Computer Fraud and Security
ISSN (online): 1873-7056

Zero-Data-Loss Disaster Recovery for Hybrid Financial ERP Landscapes: An AI-Driven Approach

Rajasekhar Reddy Putta*[1]
Pondicherry University, India.

Abstract

Enterprise financial landscapes operating across cloud-native ERP platforms, centralized finance hubs, and legacy source systems face compounding disaster recovery challenges that conventional backup-and-restore strategies are fundamentally inadequate to address. The tight operational dependencies introduced by real-time replication architectures—where a centralized universal journal continuously ingests financial postings from multiple heterogeneous source systems—demand a Zero-Data-Loss posture that is technically enforced rather than aspirationally stated. Synchronous HANA System Replication, with all financial core nodes, achieves full durability by writing every committed transaction durably until the application acknowledges completion. This gives RPO = 0 and minute-level recovery times driven by cluster automation. STONITH fencing and cluster management automation. eliminate split-brain across the database tier and the ABAP central services layer. Enqueue Replication Server communication patterns implement application lock table persistence. Safe shutdowns, deterministic message tracking-aware replay, and the recovery-side interface framework replay are required to prove that the central journal contains exactly the source transactions and is sequentially consistent. Artificial Intelligence for observability tools aggregates telemetry across replication health, queue levels, and interface. error counts and network latency. These are then analyzed by continuous anomaly detection in order to detect ZDL risk conditions before they impact services. Runbooks informed by large language models provide policy guardrails against harmful failover automation decisions. Continuous post-recovery universal journal reconciliation produces auditable evidence of recovery completeness, transforming disaster recovery from a documented contingency into a continuously verified operational guarantee for financial systems of record.

Keywords: Zero-Data-Loss Recovery, HANA System Replication, Central Finance Disaster Recovery, AIOps Runbook Automation, Cluster-Governed Failover

1. Introduction

Enterprise IT manages cloud-native financial ERP systems, centralized finance hubs, and legacy financial ERP systems. The customary approach to business continuity of backup and restore is insufficient. These financial ERP systems are tightly coupled. If the financial ERP systems lose data for a short period, this can considerably weaken financial integrity, audit compliance, and regulatory compliance. Hybrid financial architectures combine many-to-many replication chains with a finance hub architecture, where many source systems are each captured in real-time, introducing dependencies that increase recovery complexity at each level of the architecture.

Central Finance can receive financial accounting and management accounting data from one or more source systems and centralize financial processes while legacy systems remain running in parallel. This architectural reality means that any DR strategy must account not only for the target platform itself but for the uninterrupted continuity of every upstream data feed that populates the universal journal—because the central ledger's integrity is directly contingent on the integrity of every replication channel flowing into it [1]. The practical consequence is that recovery sequencing must be choreographed simultaneously across multiple system boundaries, rather than treating each platform as an isolated recovery unit that can be restored independently of its integration dependencies.

Achieving a Zero-Data-Loss posture across such a heterogeneous landscape demands a disciplined combination of synchronous database replication, cluster-governed failover automation, and continuous AI-assisted verification. HANA System Replication provides the technical foundation, supporting multiple replication modes ranging from fully synchronous configurations that guarantee RPO = 0 to asynchronous modes that balance performance with geographic distribution—each suited to a specific tier of the recovery hierarchy [2]. Together, these capabilities form an architecture

where the most stringent controls are concentrated precisely where financial data integrity is, where every recovery action is observable, automatable, and auditably verifiable from end to end.

2. Zero-Data-Loss Requirements for the Financial Core

The data in these systems is also more critical for operations than in manufacturing, logistics, or human capital systems, where bounded data loss is often repairable by reprocessing or by human intervention/override. Financial postings, being the basis of statutory, regulatory, and audit reporting, do not tolerate missing and corrupted ledger records. In hybrid systems where a central finance platform aggregates from multiple source systems using a real-time change data capture route, this can lead to inconsistent ledger states, failed reconciliations, and broken audit trails that only become visible at the end of the month or when an external request for discrepancy investigation occurs.

