovery point. It also separates the problems that replication solves, physical and provider failure, from those it worsens, logical damage, and assigns the second to isolated backups.

The design also adds governance and testing to a literature that is mostly technical. Continuity research warns that plans built on an illusion of control fail in real crises (Steen et al., 2024). Regular failover drills with recorded outcomes are the practical answer: they turn assumed recovery objectives into measured ones.

The design can also be read alongside financial approaches to outage risk. Providers transfer part of their exposure through insurance and related instruments (Mastroeni et al., 2022), but a customer's compensation under a service agreement rarely covers the loss of core administration. Cross-cloud recovery is, in effect, self-insurance in kind: the entity pays a standing premium in recovery capacity so that its own services survive a failure that no service credit would repair.

6.2 Implications for Saudi entities

For government entities, the arrival of two in-Kingdom hyperscale regions changes what is feasible. The first implication is to classify systems by criticality and assign recovery tiers before contracts are signed, since recovery capacity, interconnects and egress charges need to be priced. The second is to keep recovery independent: identity, naming, monitoring and backups should not depend on the primary provider. The third is to confirm with the national data office that both regions are acceptable for each data class, and to check that the needed services and instance types exist in each region at launch (Table 1, D2, D3, D4).

For policymakers, shared services could lower the cost. A common interconnect between the two providers' Saudi regions, model contract clauses on recovery support and exit, and a national exercise programme for cross-cloud failover would let smaller entities benefit without each building its own capability.

Skills are the binding constraint. Running SAP well on one hyperscale platform is demanding; running it across two requires staff or partners fluent in both providers' networking, identity and storage services, as well as in the database's replication. Entities should budget for this dual capability explicitly, prefer tooling that works across providers and keep runbooks that a partner team unfamiliar with the system could follow under pressure. Failure prediction research also

UNDER PEER REVIEW IN IJAP

suggests that operational analytics are only useful when engineers trust and can interpret them (Li et al., 2020), which argues for building monitoring skills alongside the tools.

6.3 Limitations and future research

The study has limits. None of the thirty articles measures cross-provider recovery for an ERP system, so the architecture transfers mechanisms from neighbouring settings. The provider announcements describe regions that were only beginning to open when this paper was written, and service catalogues will change. The tier comparison uses invented values. Future research should implement the design in a test environment across the two in-Kingdom regions, measure achieved recovery objectives, replication lag and costs, and study how entities assign tiers and govern drills in practice. Comparative work with other states that require in-country hosting would show whether the design transfers beyond the Kingdom.

7. Conclusion

A government SAP system hosted with a single cloud provider inherits that provider's worst day. This paper has designed a cross-cloud recovery architecture for Saudi entities that keeps high availability inside the primary provider, places recovery in a second provider's in-Kingdom region, protects backups against logical damage, removes hidden dependencies on the primary and matches recovery tiers to system criticality. Built from thirty studies and four documentary sources and illustrated with invented values, it shows that cross-cloud recovery need not cost as much as a full duplicate if tiers are chosen with care. Its value will be proven only by drills that turn planned recovery objectives into measured ones.

UNDER PEER REVIEW IN IJAR

References

1. Bibartiu, O., Dürr, F., Rothermel, K., Ottenwälder, B., & 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

2. Campêlo, R. A., Casanova, M. A., Guedes, D. O., & Laender, A. H. F. (2020). A brief survey on replica consistency in cloud environments. Journal of Internet Services and Applications, 11(1), Article 1. https://doi.org/10.1186/s13174-020-0122-y

3. Faraji Shoyari, M., Ataie, E., Entezari-Maleki, R., & Movaghar, A. (2021). Availability modeling in redundant OpenStack private clouds. Software: Practice and Experience, 51(6), 1218–1241. https://doi.org/10.1002/spe.2953

4. Hamrouni, T., Mokadem, R., & Khelifa, A. (2023). Review on data replication strategies in single vs. interconnected cloud systems: Focus on data correlation-aware strategies. Concurrency and Computation: Practice and Experience, 35(22), Article e7758. https://doi.org/10.1002/cpe.7758

5. Han, Y., Du, Q., Huang, Y., Li, P., Shi, X., Wu, J., Fang, P., Tian, F., & He, C. (2024). Holistic root cause analysis for failures in cloud-native systems through observability data. IEEE Transactions on Services Computing, 17(6), 3789–3802. https://doi.org/10.1109/tsc.2024.3478759

6. Hari Kumar, P., & Mala, G. S. A. (2022). H2RUN: An efficient vendor lock-in solution for multi-cloud environment using horse herd Runge Kutta based data placement optimization. Transactions on Emerging Telecommunications Technologies, 33(9), Article e4541. https://doi.org/10.1002/ett.4541

7. He, F., Sato, T., Chatterjee, B. C., Kurimoto, T., Urushidani, S., & Oki, E. (2022). Robust optimization model for primary and backup resource allocation in cloud providers. IEEE Transactions on Cloud Computing, 10(4), 2920–2935. https://doi.org/10.1109/tcc.2021.3051018

8. Kozina, O. A., Panchenko, V. I., Kolomiitsev, O. V., Usik, V. V., Stratiienko, N. K., Safoshkina, L. V., & Kucherenko, Y. F. (2024). Data consistency protocol for multicloud

UNDER PEER REVIEW IN IJAR

systems. International Journal of Cloud Computing, 13(1), 42–61. https://doi.org/10.1504/ijcc.2024.136284

