© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711764-2065

Disaster Recovery as Code: Building Self-Healing Financial Infrastructures in the Public Cloud

OREOLUWA OMOIKE

Abstract- ***Cloud computing is becoming a crucial component in the financial technology (FinTech) industry, providing digital financial services that are always-on, secure, and scalable. But this reliance also brings about a weakness that traditional disaster recovery (DR) is unable to address. Manual recovery processes, offline backups and hard copy documentation cannot match the real-time financial processes that require constant access and compliance with regulations. In this paper, the author is advancing the idea of Disaster Recovery as Code (DRaaC), a code-based, automation-driven approach which combines Infrastructure as Code (IaC), policy automation, and intelligent observability to make public cloud environments resilient and self-healing. Based on the recent studies on self-healing infrastructures (Thota, 2025; Padamati, 2023; Vankayalapati & Pandugula, 2022) and the theory of resilience, DRaaC is framed as a technical and organisational paradigm, which instantiates the logic of recovery into the programmable workflow. The framework will fit the reliability requirements of FinTech systems to the speed and flexibility of current DevOps practices, and provide an adaptive framework that combines automation, compliance and socio-technical governance.***

Keywords: Disaster Recovery as Code, Infrastructure as Code, FinTech Resilience, Self-Healing Systems, Cloud Computing, Automation

I. INTRODUCTION

The development of financial technology has changed the way financial services are provided and delivered value. Public clouds like Amazon Web Services and Google Cloud, and Microsoft Azure are now being used by institutions to execute their critical workloads. This model is both fast, elastic and economical, but introduces dependencies and risks. Once the cloud region becomes unresponsive or a deployment misconfiguration is realised, the impact is local and global. Recovery cannot take hours; it has to occur in minutes in the case of FinTech companies with millions of transactions every day (Abieba et al., 2025).

The conventional disaster recovery frameworks, which are based on stagnant documentation and manual-based implementation, are not able to accommodate these requirements. They are highly dependent on human coordination, do not have much visibility in dynamic systems, and are not tested as frequently as they should to ensure that they perform as required in case of crisis (Duvvada, 2025). According to Thota (2025), distributed architectures are volatile, which makes them demand recovery systems capable of autonomous thinking, decision-making, and action. This is not a simple case of uptime, as in the FinTech context, it is a question of trust, adherence, and system stability.

Concurrently, other laws, including the Digital Operational Resilience Act (DORA), PCI DSS, and ISO 22301, require financial organisations to ensure that their service integrity remains intact, and that they have audited recovery processes and have proven resilience (Stamenkov, 2024; PCI Security Standards Council, 2013). Resilience toolkits are currently provided by cloud providers, yet most FinTech companies continue to find it challenging to operationalise these features in coordination with complicated infrastructures.

This gap is bridged by Disaster Recovery as Code (DRaaC). Rather than considering disaster recovery as a standalone contingency plan, DRaaC makes it a part of the infrastructure deployment process. Recovery scenarios, such as provisioning of backups, orchestration of failover as well and recovery of data, are coded and incorporated in CI/CD pipelines, which automates the recovery, versions it and verifies it on the fly (Amazon Web Services, 2025). More so, it helps FinTech companies view resiliency as a dynamic process and not a reactive response. This paper examines DRaaC as a socio-technical model. It integrates automation technologies and organisational governance and learning systems to make resilience both technically and institutionally sustainable.

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711764-2065

II. THEORETICAL FOUNDATIONS: A SOCIO-TECHNICAL LENS ON RESILIENCE

2.1 Socio-Technical Systems and Resilience

FinTech resilience cannot be brought down to software or servers, but rather is a result of the interaction of people, technology, and processes. The socio-technical systems (STS) approach, which was initially theorised by Trist and Bamforth (1951), is one of many reasons why most organisations perform optimally when both the technical and social aspects develop in tandem with one another. This, in the context of cloud-native FinTech processes, is the harmonisation of automation systems, regulatory policies, and human expertise into a common control framework.

An example of such alignment is Disaster Recovery as Code, which is an automated recovery and nevertheless requires human intervention; the code is determined by engineers, auditors check adherence, and managers read the findings. Such interdependence is what Steinmann et al. (2024) define as a quality of interactions, rather than infrastructure. The feedback loops of the system, the technical, and the human, are fast, clear, and adaptive, making the system resilient.