Central Finance is specifically architected to serve as the single platform for group-wide financial steering, receiving transactional data from both SAP and non-SAP source systems and harmonizing it into a unified universal journal. This design creates a structural dependency where the integrity of the central ledger is only as reliable as the integrity of every replication feed that populates it. Importantly, Central Finance also supports configurations where financial data can be loaded without deep real-time integration in certain scenarios, but for organizations operating under ZDL mandates, the replication-driven path must be treated as the authoritative data flow—meaning its continuity, sequencing, and post-recovery validation are non-negotiable elements of the DR strategy [3]. When a DR event interrupts those feeds, recovery must restore not only the database but also the precise transactional state of every replication channel, queue, and message tracking identifier across the landscape.

This reality elevates Zero-Data-Loss from a desirable engineering property to a hard architectural requirement for the financial core. The universal journal and all data it contains are also treated as Tier-0, with an RPO of zero and recovery time objectives of minutes rather than hours. This is not just a policy but is enforced with synchronous replication modes that refuse transactions until the data is persisted on the secondary, cluster automation that drives any takeover without human intervention, and AI-based observability that checks every replication path to the financial core for signs of trouble. HANA System Replication, when configured in its synchronous modes, is the mechanism through which this RPO = 0 commitment is technically enforced—ensuring that at the moment of any primary failure, the secondary holds a complete and consistent copy of every committed transaction [4]. Intermediate integration tiers encompassing the replication engine, interface processing framework, and associated queues are classified as Tier-1, targeting near-zero or zero loss through safe deactivation procedures and deterministic replay, while peripheral systems fall under Tier-2, where business-defined tolerances govern a mix of backup and asynchronous replication strategies that are appropriate for workloads without financial reporting obligations.

Recovery Tier	Workload Scope	RPO Target	RTO Target	Enforcement Mechanism
Tier-0	Universal Journal, Financial	Zero	Minutes	Full-sync HSR, Cluster
Tier 1	Replication Engine, AIF	Near-Zero	Short	Safe Deactivation,
Tier-2	Peripheral, Non-Financial	Business-Defined	Business-Defined	Async Replication,

Table 1: Zero-Data-Loss Requirements for the Financial Core [3, 4]

3. Synchronous Database Replication Architecture

A key aspect of the Zero-Data-Loss option is synchronous HANA System Replication, in which the redo log entries are replicated from the primary system(s) to one or more secondary systems without data loss, which means that the secondary instance has every committed transaction durably saved before the application receives an acknowledgment of a commit. HANA System Replication supports four distinct operation modes—SYNC, SYNCMEM, ASYNC, and full-sync—each representing a different balance between write performance and data protection guarantee [5]. For Tier-0 financial workloads, the selection of the appropriate mode is not a performance optimization decision but a compliance requirement that must be enforced at the cluster configuration level to prevent inadvertent mode downgrades during maintenance windows or incident response actions that could silently weaken the ZDL guarantee.

In SYNC mode, the primary confirms a transaction commit to the application only after the redo log has been written to the secondary system's persistent memory. This provides strong protection against primary failure but retains a residual exposure: if the secondary experiences a simultaneous or immediately subsequent memory failure, the most recently committed transactions may not have reached persistent storage on the secondary. The full-sync operation mode

eliminates this residual exposure by holding the primary's commit acknowledgment until the redo log has been written to persistent disk storage on the secondary—providing the strongest available data durability guarantee at the cost of marginally higher write latency that is acceptable for financial workloads given the compliance imperative [5]. Full-sync is therefore the mandated configuration for Tier-0 financial workloads where RPO = 0 is a non-negotiable audit and regulatory requirement, and any planned deviation from this configuration must be subject to formal change control and risk acceptance by financial governance stakeholders.

