© OCT 2019 | IRE Journals | Volume 3 Issue 4 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725090

A Theoretical Framework for Zero-Downtime Multi-Subnet Availability Group Architectures in Mission-Critical Clinical Review Estates

Evans Ighodalo[1], Ayinoluwa Kolawole[2], Emmanuel Chukwuma Okeke[3], Oluranti Ogundapo[4]
[1] *ModajPro IT Solutions, USA*
[2] *Personalised Haven, Nigeria*
[3] *University of Texas at Austin / IDEX Corporation, USA*
[4] *Huawei Technologies, Nigeria*

Abstract- ***Clinical review estates, the database platforms behind clinical trial adjudication, pharmacovigilance and diagnostic review, must remain available, accurate and secure at all times, yet there is little theory to guide how they should be built across multiple data centres. This paper develops a theoretical framework for zero-downtime multi-subnet SQL Server Availability Group architectures in such settings. Drawing on a synthesis of 120 international and regional sources published up to 2019, it defines zero-downtime from the point of view of the clinical user and models how design choices affect availability, data loss and security. The analysis shows that separating replicas into independent failure domains matters more than adding replicas, that the way client applications reconnect is often the main cause of extended outages, and that every replica must carry the same security controls to remain compliant with health data regulation. The framework offers architects, auditors and researchers a practical basis for designing and assessing resilient clinical data platforms.***

Index Terms- ***Always On Availability Groups, clinical data security, disaster recovery, high availability, multi-subnet failover, quorum, regulatory compliance, Windows Server Failover Clustering***

I. INTRODUCTION

Healthcare and life sciences organisations have spent the past decade consolidating fragmented clinical applications into shared database estates. A single estate may now host electronic data capture repositories for multinational trials, adverse event databases subject to expedited reporting deadlines, and review platforms in which clinicians adjudicate endpoints or validate coded diagnoses. These workloads share three properties that distinguish them from general enterprise systems. First, interruption carries patient safety and regulatory consequences: a missed fifteen-day expedited safety report or an unavailable unblinding service is not merely a business inconvenience. Second, the data are among the most sensitive categories recognised in law, attracting obligations under the United States Health Insurance Portability and Accountability Act (HIPAA), the Food and Drug Administration's electronic records rule (21 CFR Part 11) and, since May 2018, the European Union General Data Protection Regulation (GDPR). Third, reviewers are globally distributed and work across time zones, so there is no natural maintenance window. Computer downtime in hospitals has been shown to disrupt clinical processes and to create patient safety hazards (Kilbridge, 2003; Larsen et al., 2018; Wang et al., 2016), while breaches of protected health information continue to grow in number and scale (Liu et al., 2015). Studies of health system delivery note that clinical and scientific advances depend on dependable underlying infrastructure (Ilodigwe & Adesemoye, 2019a, 2019b), and research on diagnostic facilities shows that laboratory and review workflows are only as resilient as the platforms that carry them (Aminu-Ibrahim et al., 2018, 2019; Ogbete et al., 2018).

Microsoft SQL Server Always On Availability Groups (AGs) (Microsoft, 2019b), introduced in SQL Server 2012 and substantially extended through SQL Server 2017 and 2019, are widely used to meet these demands. When AG replicas are placed in different IP subnets, typically corresponding to separate data centres or availability zones, the architecture becomes a multi-subnet AG. Such deployments promise both local high availability and regional disaster recovery within a single coherent construct. However, the multi-subnet case introduces failure modes that do not arise on a flat network: listener name resolution across several IP addresses, dependence on DNS record time-to-live values, asymmetric network partitions, and quorum arithmetic that spans sites with unequal node counts.

Existing guidance is largely procedural, consisting of vendor documentation, configuration checklists and practitioner reports. Adjacent work has proposed conceptual frameworks for security audit and enterprise risk assessment (Dosunmu & Ogundele, 2019), risk-based cybersecurity assurance and data

availability (Fadayomi et al., 2019) and scalable, secure cloud architectures (Ahmed & Odejobi, 2018a), but these treat the availability and protection of clinical databases only in general terms. What is missing is a theoretical framework that explains why particular configurations succeed or fail, permits quantitative comparison of alternatives, and connects technical availability properties to the security and compliance properties demanded of clinical systems. This paper addresses that gap. Its contributions are as follows:

- A formal, testable definition of effective zero-downtime appropriate to transactional clinical workloads.
- A five-phase decomposition of failover latency and an associated downtime budget model.
- An availability model for multi-subnet AGs that explicitly accounts for correlated, common-cause failure.
- A set of propositions concerning quorum, listener behaviour and replication mode that define the safe operating envelope of the architecture.
- An integrated security model mapping technical controls in the AG stack to regulatory requirements for clinical data.

The remainder of the paper is organised as follows. The rest of Section 1 synthesises the relevant literature and identifies the research gap. Section 2 describes the research method. Section 3 presents and discusses the results, namely the conceptual model, reference architecture, integrated security model, operational procedures and analytical evaluation, together with their implications, limitations and directions for future research. Section 4 concludes.

A. Dependability and Consensus Foundations

The framework draws on the dependability taxonomy of Avižienis, Laprie, Randell and Landwehr (2004), which distinguishes availability (readiness for correct service), reliability (continuity of correct service), integrity (absence of improper alterations) and confidentiality. Clinical review estates require all four simultaneously, and the central argument of this paper is that they must be designed jointly rather than in sequence. Classical work on transaction processing (Gray and Reuter, 1993) and on concurrency control and recovery (Bernstein, Hadzilacos and Goodman, 1987) provides the vocabulary of durability, logging and redo used throughout. Reliability engineering for complex engineered systems offers complementary methods for failure analysis (Gideon et al., 2019), condition monitoring (Adaramola et al., 2018) and risk-based process design (Asibor & Isong, 2019), and studies of availability-critical industrial assets show how inspection, integrity management and condition-based maintenance sustain uptime (Obetta, 2018, 2019; Obetta & Ninduwezuor-Ehiobu, 2018, 2019; Okonkwo et al., 2018).

Distributed systems theory sets hard limits on what any replicated database can promise. The impossibility result of Fischer, Lynch and Paterson (1985) shows that deterministic consensus cannot be guaranteed in a fully asynchronous system with even one faulty process, while the CAP theorem (Brewer, 2000, 2012; Gilbert and Lynch, 2002) establishes that under network partition a system must trade consistency against availability. Practical systems circumvent these limits with timeouts, leases and majority quorums, as in Paxos (Lamport, 1998) and Raft (Ongaro and Ousterhout, 2014). This line of work builds on logical clocks (Lamport, 1978), Byzantine agreement (Lamport et al., 1982), consensus under partial synchrony (Dwork et al., 1988), unreliable failure detectors (Chandra & Toueg, 1996), the state machine approach to replication (Schneider, 1990) and the simplified presentation of Paxos (Lamport, 2001). Gray et al. (1996) showed that eager replication scales poorly while lazy replication risks inconsistency, a trade-off later systematised in surveys of database replication (Wiesmann et al., 2000) and in the consistency and latency analysis of Abadi (2012). Highly available stores such as Dynamo accepted eventual consistency to remain writable during partitions (DeCandia et al., 2007; Vogels, 2009), a choice that is unsuitable for clinical systems of record. Windows Server Failover Clustering (WSFC), upon which AGs depend, is an instance of this approach: it uses majority voting with optional witnesses to decide which partition may host the primary replica.

Empirical studies of production failures add a further dimension. Gray (1986) found that administration and software, rather than hardware, caused most outages in fault-tolerant systems, and later studies of cloud services confirm that outages are frequently triggered by configuration changes, upgrades and partial failures (Gunawi et al., 2016; Yuan et al., 2014). Network partitions are more common than often assumed (Bailis & Kingsbury, 2014; Gill et al., 2011) and frequently cause catastrophic failures in replicated systems (Alquraan et al., 2018). Gray failures, in which components are degraded rather than failed, evade conventional health checks (Huang et al., 2017), and large-scale measurements show that storage and node failures are correlated in time and location (Ford et al., 2010; Schroeder & Gibson, 2007).