Resilience engineering builds on this by focusing on learning and adapting to disturbances (Hollnagel, 2016). DRaaC operationalizes this through the mechanisms of learning in code, like recovery scripts which fail in testing are automatically marked and improved on the next run, and the system can then continue to improve. Over time, this will lessen the cognitive load of the engineer and enhance the predictability of automated recovery behaviour.

2.2 Adaptive Recovery and Systems Thinking

Contemporary FinTech systems are networks of systems, interrelated APIs, data streams, and third-party services. Their strength relies on feedback and adaptation. These interdependencies can be explained through systems thinking. Instead of viewing failures as single occurrences, systems thinking views failures as feedbacks, which point to where corrections are necessary (Bhattacharjee et al., 2024).

DRaaC is a closed adaptive system. Observable tools identify anomalies, analytics process the information, automation scripts activate recovery processes, and the output is returned to subsequent settings. This is a reflection of the cybernetic control loops of resilience literature (Thota, 2025). Continuous sensing and adjusting of such nature makes disaster recovery a dynamic checklist into a live ability.

Notably, this ad hoc loop has to operate within the parameters of financial regulation and ethical control. Automated systems will not do without governance. They need policy limits which are coded in compliance-as-code to make sure that recovery measures are not unlawful and auditable (Padamati, 2023). In cases where recovery logic is held in such rules, resilience is measurable and responsible, and technological adjustment is in line with institutional trust.

III. LITERATURE REVIEW

3.1 Infrastructure as Code and Cloud Automation

Infrastructure as code (IaC) has revolutionised the way cloud systems are managed because infrastructure is now declarative, reproducible, and testable. Rather than manually defining each server, engineers write all of the system architecture in the form of code through tools like Terraform, AWS CloudFormation, and Azure Bicep. This solution eradicates configuration drift and can roll back and scale quickly (Abieba et al., 2025).

The most significant contribution of IaC to the disaster recovery field is the determinism ability, which is the ability to reproduce the same environments at any time. The traditional recovery procedure was founded on manual backup and backup procedures, which were slow and prone to errors. They are converted into repeatable pipelines by IaC, and the same deployment code that generates production can generate it in a new location within a few minutes (AWS, 2023).

Furthermore, recent studies show that the integration of IaC with version control and testing models enables the teams to test disaster recovery playbooks on an ongoing basis (Duvvada, 2025). This is to make sure recovery stays in line with infrastructure changes. DRaaC is based on this principle and takes the recovery itself as code by making not only the

configuration but also the response logic part of automated workflows. This manual for programmable recovery is one of the fundamental changes in operational resilience.

3.2 AI-Driven Observability and Self-Healing Systems

Infrastructure management has put observability as a core focus in the modern day. In contrast to the traditional monitoring that simply follows predefined metrics, observability enables systems to obtain internal states by logging from logging, trace information, and telemetry on the fly (Nangunori, 2024). Observability is an essential element of FinTech, as the trust relies directly on the reliability of latency and service, which requires visibility to identify anomalies, preventing them into outages.

AI increases observability through predictive incident management. Performance data on the extent of resource saturation, unlike a partial network degradation, and similar issues can be detected, and response measures are automatically initiated because machine learning models were trained with these types of data (Vankayalapati & Pandugula, 2022). Self-healing architectures operate based on this predictability, and examples include system-level intuition that will diagnose and correct issues.

According to Thota (2025), the system of self-healing is three-phase, with three nonstop processes involved in it: detection, diagnosis, and healing. In DRaaC, these stages are enshrined within recovery scripts, in accordance with which the response varies in accordance with particular triggers. To illustrate these, in cases where a database replica fails, the system will automatically create a new instance and match data automatically without involving a human being. This automation positively impacts the mean-time-to-recovery (MTTR) and the entire service continuity.

The AI-based observability, however, comes with new governance issues. Models to achieve automation to claim accurate results need constant retraining, and they are overly dependent on automation systems, since the latter will result in addiction beyond human control. According to the research conducted by Steinmann et al. (2024), although self-healing mechanisms increase the responsiveness of such decisions, they disregard the transparency of the decisions when it is not accompanied by human intervention. DRaaC thus has to strike a balance between autonomy and accountability such that all automated recoveries are explainable as well as auditable.