HANA System Replication operates at the database service level, replicating all data and redo log information continuously to a secondary system that maintains a warm and current state ready for immediate takeover without requiring a full database restore from backup media. The secondary continuously receives and applies redo log entries, meaning that at any moment its data state is current to within the synchronization boundary defined by the selected replication mode, and takeover can be executed in minutes rather than hours under cluster automation [5]. For configurations where intra-region high availability and inter-region disaster recovery are both required, the multi-target replication topology builds on the two-target topology and implements a star architecture whereby a synchronous secondary in the same AZ provides local failover with RPO = 0 and minute-level RTO. An asynchronous tertiary site in a separate piece of geography provides geographic redundancy with a separately negotiated RPO tolerance [6]. Multi-target replication enables replication to multiple different secondaries. Using a single primary, the local HA secondary, and the remote DR secondary can be protected with no intermediate hops, avoiding the potential impacts of latency or additional points of failure in the protection chain.

HSR Operation Mode	Log Persistence on Secondary	RPO Guarantee	Recommended Tier	Key Trade-off
Full-Sync	Disk-persistent before commit	Zero	Tier-0 Financial Core	Highest durability, marginal write latency
SYNC	Memory-persistent before commit	Near-Zero	Tier-0 / Tier-1	Strong protection, residual memory-loss risk
SYNCMEM	Memory-buffered	Near-Zero	Tier 1	Balanced latency and protection
ASYNC	No wait on secondary	Business-Defined	Tier-2 / Remote DR	Lowest latency, no ZDL guarantee

Table 2: Synchronous Database Replication Architecture [5, 6]

4. Cluster Governance, ASCS/ERS, and Split-Brain Prevention

Synchronous database replication establishes the data protection guarantee, but cluster governance is what ensures that guarantee is reliably exercised when a failure event actually occurs. Without automated cluster management, even a perfectly synchronized secondary cannot deliver minute-level RTO, because manual takeover procedures introduce operator response latency, decision uncertainty, and execution error into the recovery sequence. More critically, without cluster-enforced fencing, a failure scenario can produce a split-brain condition where both the failed primary and the promoted secondary simultaneously believe themselves to be the active system and continue accepting writes—creating a divergent data state that is irreconcilable without data loss and permanently destroys the ZDL guarantee for the affected transaction range in a way that cannot be corrected without manual journal intervention and audit disclosure.

STONITH fencing is the mandatory technical control for split-brain prevention in any cluster-managed SAP landscape. Before any secondary promotion proceeds, the cluster management framework must obtain positive confirmation that the failed primary has been physically or logically isolated—through a hardware fencing action, a hyperscaler API call to stop or deallocate the failed virtual machine instance, or a network-level isolation mechanism that prevents the failed node from accepting or committing any further writes. Only after this fencing confirmation does the cluster proceed with secondary promotion, virtual IP reassignment, and application-tier resource migration in the correct sequence [7]. This sequencing discipline is non-negotiable for financial workloads where the cost of a split-brain divergence—in terms of data recovery effort, audit remediation, and regulatory exposure—vastly outweighs the marginal additional recovery time

introduced by the fencing confirmation step, and organizations must resist the operational temptation to disable or weaken fencing in the name of faster failover.

In the ABAP central services layer, the enqueue lock table continues to exist through failovers, since the enqueue server is hosted by the ASCS instance. It manages application-level locks across all user sessions, background job sessions, and dialog work processes of an SAP application landscape. When the ASCS instance fails without lock table replication in place, all active application locks are lost, and users reconnecting after failover encounter phantom lock conflicts, data inconsistencies, or aborted transactions that leave business objects in intermediate states requiring manual resolution [7]. High availability configurations for SAP on Azure use Pacemaker cluster management with Azure-native fencing agents, coordinating both the HANA System Replication takeover at the database tier and the ASCS/ERS failover at the application tier within a unified automation framework that enforces correct sequencing—database takeover completes before application servers attempt reconnection. Azure Load Balancer health probe integration ensures that virtual IP failover is detected and propagated rapidly across the application network layer [8]. The cluster continuously monitors resource health, executes configurable failure action policies, and maintains a detailed event log that serves as audit evidence for post-event review, demonstrating that every failover action was executed by automation according to pre-approved cluster policy rather than through ad-hoc manual intervention that would be difficult to reconstruct for compliance purposes.