B. Availability Groups and Multi-Subnet Operation

An AG is a set of user databases that fail over together. One primary replica accepts read-write workload; secondary replicas receive transaction log

records and may serve read-only queries. Each secondary operates in synchronous-commit or asynchronous-commit mode. In synchronous mode the primary waits for the secondary to harden log records before acknowledging a commit, which yields a recovery point objective (RPO) of zero for that secondary at the cost of added commit latency. SQL Server 2019 raised the maximum number of synchronous replicas to five and introduced secondary-to-primary connection redirection, while SQL Server 2017 added the REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT option that allows a primary to refuse commits unless a specified number of synchronous secondaries are healthy. Accelerated Database Recovery (ADR), introduced in SQL Server 2019 (Microsoft, 2019a), bounds the undo phase of recovery and so shortens role transitions for databases with long-running transactions. The design of ADR is described by Antonopoulos et al. (2019) and builds on the ARIES write-ahead logging recovery method (Mohan et al., 1992).

Clients reach the primary through an AG listener, a virtual network name with one IP address per subnet. On a multi-subnet listener only the address in the subnet of the current primary is online. Clients that specify MultiSubnetFailover=True in their connection string resolve all registered addresses and attempt connections in parallel, binding to whichever responds. Clients that do not, including many legacy clinical applications, depend on DNS returning the online address and on the record time-to-live (TTL) expiring after failover (Mockapetris, 1987). The cluster parameters RegisterAllProvidersIP and HostRecordTTL govern this behaviour (Microsoft, 2019c) and are therefore first-class design variables in the framework. Listener registration and client authentication also rest on directory services, whose secure operation and assessment have been examined by Ladapo et al. (2018, 2019). Work on virtual machine placement, cloud network resource allocation and cloud database optimisation (Ahmed & Odejobi, 2018b; Ahmed et al., 2019; Odejobi et al., 2019), and on performance under high concurrency in multi-tenant platforms (Odejobi & Ahmed, 2018), informs the sizing of replicas hosted on virtualised infrastructure.

C. Clinical Review Estates, Security and Regulation

For the purposes of this paper, a clinical review estate is a set of database-backed applications in which qualified personnel inspect, annotate, adjudicate or sign off clinical data, together with the infrastructure that supports them. Examples include endpoint adjudication committees in clinical trials, medical coding review, pharmacovigilance triage and radiology reporting worklists. The HIPAA Security Rule (45 CFR 164.308 and 164.312) requires contingency planning, access control, audit controls, integrity and transmission security for electronic protected health information (U.S. Department of Health and Human Services, 2013; National Institute of Standards and Technology, 2008). 21 CFR Part 11 requires validated systems, secure computer-generated audit trails and controls over electronic signatures (U.S. Food and Drug Administration, 1997, 2003). The GDPR (European Parliament and Council, 2016), in Article 32, requires the ability to ensure ongoing confidentiality, integrity, availability and resilience of processing systems, and to restore availability in a timely manner after an incident. Availability is thus not merely an engineering aspiration but a legal obligation, and the evidence that it has been achieved must itself be auditable. Comparative reviews of data protection regulation and secure cloud implementation across jurisdictions (Mbonu, Aliliele, et al., 2019), legal and ethical risk modelling for data protection governance (Mbonu et al., 2018), and governance frameworks for platforms operating in regulated healthcare environments (Badmus et al., 2019b) all indicate that compliance must be engineered into the platform rather than added later. International reviews of healthcare cybersecurity identify ransomware, legacy systems and weak access control as persistent threats (Coventry & Branley, 2018; Kruse et al., 2017; Martin et al., 2017), and a systematic review of electronic health record security names encryption, access control and audit as the core safeguards (Fernández-Alemán et al., 2013). Clinical trial data are further governed by Good Clinical Practice requirements for validated computerised systems (International Council for Harmonisation, 2016), and sociotechnical models of health information technology stress that technical availability cannot be separated from the clinical workflows that depend on it (Sittig & Singh, 2010). The same concern for integrity appears in regulatory-compliant laboratory design (Ogbete et al., 2019), diagnostic quality systems (Yusuf & Abdulrazaq, 2018), integrated laboratory information systems (Abdulrazaq et al., 2019), surveillance data quality (Ekemezie et al., 2018) and pharmaceutical chain-of-custody traceability (Aneke & Adesemoye, 2019).

D. Literature Synthesis and Research Gap

Table 1 summarises the four themes. Read together, the literature establishes the theoretical limits of replication, describes the mechanics of Availability Groups in detail, and sets out strong expectations for security, integrity and compliance in regulated settings. What it does not provide is an integrated account that links specific multi-subnet design choices

© OCT 2019 | IRE Journals | Volume 3 Issue 4 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725090

to client-observed availability, data loss, and auditable compliance in clinical estates. Dependability theory is largely platform-neutral; vendor documentation is procedural and rarely quantitative; security and governance studies seldom model failover; and reliability studies from industrial domains are rich in method but have not been transferred to replicated clinical databases. This gap motivates the conceptual model that follows.

Table 1. Synthesis of literature themes

Theme	Representative sources	Principal contribution	Gap for this study
Dependability and consensus	Avižienis et al. (2004); Fischer et al. (1985); Gilbert & Lynch (2002); Lamport (1998); Ongaro & Ousterhout (2014); Gray (1986); Gunawi et al. (2016)	Limits on consensus under partition; majority quorum; dependability attributes	Platform-neutral; no mapping to AG configuration variables
Availability group mechanics	Microsoft (2019a, 2019b, 2019c); Ahmed et al. (2019); Odejobi et al. (2019)	Replica roles, commit modes, listener and recovery behaviour; cloud resource sizing	Procedural; little quantitative or client-side analysis
Security and data governance	Mbonu et al. (2018); Dosunmu & Ogundele (2019); Fadayomi et al. (2019); Oshoba et al. (2019); Akeju et al. (2018)	Risk-based assurance, identity, audit and data protection governance	Seldom considers how failover alters the control surface
Regulated health data and reliability practice	Ogbete et al. (2019); Abdulrazaq et al. (2019); Larsen et al. (2018); Fernández-Alemán et al. (2013); Gideon et al. (2019); Obetta (2019)	Integrity, quality systems, reliability methods and compliance monitoring	Not yet transferred to replicated clinical databases

II. RESEARCH METHOD

A. Research Design

The study adopts a conceptual, theory-building research design. Rather than reporting a single empirical deployment, it seeks to explain why multi-subnet Availability Group configurations succeed or fail, and to express that explanation as definitions, propositions and a conceptual model that can later be tested empirically. This approach is appropriate where practice has run ahead of theory, as is the case for high-availability clinical database platforms, and where the phenomena of interest (rare failures, regulatory consequences, correlated outages) are costly or unsafe to reproduce in production clinical systems. The design proceeds in three stages: an integrative literature synthesis, theory construction, and analytical evaluation.

B. Literature Identification and Synthesis

The literature base was assembled from four bodies of work: foundational dependability, transaction processing and distributed systems research; vendor technical documentation for SQL Server and Windows Server Failover Clustering; regulatory and standards texts governing health data in the United States and the European Union; and applied studies on cybersecurity, cloud and enterprise IT, clinical laboratory and health information systems, reliability engineering, and risk governance. Sources were included if they (a) were published in or before 2019, consistent with the time frame of the study, (b) addressed availability, integrity, confidentiality, reliability or compliance of information or critical infrastructure systems, and (c) offered concepts, models or evidence transferable to replicated clinical databases. A total of 120 sources met these criteria, drawn from both international and regional literature. Each was coded against four themes, namely dependability and consensus theory, availability group mechanics, security and data governance, and regulated health data and reliability practice, and the themes were then compared to identify convergent findings and unresolved gaps (Section 1.4).

C. Theory Construction and Analytical Evaluation

Constructs and propositions were derived deductively from the synthesised literature and from the documented behaviour of the AG and WSFC stack. Each proposition was stated so that it identifies a design variable, the mechanism through which that variable acts, and the outcome it affects, which together form the conceptual model in Section 3.1.

© OCT 2019 | IRE Journals | Volume 3 Issue 4 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725090

The model was then evaluated analytically, by applying parameterised availability and latency calculations to representative configurations (Section 3.5). Parameter values were chosen to be typical of well-run enterprise estates and are reported transparently so that others may substitute their own.

D. Problem Statement and Assumptions

The research problem is to construct a framework that allows an architect to (a) state precisely what zero-downtime means for a clinical review estate, (b) predict whether a given multi-subnet AG design meets that definition, (c) identify the configuration variables that dominate the outcome, and (d) demonstrate that availability mechanisms do not weaken security or compliance controls. The framework rests on the following assumptions, which bound the scope of the analysis.