9. Kozina, O., Machado, J., Volk, M., Heiko, H., Panchenko, V., Kozin, M., & Ivanova, M. (2025). Opportunities for adapting data write latency in geo-distributed replicas of multicloud systems. Future Internet, 17(10), Article 442. https://doi.org/10.3390/fi17100442

10. Li, Y., Jiang, Z. M., Li, H., Hassan, A. E., He, C., Huang, R., Zeng, Z., Wang, M., & Chen, P. (2020). Predicting node failures in an ultra-large-scale cloud computing platform. ACM Transactions on Software Engineering and Methodology, 29(2), 1–24. https://doi.org/10.1145/3385187

11. Liu, J., Zhao, G., Xu, H., Yang, P., Wang, B., & Qiao, C. (2024). Toward a service availability-guaranteed cloud through VM placement. IEEE/ACM Transactions on Networking, 32(5), 3993–4008. https://doi.org/10.1109/tnet.2024.3401758

12. Maher, R., & Nasr, O. A. (2021). DropStore: A secure backup system using multi-cloud and fog computing. IEEE Access, 9, 71318–71327. https://doi.org/10.1109/access.2021.3078887

13. Malhotra, A., Elsayed, A., Torres, R., & Venkatraman, S. (2023). Evaluate solutions for achieving high availability or near zero downtime for cloud native enterprise applications. IEEE Access, 11, 85384–85394. https://doi.org/10.1109/access.2023.3303430

14. Martinez, H. F., Mondragon, O. H., Rubio, H. A., & Marquez, J. (2022). Computational and communication infrastructure challenges for resilient cloud services. Computers, 11(8), Article 118. https://doi.org/10.3390/computers11080118

15. Mastroeni, L., Mazzoccoli, A., & Naldi, M. (2022). Pricing cat bonds for cloud service failures. Journal of Risk and Financial Management, 15(10), Article 463. https://doi.org/10.3390/jrfm15100463

16. Mendonça, J., Lima, R., & Andrade, E. (2020). Evaluating and modelling solutions for disaster recovery. International Journal of Grid and Utility Computing, 11(5), Article 110055. https://doi.org/10.1504/ijguc.2020.110055

UNDER PEER REVIEW IN IJAR

17. Ochara, N. M., Kutame, F. N., & Kadyamatimba, A. (2022). Adoption of cloud computing in business continuity management for container terminal operations in South Africa. International Journal of Business Continuity and Risk Management, 12(2), Article 124834. https://doi.org/10.1504/ijbcrm.2022.124834

18. Prokhorenko, V., & Ali Babar, M. (2020). Architectural resilience in cloud, fog and edge systems: A survey. IEEE Access, 8, 28078–28095. https://doi.org/10.1109/access.2020.2971007

19. Ramalingam, C., & Mohan, P. (2021). Addressing semantics standards for cloud portability and interoperability in multi cloud environment. Symmetry, 13(2), Article 317. https://doi.org/10.3390/sym13020317

20. Rehman, A. U., Aguiar, R. L., & 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

21. Sawalha, I. H. (2021). Views on business continuity and disaster recovery. International Journal of Emergency Services, 10(3), 351–365. https://doi.org/10.1108/ijes-12-2020-0074

22. Singh, D., Kumar, M., Patel, M., & Singh, U. (2025). Interoperable cloud services for multi-cloud architectures: A review of approaches, challenges and solutions. IT Professional, 27(5), 82–89. https://doi.org/10.1109/mitp.2025.3591458

23. Stamenkov, G. (2022). Layered business continuity and disaster recovery model. Continuity & Resilience Review, 4(3), 267–279. https://doi.org/10.1108/crr-05-2022-0008

24. Stamenkov, G. (2025). Cloud service models, business continuity and disaster recovery plans, and responsibilities. International Journal of Organizational Analysis, 33(3), 437–451. https://doi.org/10.1108/ijoa-12-2023-4127

25. Steen, R., Haug, O. J., & Patriarca, R. (2024). Business continuity and resilience management: A conceptual framework. Journal of Contingencies and Crisis Management, 32(1), Article e12501. https://doi.org/10.1111/1468-5973.12501

UNDER PEER REVIEW IN IJAR

26. Swagatika, S., & Panda, N. (2022). Cloud-based backup and data recovery. Journal of Information and Optimization Sciences, 43(5), 923–932. https://doi.org/10.1080/02522667.2022.2091097

27. Tengku Asmawi, T. N., Ismail, A., & Shen, J. (2022). Cloud failure prediction based on traditional machine learning and deep learning. Journal of Cloud Computing, 11(1), Article 47. https://doi.org/10.1186/s13677-022-00327-0

28. Wang, P., Zhao, C., Wei, Y., Wang, D., & Zhang, Z. (2020). An adaptive data placement architecture in multicloud environments. Scientific Programming, 2020, 1–12. https://doi.org/10.1155/2020/1704258

29. Welsh, T., & Benkhelifa, E. (2021). On resilience in cloud computing. ACM Computing Surveys, 53(3), 1–36. https://doi.org/10.1145/3388922

30. Zhou, W., Peng, Q., Zhang, Z., Zhang, Y., Ren, Y., Li, S., Fu, G., Cui, Y., Li, Q., Wu, C., Han, S., Wang, S., Li, G., & Yu, G. (2023). GeoGauss: Strongly consistent and light-coordinated OLTP for geo-replicated SQL database. Proceedings of the ACM on Management of Data, 1(1), 1–27. https://doi.org/10.1145/3588916

UNDER PEER REVIEW IN IJAR