Cluster Component	Function	Failure Risk Mitigated	Key Configuration Requirement
STONITH Fencing	Isolates failed primary before promotion	Split-brain, data divergence	Mandatory for all Tier-0 clusters
Pacemaker Resource Agent	Automates failover sequencing	Manual error, RTO overrun	Configured for HANA and ASCS/ERS
Enqueue Replication Server	Mirrors the ASCS lock table continuously	Lock table loss on ASCS failure	Active standby with cluster promotion
Virtual/Overlay IP	Routes application traffic post-failover	DNS propagation delay	Pre-configured on all cluster nodes

Table 3: Cluster Governance, ASCS/ERS, and Split-Brain Prevention [7, 8]

5. Replication Fabric Integrity and CFIN-Specific DR Procedures

Centralized finance replication introduces a dimension of DR complexity that is entirely absent from simpler single-ERP landscapes. The replication fabric operates as a continuous change data capture layer between source systems and the central finance target, maintaining a detailed tracking state that indexes every replicated document against its originating source transaction and its position in the source system's change log sequence. This tracking mechanism is the foundation of deterministic replay—the ability to resume replication after an interruption and guarantee that every source transaction is replicated exactly once, in the correct sequence, without duplication or omission, producing a central journal that is provably complete and consistent with the authoritative state of the source systems [9].

During a DR event, uncontrolled interruption of the replication engine risks queue corruption, duplicate universal journal postings, or tracking state gaps that break the deterministic replay guarantee and require costly manual reconciliation to resolve. Safe deactivation is the prescribed technical control: replication jobs are paused in an orderly sequence before any database takeover proceeds, active queue depths and tracking states are captured and recorded as the verified baseline for post-recovery replay validation, and the replication engine is brought to a clean stop that leaves its internal state consistent with the database state at the precise point of takeover [9]. Organizations that skip the safe deactivation step and allow the replication engine to be interrupted mid-batch routinely encounter post-recovery reconciliation gaps that require manual journal corrections—a process that is both operationally costly and audit-sensitive, often requiring external auditor notification and formal remediation documentation that extends the effective recovery timeline well beyond the technical RTO.

Post-recovery replication content validation is a step that organizations systematically underestimate in their DR planning. Version alignment between the replication transport component and the target system release is a prerequisite

for correct field mapping, account determination, and derivation rule execution following any DR event that involves a system upgrade or platform migration on the recovery path [9]. A mismatch between the replication component version and the target platform release can produce technically valid but financially incorrect journal entries—postings that pass automated document validation but carry incorrect account assignments, missing profit center derivations, or malformed cost object references that only surface during manual reconciliation or external audit review. The Application Interface Framework provides the integration monitoring and error management layer for financial message processing, and during and after a DR event, AIF queues must be explicitly cleared and reprocessed, with error queue states validated against the restored universal journal to confirm that every in-flight message at the time of the event has been either successfully reprocessed or explicitly dispositioned with documented rationale [10]. Value mapping consistency must be verified post-recovery to ensure that the mapping tables used during replay match those present at the time of the original postings, preventing account determination divergence between original and replayed journal entries that would produce an unreconciled discrepancy in the central ledger and trigger audit findings.

Cluster Component	Function	Failure Risk Mitigated	Key Configuration Requirement
STONITH Fencing	Isolates failed primary before promotion	Split-brain, data divergence	Mandatory for all Tier-0 clusters
Pacemaker Resource Agent	Automates failover sequencing	Manual error, RTO overrun	Configured for HANA and ASCS/ERS
Enqueue Replication Server	Mirrors the ASCS lock table continuously	Lock table loss on ASCS failure	Active standby with cluster promotion
Virtual/Overlay IP	Routes application traffic post-failover	DNS propagation delay	Pre-configured on all cluster nodes

Table 4: Cluster Governance, ASCS/ERS, and Split-Brain Prevention [9, 10]

6. AI-Driven Observability, Runbook Automation, and Continuous Validation

AI and machine learning transform disaster recovery from a reactive discipline—activated only after a failure has already occurred and impact is already accumulating—into a continuously verified operational posture where risks are detected before they escalate, and recovery readiness is proven through regular automated validation rather than assumed from static documentation. A consolidated telemetry pipeline ingests HANA System Replication latency and synchronization state metrics, replication engine queue depths and tracking identifier states, interface framework error rates, application log anomalies, and network throughput measurements—feeding ML models trained on historical operational baselines to detect the early warning indicators of ZDL risk before they produce observable service impact that triggers manual escalation [11].