- A1. Replicas run SQL Server 2017 or later on Windows Server 2016 or later, in a single WSFC that spans at least two subnets.
- A2. Network round-trip time between sites hosting synchronous replicas is low enough (typically under 5 ms) that synchronous commit is tolerable for the workload.
- A3. Client applications implement, or can be wrapped with, idempotent retry logic for transient connection failures.
- A4. Failures of individual components are independent except where explicitly modelled as common-cause.
- A5. The estate operates under a documented validation regime, so configuration is controlled and changes are traceable, consistent with the use of standard operating procedures as change infrastructure (Agboada et al., 2019).

III. RESULTS AND DISCUSSION

This section reports the results of the theory construction and analytical evaluation and discusses them. It begins with the conceptual model (Section 3.1), then derives a reference architecture (Section 3.2), an integrated security model (Section 3.3) and operational procedures (Section 3.4), evaluates the model analytically (Section 3.5), and closes with a discussion of findings, implications, limitations and future research (Sections 3.6 to 3.9).

A. Conceptual Model

The conceptual model first defines the central outcome, effective zero-downtime, then specifies the mechanisms through which design choices affect that outcome, and states five propositions. Figure 1 depicts the complete model.

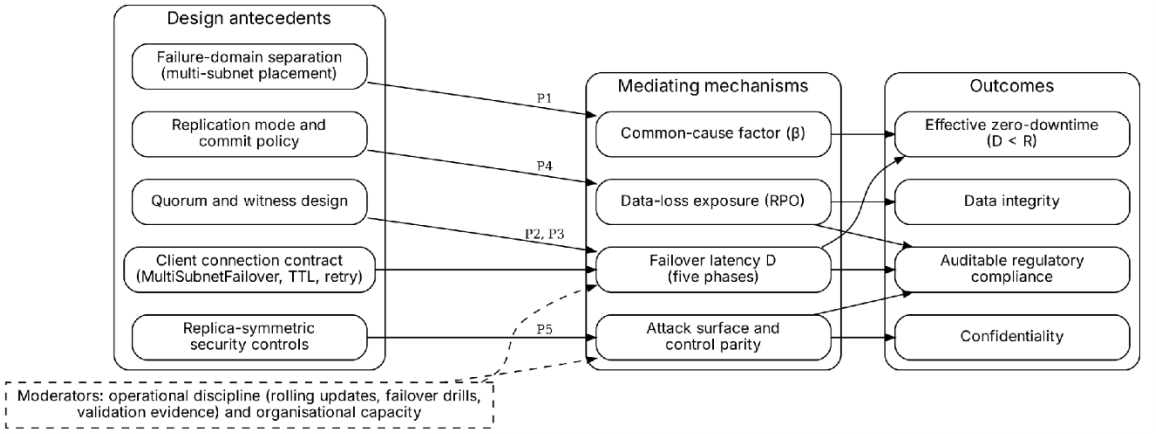


Figure 1. Conceptual model of zero-downtime multi-subnet AG architecture in clinical review estates

Effective Zero-Downtime

Literal zero downtime, meaning that no request is ever delayed by a failure, is unattainable in any system bound by the impossibility results cited above. A useful definition must instead refer to what the user of the service observes. Let R denote the retry budget of a client, defined as the maximum cumulative time the client will transparently retry a failed operation before surfacing an error to the human reviewer. Let D denote the duration of a service interruption, measured from the first rejected request to the first accepted request on the new primary.

Definition 1 (Effective zero-downtime). A service exhibits effective zero-downtime with respect to an event if D < R and no committed transaction is lost (RPO = 0). An architecture is effectively zero-downtime for a class of events if every event in the class satisfies this condition with probability at least p over the operating period.

This definition has three advantages. It is measurable, because both D and R can be

instrumented. It ties the infrastructure requirement to application behaviour, which exposes the common failure in which a well-engineered cluster is undermined by a client that does not retry. And it separates planned events, such as patching, where p can approach one, from unplanned events, where p is a design target.

Failover Latency Decomposition

The interruption D for an automatic failover is decomposed into five sequential phases:

$$D = T_{\text{detect}} + T_{\text{arbitrate}} + T_{\text{role}} + T_{\text{recover}} + T_{\text{client}} \quad (1)$$

T_detect is the time for the cluster or the SQL Server health detection mechanism to recognise failure. It is governed by the cluster heartbeat parameters (SameSubnetThreshold, CrossSubnetThreshold and their associated delays), the AG HEALTH_CHECK_TIMEOUT and FAILURE_CONDITION_LEVEL, and the lease timeout. T_arbitrate is the time for WSFC to confirm quorum and select a new owner. T_role covers the transition of the chosen secondary to the primary role, including bringing the listener IP resource online in its subnet. T_recover is database recovery on the new primary, consisting chiefly of redo of hardened but unapplied log and, without ADR, undo of uncommitted transactions. T_client is the interval between the listener coming online and the client re-establishing a session.

The decomposition yields a first insight: in multi-subnet deployments T_client frequently dominates. A client without MultiSubnetFailover support, resolving a listener whose DNS record carries the default HostRecordTTL of 1200 seconds, may experience an interruption of up to twenty minutes even when the server-side phases complete in under thirty seconds. The framework therefore treats the client connection contract as part of the architecture, not as an application concern.

Availability Under Correlated Failure

Let each replica have steady-state availability A = MTBF / (MTBF + MTTR) (Trivedi, 2002), and unavailability U = 1 - A. Under strict independence, a service that remains available while any one of n replicas is available has unavailability U^n. This idealisation drastically overstates real-world availability because replicas frequently share failure causes: a common storage array, power feed, hypervisor host, network core, patch, or operator (Gideon et al., 2019). Following the beta-factor method used in reliability engineering (Mosleh et al., 1998; Rausand & Høyland, 2004), a fraction beta of each replica's unavailability is attributed to common causes that disable all co-located replicas at once. System unavailability is then approximated by:

$$U_{\text{sys}} \approx (1-\beta)\,U^n + \beta\,U \quad (2)$$

When beta is non-trivial, the second term dominates and adding replicas yields sharply diminishing returns. The principal architectural lever is therefore not replica count but reduction of beta, which is exactly what placing replicas in separate subnets, sites and failure domains achieves, consistent with the finding that correlated failures dominate unavailability in globally distributed storage (Ford et al., 2010). This gives the framework its central proposition.

Proposition 1 (Failure-domain primacy). For any beta greater than U^(n-1), reducing beta improves system availability more than adding a further co-located replica. Multi-subnet placement is justified primarily as a beta-reduction mechanism, and only secondarily as a disaster recovery mechanism.

Quorum Across Subnets

WSFC grants the right to host the primary to the partition holding a strict majority of votes. Let V be the total number of votes, including any witness. A partition with v votes may run the AG only if v > V/2. Two properties follow.

Proposition 2 (Witness placement). In a two-site topology with equal node counts per site, the witness must reside in a third failure domain; otherwise the loss of the site hosting the witness also removes quorum from the surviving site, converting a single-site failure into a total outage.

Proposition 3 (Asymmetric vote weighting). Where one site is designated for disaster recovery only, its nodes should have their votes removed (NodeWeight = 0) so that transient inter-site link loss cannot cause the primary site to lose quorum. Dynamic quorum and dynamic witness, available since Windows Server 2012 R2, reduce but do not eliminate the need for this explicit design decision.

Quorum is also a safety mechanism. The lease between the SQL Server instance and the cluster ensures that a primary isolated from quorum stops accepting writes before another replica is promoted, preventing split-brain, an application of the lease mechanism introduced by Gray and Cheriton (1989) and used in production coordination services (Burrows, 2006). The lease timeout is therefore a lower bound on T_detect that cannot be tuned away without weakening this guarantee, illustrating the inherent tension between availability and consistency described by the CAP theorem.

Replication Mode and Data Loss

Proposition 4 (Zero data loss envelope). Automatic failover with RPO = 0 is possible only to a

synchronous-commit secondary that is in the SYNCHRONIZED state at the moment of failure. To guarantee that such a secondary exists, the primary must be configured with REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT of at least one, accepting that the primary will block commits if no synchronous secondary is healthy.

This proposition formalises a choice that clinical estates must make consciously. Blocking commits preserves integrity at the expense of availability; allowing commits to proceed preserves availability but permits a subsequent failover to lose acknowledged transactions. For audit trail tables governed by 21 CFR Part 11, the framework recommends the integrity-preserving setting, because a lost audit record is a compliance failure whereas a brief write pause is recoverable within the retry budget. This preference mirrors risk-based internal control models in which the integrity of the record takes precedence over throughput (Akomolafe & Agu, 2019a), and reconciliation controls that refuse to post unmatched entries (Dogbatsey et al., 2019).

Model Summary

Figure 1 organises the framework into three layers. Design antecedents are the variables an architect controls: failure-domain separation, quorum and witness design, replication mode and commit policy, the client connection contract, and replica-symmetric security controls. Mediating mechanisms are the quantities through which those choices act: the common-cause factor, failover latency, data-loss exposure, and the attack surface and parity of controls. Outcomes are the properties the clinical estate must demonstrate: effective zero-downtime, integrity, confidentiality and auditable compliance. Operational discipline and organisational capacity moderate the mechanisms, because even a sound design degrades if updates, drills and evidence collection are performed inconsistently. Table 2 lists the propositions and their place in the model; Proposition 5 is developed in Section 3.3.

Table 2. Propositions of the conceptual model

Proposition	Antecedent	Mechanism	Outcome affected
P1 Failure-domain primacy	Multi-subnet placement	Reduces common-cause factor β	Availability; effective zero-downtime
P2 Witness placement	Witness in third domain	Preserves quorum after site loss	Availability
P3 Asymmetric vote weighting	Vote removal at DR site	Prevents quorum loss on link failure	Availability
P4 Zero data loss envelope	Synchronous commit with required secondaries	Bounds RPO at zero	Integrity; compliance
P5 Replica symmetry	Identical controls on all replicas	Maintains control parity after failover	Confidentiality; compliance

B. Reference Architecture

Applying the framework, together with established high-availability design practice (Marcus & Stern, 2003), yields a reference architecture for a two-region clinical review estate, summarised in Table 3. The primary region contains two subnets in physically separate data halls or availability zones, each hosting one synchronous replica with automatic failover. A secondary region hosts one or two asynchronous replicas for disaster recovery and read-scale reporting. A cloud witness or file share witness in a third failure domain completes the quorum.

Table 3. Reference topology derived from the framework

Component	Placement	Mode	Rationale
Replica 1 (primary)	Region A, Subnet 1	Synchronous, automatic	Serves read-write clinical review workload
Replica 2	Region A, Subnet 2	Synchronous, automatic	Zero RPO target in an independent failure domain (Proposition 1)
Replica 3	Region B, Subnet 3	Asynchronous, manual	Regional disaster recovery; vote removed (Proposition 3)
Replica 4 (optional)	Region B, Subnet 4	Asynchronous, manual	Read-only routing for reporting and analytics
Witness	Third domain (cloud or site C)	Vote	Breaks ties between Subnets 1 and 2 (Proposition 2)

Component	Placement	Mode	Rationale
Listener	One IP per subnet	RegisterAllProvidersIP = 1	Parallel connection with MultiSubnetFailover

Distributed Availability Groups, available since SQL Server 2016, may be used where the secondary region must be operated by an independent cluster, for example to decouple cluster upgrades or to satisfy data residency boundaries. In that variant the inter-region link becomes an AG-to-AG relationship and the quorum domains of the two regions are fully independent, which further reduces correlated failure at the cost of manual cross-region failover.

Read-only routing lists direct reporting and analytics connections that specify ApplicationIntent=ReadOnly to secondaries, protecting the primary from heavy analytical queries that are common in pharmacovigilance signal detection. Because asynchronous secondaries may lag, the architecture requires that any read used to support a clinical decision be served from the primary or from a synchronous secondary, and that reporting interfaces display data currency to reviewers.

C. Integrated Security Model

High availability multiplies copies of sensitive data and opens new network paths between sites, so it expands the attack surface. The framework treats every availability mechanism as also a security object and requires that controls be equivalent on every replica, a property termed replica-symmetric security. The principle extends the security audit and enterprise risk assessment logic of Dosunmu and Ogundele (2019) to replicated estates, and responds to insider threat models in which privileged administrators are themselves a source of risk (Akeju et al., 2018).

Proposition 5 (Replica symmetry). The security posture of an AG is the minimum of the postures of its replicas and the channels between them. Any control present on the primary but absent on a secondary is, in effect, absent from the system, because failover can promote that secondary without human review.

The controls follow the principles of least privilege and complete mediation (Saltzer & Schroeder, 1975), role-based access control (Sandhu et al., 1996) and database-specific security models (Bertino & Sandhu, 2005), and can be governed within an information security management system (International Organization for Standardization, 2013). The following control families implement this proposition:

- Encryption at rest. Transparent Data Encryption (TDE) protects data and log files and backups. The database encryption key is protected by a certificate that must be present on every replica before the database is joined; certificates and their private keys are backed up to a protected location and their lifecycle is managed under change control.
- Column-level protection. Always Encrypted, and from SQL Server 2019 Always Encrypted with secure enclaves, keeps direct patient identifiers encrypted even from database administrators, supporting the separation of duties expected for unblinded trial data, in line with research on encrypted query processing (Popa et al., 2011).
- Encryption in transit. Database mirroring endpoints use certificate-based authentication with AES encryption, and client connections enforce TLS 1.2 (Dierks & Rescorla, 2008). Inter-site replication traffic additionally traverses encrypted private links or IPsec tunnels.
- Identity. Service accounts are Group Managed Service Accounts, eliminating static passwords shared across replicas. Logins, SQL Agent jobs, linked servers and credentials, which are instance-level objects not replicated by the AG, are synchronised by automated, audited tooling. Federated identity and access management models (Oshoba et al., 2019) and disciplined Active Directory implementation (Ladapo et al., 2019) support this control.
- Audit. SQL Server Audit specifications are deployed identically on every replica and write to a central, append-only store outside the cluster, so that the audit trail survives the loss of any replica and cannot be altered by the database administrators being audited. Centralised log analytics and automated incident response, as described by Mbonu, Iwuanyanwu, et al. (2019), allow anomalies across replicas to be correlated in one place.
- Network segmentation. Replication, cluster heartbeat, client and management traffic are separated by VLAN or subnet and firewall policy, limiting lateral movement and reducing the chance that a saturated client network delays heartbeats and triggers a spurious failover.

Table 4 maps these controls to the regulatory requirements discussed in Section 1.3.

Table 4. Mapping of AG controls to clinical regulatory requirements

Control	HIPAA Security Rule	21 CFR Part 11	GDPR
Multi-subnet AG with tested failover	164.308(a)(7) contingency plan	11.10(a) validation; 11.10(c) record protection	Art. 32(1)(b), (c) resilience and restoration
TDE and encrypted backups	164.312(a)(2)(iv) encryption	11.10(c) record protection	Art. 32(1)(a) encryption
Always Encrypted for identifiers	164.312(a)(1) access control	11.10(d) limited access	Art. 25 data protection by design
Certificate-secured endpoints, TLS	164.312(e)(1) transmission security	11.30 open system controls	Art. 32(1)(a)
Central immutable audit	164.312(b) audit controls	11.10(e) audit trails	Art. 30 records of processing
Synchronous commit with required secondaries	164.312(c)(1) integrity	11.10(a) accuracy	Art. 5(1)(f) integrity

D. Zero-Downtime Operations

Unplanned failures are rare; planned maintenance is frequent. In practice, most downtime in database estates is caused by patching, upgrades and schema changes. The framework therefore treats operational procedure as part of the architecture.

Rolling Patching and Upgrade

A rolling update proceeds by patching secondaries first, then performing a planned manual failover to a patched synchronous secondary, and finally patching the former primary. Because a planned failover to a synchronised secondary skips T_detect and T_arbitrate and requires no undo when ADR is enabled, D is typically a few seconds, well within a retry budget of thirty to sixty seconds. Planned events can therefore satisfy Definition 1 with probability approaching one, provided that the client contract is honoured. Cluster operating system upgrades follow the same pattern using Cluster OS Rolling Upgrade, available since Windows Server 2016. Standardised, maturity-assessed support operations across distributed sites (Anyanwu, 2019) and automated deployment pipelines (Badmus et al., 2018) reduce the risk that a rolling update is executed inconsistently between subnets.