The Active/Active read-enabled replication configuration is a specific area where continuous AI monitoring delivers significant operational value. When read workloads—analytics queries, reporting extracts, and batch processing jobs—are routed to the synchronous secondary, the secondary's resource utilization, read latency, and replication log application throughput must be continuously monitored to detect conditions where read load is beginning to affect log application speed [11]. This degradation pattern, if uncorrected, can cause the secondary to fall progressively behind the primary's commit rate and ultimately threaten the synchronous replication guarantee in a way that may not be immediately visible to human operators monitoring individual system dashboards in isolation. ML anomaly detection models identify these compound degradation patterns at a point where corrective action—such as read workload throttling, query prioritization adjustment, or secondary resource scaling—is still straightforward and can be executed without service disruption, well before the secondary's synchronization lag reaches a threshold that would trigger a cluster-enforced replication mode downgrade or a split-brain prevention fencing action.

Large language models augment the observability layer by synthesizing multi-source telemetry into concise and actionable incident narratives that reduce the cognitive load on operators responding to complex, multi-system anomalies in hybrid financial landscapes. An incident that simultaneously manifests as a replication queue depth increase, an interface framework error rate spike, and a network latency excursion—each visible in a different monitoring system—is

consolidated by LLM-assisted diagnosis into a unified root cause hypothesis and recommended remediation sequence, substantially reducing mean time to diagnosis and accelerating human decision-making under operational pressure [12]. Critically, LLM-assisted runbooks enforce policy guardrails that prevent automation from executing actions that would compromise the ZDL guarantee—automated failover is explicitly blocked when the secondary database is not confirmed to be in a synchronized and active replication state, preventing the speed-for-integrity trade-off that may be tempting under live incident pressure but carries regulatory consequences that far outlast the operational event itself. AWS documentation for SAP HANA on cloud infrastructure similarly emphasizes that combining system replication with structured automated monitoring, health validation, and documented recovery runbooks is the foundation of operationally proven recovery readiness—a principle that AI-driven DR architectures extend by making the validation continuous, automated, and evidence-producing rather than periodic and manually executed [12]. Post-drill and post-event, ML models perform universal journal reconciliation—comparing document counts, posting values, and tracking identifier queue states across the central journal before and after each recovery event—to produce structured, auditable evidence of recovery completeness that satisfies both internal governance requirements and external regulatory audit standards, transforming the DR program from a compliance document into a continuously exercised and verified operational guarantee.

DR Procedure Step	Purpose	Risk if Skipped	Validation Checkpoint
Safe Deactivation	Cleanly stops replication before takeover	Queue corruption, duplicate postings	MTID state and queue depth captured
Content Version Validation	Aligns field mappings with the target release	Silent incorrect account determination	Component version vs. target release check
Transfer Status Reset	Clears inconsistent replication baseline	Document range duplication or omission	Clean transfer status confirmed
AIF Queue Reprocessing	Resolves in-flight messages post-recovery	Incomplete journal, value mapping errors	Error queue cleared, reprocessing confirmed

Table 5: Replication Fabric Integrity and CFIN-Specific DR Procedures [11, 12]