Schema Change

Schema changes on the primary are replicated as log records and cannot be staged on secondaries. Zero-downtime schema evolution therefore relies on application-level techniques: expand-and-contract migrations in which new columns or tables are added in a backward-compatible manner, applications are deployed to use them, and obsolete structures are removed later. Online index operations and resumable index rebuilds, extended in SQL Server 2019, limit blocking during physical changes. Large operations must also be sized against the log send and redo capacity of remote secondaries, since a burst of log generation can push synchronous secondaries out of the SYNCHRONIZED state and so violate Proposition 4. Data integration and extract, transform and load processes that feed downstream review applications must be designed for the same backward compatibility (Badmus et al., 2019a).

Validation and Evidence

Regulated estates must demonstrate that availability controls work. The framework recommends periodic, scheduled failover exercises across every subnet, with automatic capture of the five latency phases, the client-observed interruption and the synchronisation state before and after. These records serve simultaneously as operational telemetry, as validation evidence under Part 11, and as evidence of tested restoration capability under GDPR Article 32(1)(d). The practice is closely related to chaos engineering (Basiri et al., 2016) and to the site reliability engineering discipline of deliberately testing recovery (Beyer et al., 2016), and it supports the contingency planning and business continuity requirements of NIST SP 800-34 (Swanson et al., 2010) and ISO 22301 (International Organization for Standardization, 2012). This approach parallels predictive compliance monitoring and adaptive safety management in other regulated sectors (Adeyelu, 2018, 2019), data-driven risk control in large infrastructure programmes (Obriki & Arumosoye, 2018), technology-enabled internal audit (Akomolafe & Agu, 2018) and operational integrity frameworks that balance productivity with compliance (Nwangwu et al., 2018).

E. Analytical Evaluation

Availability Scenarios

Consider replicas each with availability 0.999 (U = 0.001), a figure representative of a well-run single server including planned and unplanned outages.

© OCT 2019 | IRE Journals | Volume 3 Issue 4 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725090

Table 5 applies the model of Section 3.1.3 to three configurations with two automatic-failover replicas.

Table 5. Modelled availability of two-replica configurations (U = 0.001)

Configuration	β	U_sys	A_sys
Ideal independence (theoretical)	0	1.0×10^{-6}	0.999999
Same subnet, shared rack and storage	0.05	5.1×10^{-5}	0.999949
Separate subnets and failure domains	0.005	6.0×10^{-6}	0.999994

The co-located configuration delivers roughly 99.995 percent availability, about 27 minutes of downtime per year, almost entirely attributable to common causes. Moving the second replica to a separate subnet and failure domain reduces modelled unavailability by a factor of about 8.5 and brings annual downtime to roughly 3 minutes, inside a five-nines budget. Adding a third co-located replica to the first configuration would barely change the result, because the beta term would still dominate, confirming Proposition 1.

Failover Budget

A 99.999 percent annual availability target permits approximately 315 seconds of downtime per year. Table 6 shows an illustrative failover latency profile for a tuned multi-subnet AG with ADR enabled and clients using MultiSubnetFailover, alongside a legacy client relying on DNS.

Table 6. Illustrative failover latency profile (seconds)

Phase	Tuned, modern client	Tuned, legacy client (TTL 300 s)	Default, legacy client (TTL 1200 s)
T_detect	10	10	30
T_arbitrate	2	2	5
T_role	5	5	10
T_recover	3	3	20
T_client	1	up to 300	up to 1200
Total D (worst case)	21	320	1265

With a modern client, a single unplanned failover consumes about 21 seconds, so the annual budget accommodates roughly fifteen such events, and each falls within a typical thirty-second retry budget, satisfying Definition 1. With a legacy client, a single failover can exceed the entire annual five-nines budget. The analysis demonstrates quantitatively that, in multi-subnet clinical estates, the connection behaviour of the oldest application is often the binding constraint on availability, and that reducing HostRecordTTL or fronting legacy clients with a connection broker is a higher-value intervention than further server-side tuning.

F. Discussion of Findings

The analysis supports three principal findings. First, availability in multi-subnet clinical estates is governed more by correlated failure than by replica count. The beta-factor model shows that a second replica placed in a separate failure domain improves availability far more than additional co-located replicas, which reframes multi-subnet placement as a primary high-availability measure rather than only a disaster recovery one. This is consistent with reliability engineering practice in other critical infrastructures, where separation of failure modes and condition-based maintenance are the main levers on uptime (Gideon et al., 2019; Obetta & Ninduwezuor-Ehiobu, 2018; Okonkwo et al., 2018).

Second, client connectivity is frequently the binding constraint. The latency decomposition shows that server-side failover can complete in tens of seconds while a legacy client relying on DNS remains disconnected for minutes. Most existing guidance treats the client as outside the database architecture; the model indicates that the connection contract belongs within it, alongside quorum and replication decisions.

Third, availability and security cannot be designed in sequence. Every replica is a copy of protected health information, and every failover can promote a server without human review, so the security posture of the estate equals that of its weakest replica. This extends risk-based assurance and security audit frameworks (Dosunmu & Ogundele, 2019; Fadayomi et al., 2019) to replicated systems, and aligns with governance studies that call for compliance to be engineered into platforms in regulated healthcare environments (Badmus et al., 2019b; Mbonu, Aliliele, et al., 2019).

These findings sit within a broader organisational context. Digital transformation of operations (Agboada et al., 2018), data-driven executive decision systems (Lawal & Oduleye, 2019), data-driven risk evaluation (Akomolafe & Agu, 2019b), financial resilience through integrated governance (Akomolafe & Agu, 2019c), transparent accounting and compliance systems (Bola-Sadipe et al., 2019), multinational compliance strategy (Anichukwueze et al., 2019) and supply chain risk management for critical projects (Agbabiaka et al., 2019) all shape whether an organisation can fund and sustain the architecture described here. Clinical services in low-connectivity settings may additionally need offline-first designs that tolerate prolonged loss of the primary site (Unamma et al., 2018).

G. Implications

Implications for Practice

- Architects should prioritise placing synchronous replicas in independent failure domains before adding further replicas within one site.
- The client connection contract (MultiSubnetFailover support, HostRecordTTL and retry budget) should be specified, tested and change-controlled as part of the database design.
- Quorum configuration, including witness location and vote weighting, should be documented as an explicit design decision and revisited whenever the topology changes.
- Security controls, certificates, logins and audit specifications should be deployed and verified identically on every replica, with automated drift detection.
- Scheduled failover drills should capture the five latency phases so that operational telemetry also serves as validation evidence.

Implications for Theory

The study contributes a client-referenced definition of zero-downtime that makes the concept measurable, a phase decomposition that connects distributed systems theory to observable failover behaviour, and a set of propositions that link design antecedents to outcomes through identifiable mechanisms. It also demonstrates that reliability concepts developed for industrial assets, such as common-cause analysis and risk-based inspection (Obetta, 2018; Asibor & Isong, 2019), transfer productively to replicated information systems.

Implications for Regulators and Auditors

For auditors and quality assurance functions, the framework provides concrete evidence points for availability claims: recorded failover phase durations, synchronisation state at the time of failure, quorum configuration and replica control parity. These evidence points map directly to HIPAA contingency planning, 21 CFR Part 11 audit trail and validation requirements, and GDPR Article 32 obligations, supporting a shift from document-based to evidence-based assessment of resilience, in line with technology-enabled audit and predictive compliance monitoring approaches (Akomolafe & Agu, 2018; Adeyelu, 2018).

H. Limitations

The framework is analytical and its numerical outputs depend on parameter estimates. Beta factors in particular are difficult to measure and vary between organisations; the values used here are illustrative, and practitioners should calibrate them from incident histories. The model also simplifies several phenomena. It treats phases of failover as sequential and deterministic, whereas real failovers exhibit variance and occasional retries of the arbitration step. It does not model gray failures, in which a replica is degraded but not failed, and which may evade health detection while harming performance. Nor does it address human error during recovery, which empirical studies of operations consistently identify as a leading cause of extended outages (Gray, 1986; Reason, 1990).

The scope is also limited to the database tier. Clinical review services depend on application servers, identity providers, document and imaging storage, and network services, each with its own availability and security profile. Finally, the literature synthesis is restricted to sources published in or before 2019 and to a single database platform, so the conclusions may not transfer directly to other database engines or to later platform versions.

I. Future Research

- Empirical validation of the propositions through controlled failure injection in representative, non-production clinical environments.
- Measurement of common-cause factors from incident records across multiple healthcare and life sciences organisations.
- Extension of the latency decomposition and replica symmetry principle to application, identity and storage tiers.
- Stochastic rather than deterministic models of failover phases, including the effect of gray failures.
- Adaptation of the framework to container-orchestrated deployments such as SQL Server 2019 Big Data Clusters on Kubernetes, where detection and arbitration mechanisms differ considerably.
- Study of human and organisational factors in failover execution, building on work on SOP-driven change and IT support maturity (Agboada et al., 2019; Anyanwu, 2019).

IV. CONCLUSION

Mission-critical clinical review estates require continuous availability, uncompromised integrity and rigorous confidentiality, all under regulatory scrutiny. Multi-subnet Always On Availability Groups can meet these requirements, but only when designed with an explicit understanding of how failure detection, quorum, replication mode, client connectivity and security controls interact. This paper has offered a theoretical framework for that understanding. It defines effective zero-downtime in terms of client retry budgets and zero data loss, decomposes failover

into measurable phases, models correlated failure, and states five propositions that bound the safe operating envelope: failure-domain primacy, witness placement, asymmetric vote weighting, the zero data loss envelope and replica symmetry. The analytical evaluation shows that separating failure domains, rather than multiplying replicas, is the dominant lever on availability, and that client connection behaviour is frequently the binding constraint in multi-subnet deployments. By uniting availability engineering with security and compliance reasoning in a single model, the framework aims to help architects build, auditors assess, and researchers compare the database infrastructure on which modern clinical review depends.

REFERENCES

[1] Abadi, D. (2012). Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. Computer, 45(2), 37–42.

[2] Abdulrazaq, K., Yusuf, H. D., Osinem, H. E., & Eribenne, F. (2019). A model for improving infection detection accuracy through integrated laboratory information systems. Iconic Research and Engineering Journals, 3(6), 650-675. https://doi.org/10.64388/IREV3I6-1723089

[3] Adaramola, T. S., Fadero, S., & Gideon, E. N. (2018). Predictive maintenance and condition monitoring in critical power and energy infrastructure. Iconic Research and Engineering Journals, 2(5), 454-477. https://doi.org/10.64388/IREV2I5-1722489

[4] Adeyelu, O. O. (2018). A predictive compliance monitoring framework for detecting systemic safety risks through aviation consumer complaint data. Iconic Research and Engineering Journals, 1(8), 250-276. https://doi.org/10.64388/IREV1I8-1718478

[5] Adeyelu, O. O. (2019). An adaptive safety management system implementation model for aerodromes in developing economy contexts. Iconic Research and Engineering Journals, 3(4), 628-654. https://doi.org/10.64388/IREV3I4-1718479

[6] Agbabiaka, J., Okonkwo, C. S., Ogunwole, O., Mayo, W., & Okeke, O. T. (2019). Supply chain risk management model for EPC and gas processing projects. Iconic Research and Engineering Journals, 3(2), 968-980. https://doi.org/10.64388/IREV3I2-1713124

[7] Agboada, P., Ezeako, R., Okoli, I. E., & Uzondu, C. S. (2018). Digital transformation in operations management: A review of advances in automation, analytics, and process redesign. International Journal of Engineering Technology Research & Management, 2(11), 109-134.

[8] Agboada, P., Ezeako, R., Okoli, I. E., & Uzondu, C. S. (2019). Standard operating procedures as change infrastructure: A conceptual framework for sustaining operational improvement. Iconic Research and Engineering Journals, 3(6), 706-730. https://doi.org/10.64388/IREV3I6-1723313

[9] Ahmed, K. S., & Odejobi, O. D. (2018a). Conceptual framework for scalable and secure cloud architectures for enterprise messaging. IRE Journals, 2(1), 1-15.

[10] Ahmed, K. S., & Odejobi, O. D. (2018b). Resource allocation model for energy-efficient virtual machine placement in data centers. IRE Journals, 2(3), 1-10.

[11] Ahmed, K. S., Odejobi, O. D., & Oshoba, T. O. (2019). Algorithmic model for constraint satisfaction in cloud network resource allocation. IRE Journals, 2(12), 516-532.

[12] Akeju, B., Edivri, J., Ogbole, J. I., Okoruwa, P. O., Fadayomi, O., & Abolaji, T. O. (2018). Conceptual model for insider threat classification and risk modeling in complex digital systems. Iconic Research and Engineering Journals, 1(9), 476-492. https://doi.org/10.64388/IREV1I9-1713778

[13] Akomolafe, O., & Agu, M. U. (2018). A conceptual model for enhancing internal audit quality through technology-enabled risk assessment frameworks. Iconic Research and Engineering Journals, 1(9), 458-475.

[14] Akomolafe, O., & Agu, M. U. (2019a). A conceptual framework for developing risk-based internal control models in the insurance and banking sectors. Iconic Research and Engineering Journals, 2(8), 335-354.

[15] Akomolafe, O., & Agu, M. U. (2019b). A review of data-driven risk evaluation models for emerging market financial institutions. Iconic Research and Engineering Journals, 3(6), 433-448.

[16] Akomolafe, O., & Agu, M. U. (2019c). Advances in financial resilience through integrated governance and compliance strategies. Iconic Research and Engineering Journals, 2(10), 607-620.

[17] Alquraan, A., Takruri, H., Alfatafta, M., & Al-Kiswany, S. (2018). An analysis of network-partitioning failures in cloud systems. In

Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (pp. 51–68). USENIX Association.

[18] Aminu-Ibrahim, A. Y., Ogbete, J. C., & Ambali, K. B. (2018). Developing sustainable diagnostic laboratory infrastructure models for emerging and resource constrained health systems. Iconic Research and Engineering Journals, 1(8), 118-132. https://doi.org/10.64388/IREV1I8-1713586

[19] Aminu-Ibrahim, A. Y., Ogbete, J. C., & Ambali, K. B. (2019). Capital project delivery models for high risk healthcare infrastructure in developing national health systems. Iconic Research and Engineering Journals, 2(10), 626-649. https://doi.org/10.64388/IREV2I10-1713588

[20] Aneke, O. B., & Adesemoye, A. C. (2019). Toward an integrated conceptual framework for traceability and chain-of-custody integrity in pharmaceutical distribution. Iconic Research and Engineering Journals, 3(4), 655-672.

[21] Anichukwueze, C. C., Osuji, V. C., & Oguntegbe, E. E. (2019). Global marketing law and consumer protection challenges: A strategic framework for multinational compliance. IRE Journals, 3(6), 325-333.

[22] Antonopoulos, P., Byrne, P., Chen, W., Diaconu, C., Kodandaramaih, R. T., Kodavalla, H., Purnananda, P., Radu, A.-L., Ravella, C. S., & Venkataramanappa, G. M. (2019). Constant time recovery in Azure SQL Database. Proceedings of the VLDB Endowment, 12(12), 2143–2154. https://doi.org/10.14778/3352063.3352131

[23] Anyanwu, K. C. E. (2019). Developing a maturity model for standardizing IT support operations across geographically distributed multi-office organizations. Iconic Research and Engineering Journals, 3(5), 580-598.

[24] Asibor, E., & Isong, D. E. (2019). Risk-based process engineering framework for strengthening reliability across complex natural gas infrastructure systems. Iconic Research and Engineering Journals, 3(2), 1020-1060. https://doi.org/10.64388/IREV3I2-1723153

[25] Avižienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1(1), 11–33.

[26] Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2018). A systematic review of CI/CD pipeline strategies in Salesforce DevOps: Tools, practices, and deployment outcomes. Iconic Research and Engineering Journals, 2(6).

[27] Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2019a). A conceptual model for ETL design and data integration in Salesforce-centric enterprise architectures. Iconic Research and Engineering Journals, 3(5).

[28] Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2019b). A governance framework for Salesforce platform management in regulated healthcare environments. Iconic Research and Engineering Journals, 3(6).

[29] Bailis, P., & Kingsbury, K. (2014). The network is reliable. Communications of the ACM, 57(9), 48–55.

[30] Basiri, A., Behnam, N., de Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J., & Rosenthal, C. (2016). Chaos engineering. IEEE Software, 33(3), 35–41.

[31] Bernstein, P. A., Hadzilacos, V., & Goodman, N. (1987). Concurrency control and recovery in database systems. Reading, MA: Addison-Wesley.

[32] Bertino, E., & Sandhu, R. (2005). Database security: Concepts, approaches, and challenges. IEEE Transactions on Dependable and Secure Computing, 2(1), 2–19.