Conclusion

A credible Zero-Data-Loss strategy for hybrid financial ERP landscapes depends on the precise integration of four technically and operationally interdependent capabilities. Synchronous full-sync HANA System Replication with multi-target topology enforces RPO = 0 at the financial core by ensuring every committed transaction is durably persisted on the secondary before application confirmation, while cluster-governed ASCS/ERS failover with mandatory STONITH fencing eliminates the split-brain risk that would otherwise render the replication guarantee meaningless under real failure conditions. Disciplined replication fabric procedures—encompassing safe deactivation before database takeover, message-tracking-state-aware deterministic replay, and Application Interface Framework reprocessing with value mapping validation—preserve centralized finance ledger integrity across every recovery event, ensuring that the universal journal remains a provably complete and sequentially consistent financial record regardless of the nature or scope of the disruption. AI-driven observability closes the operational loop on replication health, predicting ZDL risk conditions from compound telemetry signals before they materialize, failing over only via policy guardrails that block unsafe actions within automated runbooks, and producing structured universal journal reconciliation evidence that satisfies internal governance and external regulatory audit requirements in one motion. The cumulative effect is a disaster recovery posture that does not rely on periodic manual validation or optimistic assumptions about recovery behavior, but rather where every layer is continuously monitored, every recovery action is automation-governed, and every recovery event produces verifiable evidence of data completeness. The financial DR program thus becomes an always-on process with a continuously proven operational guarantee to enforce the non-negotiable integrity requirements of financial systems of record.

References

[1] Johan Steenstra, "What is SAP Central Finance?" KPMG. [Online]. Available: https://kpmg.com/ch/en/insights/technology/sap-s4hana-erp-potential/sap-central-finance.html

[2] SAP, "SAP HANA Administration Guide for SAP HANA Platform." [Online]. Available: https://help.sap.com/docs/SAP_HANA_PLATFORM/6b94445c94ae495c83a19646e7c3fd56/330e5550b09d4f0f8b6cceb14a64cd22.html

[3] SAP, "Replication Modes for SAP HANA System Replication." [Online]. Available: https://help.sap.com/docs/SAP_HANA_PLATFORM/6b94445c94ae495c83a19646e7c3fd56/c039a1a5b8824ecfa754b55e0caffc01.html

[4] SAP, "AIF-Error Handling." [Online]. Available: https://help.sap.com/docs/SUPPORT_CONTENT/abapconn/3354079588.html

[5] SAP, "AP HANA Administration Guide for SAP HANA Platform, Active/Active (Read Enabled)." [Online]. Available: https://help.sap.com/docs/SAP_HANA_PLATFORM/6b94445c94ae495c83a19646e7c3fd56/fe5fc53706a34048bf4a3a93a5d7c866.html

[6] SAP, "SAP HANA Multitarget System Replication." [Online]. Available: https://help.sap.com/docs/SAP_HANA_PLATFORM/4e9b18c116aa42fc84c7dbfd02111aba/ba457510958241889a459e606bbcf3d3.html

[7] SAP Community, "Implementation of SAP S/4HANA 2022 using SAP best Practice," 2025. [Online]. Available: https://community.sap.com/t5/technology-q-a/implementation-of-sap-s-4hana-2022-using-sap-best-practice/qaq-p/14199942

[8] Red Hat Customer Portal, "RHEL 7.6+ Guidelines for Configuring SAP S/4HANA ASCS/ERS with Standalone Enqueue Server 2 (ENSA2) in Pacemaker," 2025. [Online]. Available: https://access.redhat.com/articles/3974941

[9] SAP, "Upgrading Third-Party System Interfaces to Central Finance," 2025. [Online]. Available: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/26c2d5e366bc44c1a98f2a9212a0c49d/fa9b7f91203e43f890e89a18c230bdf8.html

[10] SAP, "Stop SLT Replication and Reset Transfer Status." [Online]. Available: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/26c2d5e366bc44c1a98f2a9212a0c49d/9bac171f0099450da779c9e332268c92.html

[11] Microsoft, "High availability for SAP HANA on Azure VMs on SUSE Linux Enterprise Server," 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/sap/workloads/sap-hana-high-availability?tabs=lb-portal%2Csaphanasr-angi

[12] Amazon Web Services, "SAP HANA Guides." [Online]. Available: https://docs.aws.amazon.com/sap/latest/sap-hana/welcome.html