[33] Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (Eds.). (2016). Site reliability engineering: How Google runs production systems. O'Reilly Media.

[34] Bola-Sadipe, D., Aliliele, C., & Eboh, E. E. (2019). Conceptual model for strategic accounting systems strengthening financial transparency and regulatory compliance. IRE Journals, 3(6), 535-564. https://doi.org/10.64388/IREV3I6-1715327

[35] Brewer, E. (2012). CAP twelve years later: How the "rules" have changed. Computer, 45(2), 23–29.

[36] Brewer, E. A. (2000). Towards robust distributed systems. Keynote at the 19th ACM Symposium on Principles of Distributed Computing (PODC), Portland, OR.

[37] Burrows, M. (2006). The Chubby lock service for loosely-coupled distributed systems. In Proceedings of the 7th USENIX Symposium on Operating Systems Design and

Implementation (pp. 335–350). USENIX Association.

[38] Chandra, T. D., & Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. Journal of the ACM, 43(2), 225–267.

[39] Coventry, L., & Branley, D. (2018). Cybersecurity in healthcare: A narrative review of trends, threats and ways forward. Maturitas, 113, 48–52.

[40] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., & Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. In Proceedings of the 21st ACM Symposium on Operating Systems Principles (pp. 205–220). ACM.

[41] Dierks, T., & Rescorla, E. (2008). The Transport Layer Security (TLS) protocol version 1.2 (RFC 5246). Internet Engineering Task Force.

[42] Dogbatsey, E. A., Ebhojie, O., & Oyeleye, A. O. (2019). Reconciliation control and financial workflow redesign in emerging market organizations: A conceptual framework. Zenodo. https://doi.org/10.5281/zenodo.20447103

[43] Dosunmu, A. A., & Ogundele, P. O. (2019). Security audit and enterprise risk assessment frameworks for resilient information systems. Iconic Research and Engineering Journals, 3(5), 434-447. https://doi.org/10.64388/IREV3I5-1713225

[44] Dwork, C., Lynch, N., & Stockmeyer, L. (1988). Consensus in the presence of partial synchrony. Journal of the ACM, 35(2), 288–323.

[45] Ekemezie, U. C., Ohunyon, G., & Sydney, E. (2018). Evaluating IDSR implementation in Benue State, Nigeria: DHIS2 data quality and surveillance capacity. Iconic Research and Engineering Journals, 2(6).

[46] European Parliament and Council. (2016). Regulation (EU) 2016/679 (General Data Protection Regulation). Official Journal of the European Union, L 119, 1–88.

[47] Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability: Limitations, advances and future research opportunities. Iconic Research and Engineering Journals, 2(12), 602-617. https://doi.org/10.64388/IREV2I12-1713779

[48] Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., & Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. Journal of Biomedical Informatics, 46(3), 541–562.

[49] Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2), 374–382.

[50] Ford, D., Labelle, F., Popovici, F. I., Stokely, M., Truong, V.-A., Barroso, L., Grimes, C., & Quinlan, S. (2010). Availability in globally distributed storage systems. In Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (pp. 61–74). USENIX Association.

[51] Gideon, E. N., Adaramola, T. S., & Fadero, S. (2019). Reliability engineering and failure analysis in complex engineering systems. Iconic Research and Engineering Journals, 2(11), 703-727. https://doi.org/10.64388/IREV2I11-1722490

[52] Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, 33(2), 51–59.

[53] Gill, P., Jain, N., & Nagappan, N. (2011). Understanding network failures in data centers: Measurement, analysis, and implications. In Proceedings of the ACM SIGCOMM 2011 Conference (pp. 350–361). ACM.

[54] Gray, C., & Cheriton, D. (1989). Leases: An efficient fault-tolerant mechanism for distributed file cache consistency. In Proceedings of the 12th ACM Symposium on Operating Systems Principles (pp. 202–210). ACM.

[55] Gray, J. (1986). Why do computers stop and what can be done about it? In Proceedings of the 5th Symposium on Reliability in Distributed Software and Database Systems (pp. 3–12). IEEE Computer Society.

[56] Gray, J., & Reuter, A. (1993). Transaction processing: Concepts and techniques. San Mateo, CA: Morgan Kaufmann.

[57] Gray, J., Helland, P., O'Neil, P., & Shasha, D. (1996). The dangers of replication and a solution. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (pp. 173–182). ACM.

[58] Gunawi, H. S., Hao, M., Suminto, R. O., Laksono, A., Satria, A. D., Adityatama, J., & Eliazar, K. J. (2016). Why does the cloud stop computing? Lessons from hundreds of service

outages. In Proceedings of the Seventh ACM Symposium on Cloud Computing (pp. 1–16). ACM.

[59] Huang, P., Guo, C., Zhou, L., Lorch, J. R., Dang, Y., Chintalapati, M., & Yao, R. (2017). Gray failure: The Achilles' heel of cloud-scale systems. In Proceedings of the 16th Workshop on Hot Topics in Operating Systems (pp. 150–155). ACM.

[60] Ilodigwe, L., & Adesemoye, A. C. (2019a). A critical review of health insurance financing mechanisms and provider payment reform strategies for balancing coverage expansion and financial protection in emerging markets. International Journal of Health and Pharmaceutical Research, 5(2), 31-55. https://doi.org/10.56201/ijhpr.vol.5.no2.2019.pg31.55

[61] Ilodigwe, L., & Adesemoye, A. C. (2019b). From molecules to health systems: A conceptual model explaining why scientific innovation fails to improve patient outcomes without effective healthcare delivery infrastructure. International Journal of Health and Pharmaceutical Research, 5(2), 56-79. https://doi.org/10.56201/ijhpr.vol.5.no2.2019.pg56.79

[62] International Council for Harmonisation. (2016). Integrated addendum to ICH E6(R1): Guideline for good clinical practice E6(R2). ICH.

[63] International Organization for Standardization. (2012). Societal security: Business continuity management systems: Requirements (ISO 22301:2012). ISO.

[64] International Organization for Standardization. (2013). Information technology: Security techniques: Information security management systems: Requirements (ISO/IEC 27001:2013). ISO.

[65] Kilbridge, P. (2003). Computer crash: Lessons from a system failure. New England Journal of Medicine, 348(10), 881–882.

[66] Kruse, C. S., Frederick, B., Jacobson, T., & Monticone, D. K. (2017). Cybersecurity in healthcare: A systematic review of modern threats and trends. Technology and Health Care, 25(1), 1–10.

[67] Ladapo, O. O., Dosunmu, A. A., Jooda, D., & Abolaji, T. O. (2018). Lessons learned from offline assessment of security-critical systems: The case of Microsoft Active Directory. Iconic Research and Engineering Journals, 2(6), 277-299. https://doi.org/10.64388/IREV2I6-1717205

[68] Ladapo, O. O., Jooda, D., Dosunmu, A. A., & Abolaji, T. O. (2019). Implementation of Active Directory for efficient management of enterprise networks. Iconic Research and Engineering Journals, 3(4), 608-627. https://doi.org/10.64388/IREV3I4-1717206

[69] Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7), 558–565.

[70] Lamport, L. (1998). The part-time parliament. ACM Transactions on Computer Systems, 16(2), 133–169.

[71] Lamport, L. (2001). Paxos made simple. ACM SIGACT News, 32(4), 51–58.

[72] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3), 382–401.

[73] Larsen, E., Fong, A., Wernz, C., & Ratwani, R. M. (2018). Implications of electronic health record downtime: An analysis of patient safety event reports. Journal of the American Medical Informatics Association, 25(2), 187–191. https://doi.org/10.1093/jamia/ocx057

[74] Lawal, O. A., & Oduleye, T. E. (2019). Conceptualizing data driven executive decision systems for strategic financial planning. Iconic Research and Engineering Journals, 3(3).

[75] Liu, V., Musen, M. A., & Chou, T. (2015). Data breaches of protected health information in the United States. JAMA, 313(14), 1471–1473.

[76] Marcus, E., & Stern, H. (2003). Blueprints for high availability (2nd ed.). Wiley.

[77] Martin, G., Martin, P., Hankin, C., Darzi, A., & Kinross, J. (2017). Cybersecurity and healthcare: How safe are we? BMJ, 358, j3179. https://doi.org/10.1136/bmj.j3179

[78] Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Oluoha, O. M. (2018). A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. Iconic Research and Engineering Journals, 2(2), 207-226. https://doi.org/10.64388/IREV2I2-1714911

[79] Mbonu, I. S., Aliliele, C., Uzoka, E., & Oluoha, O. M. (2019). A review of comparative data protection regulations and secure cloud implementation strategies across jurisdictions. Iconic Research and Engineering Journals, 2(9), 482-501. https://doi.org/10.64388/IREV2I9-1714912

[80] Mbonu, I. S., Iwuanyanwu, U., Uzoka, E., & Oluoha, O. M. (2019). Advances in enterprise log analytics and automated incident response architectures using Python and SIEM platforms. Iconic Research and Engineering Journals, 3(2), 1000-1019. https://doi.org/10.64388/IREV3I2-1714915

[81] Microsoft. (2019a). Accelerated database recovery. SQL Server documentation. Redmond, WA: Microsoft Corporation.

[82] Microsoft. (2019b). Always On availability groups: A high-availability and disaster-recovery solution. SQL Server documentation. Redmond, WA: Microsoft Corporation.

[83] Microsoft. (2019c). Configure a multi-subnet Always On availability group and listener. SQL Server documentation. Redmond, WA: Microsoft Corporation.

[84] Mockapetris, P. (1987). Domain names: Implementation and specification (RFC 1035). Internet Engineering Task Force.

[85] Mohan, C., Haderle, D., Lindsay, B., Pirahesh, H., & Schwarz, P. (1992). ARIES: A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. ACM Transactions on Database Systems, 17(1), 94–162.

[86] Mosleh, A., Rasmuson, D. M., & Marshall, F. M. (1998). Guidelines on modeling common-cause failures in probabilistic risk assessment (NUREG/CR-5485). U.S. Nuclear Regulatory Commission.

[87] National Institute of Standards and Technology. (2008). An introductory resource guide for implementing the HIPAA Security Rule (NIST SP 800-66 Rev. 1). Gaithersburg, MD: NIST.

[88] National Institute of Standards and Technology. (2013). Security and privacy controls for federal information systems and organizations (NIST SP 800-53 Rev. 4). Gaithersburg, MD: NIST.

[89] Nwangwu, C. V., Duvbiama-Owasanoye, O. A., & Onwumere, A. (2018). A framework for operational integrity in high-risk drilling: Balancing productivity and safety compliance. Iconic Research and Engineering Journals, 2(5), 436-453.

[90] Obetta, P. C. (2018). A risk-based inspection concept for fitness-for-service assessment of ageing pressure vessels on offshore joint venture facilities. Iconic Research and Engineering Journals, 2(5), 497-520. https://doi.org/10.64388/IREV2I5-1722578

[91] Obetta, P. C. (2019). Reliability and integrity management of fired heaters, boilers and furnaces in upstream oil and gas: A review. Iconic Research and Engineering Journals, 3(5), 599-626. https://doi.org/10.64388/IREV3I5-1722580

[92] Obetta, P. C., & Ninduwezuor-Ehiobu, N. (2018). A condition-based maintenance concept for CALM buoy bearings and swivels to improve offshore crude export availability. Iconic Research and Engineering Journals, 2(6), 385-410. https://doi.org/10.64388/IREV2I6-1722579

[93] Obetta, P. C., & Ninduwezuor-Ehiobu, N. (2019). Integrity assurance of FSO turret mooring systems: A review of inspection, monitoring and life-extension strategies. Iconic Research and Engineering Journals, 3(6), 599-629. https://doi.org/10.64388/IREV3I6-1722581

[94] Obriki, O. D., & Arumosoye, O. M. (2018). Conceptual modeling of data-driven occupational safety risk control in large-scale energy infrastructure projects. Iconic Research and Engineering Journals, 1(7), 169-189. https://doi.org/10.64388/IREV1I7-1714414

[95] Odejobi, O. D., & Ahmed, K. S. (2018). Performance evaluation model for multi-tenant Microsoft 365 deployments under high concurrency. IRE Journals, 1(11), 92-107.

[96] Odejobi, O. D., Hammed, N. I., & Ahmed, K. S. (2019). Approximation complexity model for cloud-based database optimization problems. IRE Journals, 2(9), 1-10.

[97] Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2018). Optimizing laboratory spatial planning strategies to improve diagnostic accuracy, safety, and clinical throughput. Iconic Research and Engineering Journals, 2(1), 87-113. https://doi.org/10.64388/IREV1I7-1713587

[98] Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2019). Regulatory compliant design systems for molecular and pathology laboratories in highly controlled environments. Iconic Research and Engineering Journals, 3(4), 607-631. https://doi.org/10.64388/IREV3I4-1713589

[99] Okonkwo, C. S., Ogunwole, O., & Okeke, O. T. (2018). Model for inventory availability and plant uptime improvement in energy facilities. Iconic Research and Engineering Journals, 2(4), 160-172. https://doi.org/10.64388/IREV2I4-1713120

[100] Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In Proceedings of the 2014 USENIX Annual Technical Conference (pp. 305–319).

[101] Oshoba, T. O., Hammed, N. I., & Odejobi, O. D. (2019). Secure identity and access management model for distributed and federated systems. IRE Journals, 3(4), 550-567.

[102] Popa, R. A., Redfield, C. M. S., Zeldovich, N., & Balakrishnan, H. (2011). CryptDB: Protecting confidentiality with encrypted query processing. In Proceedings of the 23rd ACM Symposium on Operating Systems Principles (pp. 85–100). ACM.

[103] Rausand, M., & Høyland, A. (2004). System reliability theory: Models, statistical methods, and applications (2nd ed.). Wiley.

[104] Reason, J. (1990). Human error. Cambridge University Press.

[105] Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. Proceedings of the IEEE, 63(9), 1278–1308.

[106] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. Computer, 29(2), 38–47.

[107] Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. ACM Computing Surveys, 22(4), 299–319.

[108] Schroeder, B., & Gibson, G. A. (2007). Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In Proceedings of the 5th USENIX Conference on File and Storage Technologies (pp. 1–16). USENIX Association.

[109] Sittig, D. F., & Singh, H. (2010). A new sociotechnical model for studying health information technology in complex adaptive healthcare systems. Quality and Safety in Health Care, 19(Suppl. 3), i68–i74.

[110] Swanson, M., Bowen, P., Phillips, A. W., Gallup, D., & Lynes, D. (2010). Contingency planning guide for federal information systems (NIST Special Publication 800-34 Rev. 1). National Institute of Standards and Technology.

[111] Trivedi, K. S. (2002). Probability and statistics with reliability, queuing, and computer science applications (2nd ed.). Wiley.

[112] U.S. Department of Health and Human Services. (2013). HIPAA Security Rule, 45 CFR Part 160 and Subparts A and C of Part 164.

[113] U.S. Food and Drug Administration. (1997). Electronic records; electronic signatures (21 CFR Part 11). Federal Register, 62(54), 13430–13466.

[114] U.S. Food and Drug Administration. (2003). Guidance for industry: Part 11, electronic records; electronic signatures, scope and application. Silver Spring, MD: FDA.

[115] Unamma, U. M.-L., Okwesa, I. J., Iwuanyanwu, U., & Ajiga, D. (2018). An offline-first design framework for digital learning in low-connectivity environments. Iconic Research and Engineering Journals, 2(3), 192-211. https://doi.org/10.64388/IREV2I3-1722627

[116] Vogels, W. (2009). Eventually consistent. Communications of the ACM, 52(1), 40–44.

[117] Wang, Y., Coiera, E., Gallego, B., Concha, O. P., Ong, M.-S., Tsafnat, G., Roffe, D., Jones, G., & Magrabi, F. (2016). Measuring the effects of computer downtime on hospital pathology processes. Journal of Biomedical Informatics, 59, 308–315.

[118] Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., & Alonso, G. (2000). Understanding replication in databases and distributed systems. In Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (pp. 464–474). IEEE.

[119] Yuan, D., Luo, Y., Zhuang, X., Rodrigues, G. R., Zhao, X., Zhang, Y., Jain, P. U., & Stumm, M. (2014). Simple testing can prevent most critical failures: An analysis of production failures in distributed data-intensive systems. In Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (pp. 249–265). USENIX Association.

[120] Yusuf, H. D., & Abdulrazaq, K. (2018). A conceptual framework for strengthening diagnostic quality systems in resource-limited clinical laboratories. Iconic Research and Engineering Journals, 1(11), 145-179. https://doi.org/10.64388/IREV1I11-1723090