3.3 Compliance-as-Code in FinTech Resilience

FinTech systems are conducted within a very high regulatory regime that regulates data sovereignty, the security of transactions, and the availability of services. Compliance-as-Code is a new approach to enforcing these regulations in the logic of infrastructure (Google Cloud, 2024). Compliance rules are represented as executable policies rather than depending on post-deployment audits, which constantly check the configurations against standards such as PCI DSS, GDPR, and ISO 27001 (PCI SSC, 2013)

In DRaaC, compliance-as-code makes recovery processes comply with legal and organisational limitations even in an automated fashion. As an example, it can be limited to failover operations to a particular jurisdiction in order to satisfy data residency. Open Policy Agent and HashiCorp Sentinel are the tools that allow such policies to be operated automatically as a part of CI/CD pipelines (Stamenkov, 2024).

Padamati (2023) argues that compliance validation, when implemented together with automated recovery, will enforce legal compliance, as well as promote trust in automation. Regulators of FinTech are becoming more demanding in proving control of resiliency mechanisms. This is guaranteed by codified compliance, which fills the gap between accountability and innovation.

3.4 Gaps in Literature

While automation of disaster recovery has been gaining ground rapidly, the current research is still disjointed. The majority of the studies dwell on particular aspects, IaC, observability, or AI, yet none provide a cohesive framework that links them within the context of an organisation. In particular, the socio-technical perspective is under-researched.

Current frameworks do not discuss the manner in which automation redefines human positions, risk management, and institutional learning. In addition, there are unanswered ethical issues on automated

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711764-2065

decision-making in high-stakes settings such as finance. With automation assuming some of the most important recovery roles, it is necessary to learn how to ensure that human accountability and interpretability are preserved. Disaster Recovery as Code is an emerging idea that bridges this gap by integrating technological automation with the theory of governance and provides a comprehensive perspective on how FinTech organisations can make resilience in the public cloud operational.

IV. DISASTER RECOVERY AS CODE (DRaaC): A FRAMEWORK FOR FINTECH CLOUDS

4.1 Core Principles

Disaster Recovery as Code is an extension of the Infrastructure as Code logic to the business continuity concept. It is based on three principles, namely codification, automation, and adaptation. To begin with, codification transforms the recovery policies into code artefacts. All backup, replication and failover instructions are in declarative syntax such that the whole recovery process is reproducible and versioned. Second, these recovery codes are implemented by automation using continuous integration pipelines. This removes manual work, lessens downtime, and guarantees that there is uniformity among development, staging, and production settings (Abieba et al., 2025). Thirdly, adaptation brings in intelligence. The system improves its recovery behaviour with the help of observability information and AI-driven analytics, depending on real-time performance and historical incidents (Thota, 2025). This learning loop converts DR to be a dynamic capability as opposed to being a fixed contingency..

4.2 Architecture and Feedback Loops

A DRaaC system is functional and consists of four layers that are interdependent, including infrastructure, observability, orchestration, and governance. The resources and dependencies are defined by IaC templates in the infrastructure layer. It defines areas of replication, data retention and network settings required to ensure continuity. The observability layer checks the health of the system through the logs, metrics, and distributed tracing. In case of anomalies, this layer is used to signal the orchestration layer to initiate corrective workflow. In the orchestration layer, response logic is carried out. It implements recovery resources, reroutes traffic and checks restored services based on serverless functions or Kubernetes operators. Finally, the governance layer ensures compliance policies and logs all the actions to make them auditable. It provides that automated recovery complies with internal and regulatory standards (Stamenkov, 2024).

These layers create a closed feedback system that is closed. Every incident produces data that feeds subsequent set-ups, enabling the self-enhancement of the system as time goes by. A simplified conceptual model may be represented in the form of a loop that travels around Detection → Decision → Recovery → Verification → Learning.

V. ORGANIZATIONAL AND SOCIO-TECHNICAL IMPLICATIONS

DRaaC implementation is not a technical task but a process that requires adjustment of the organisation. Automation changes the power dynamics in the FinTech teams, reshaping the power dynamic between engineers, compliance officers, and the AI systems. This reconfiguration presents some advantages and strains from a socio-technical perspective. Automation helps minimise cognitive and human error, yet it also helps to hide responsibility in case of failures. Teams thus need to come up with new governance systems that integrate automated response and clarity of responsibility by humans.

Training becomes critical. The engineers should be fluent in code and compliance, and the management should be aware of the scope of automation. Having a common language among the technical and regulatory stakeholders will lead to the joint optimisation that Trist and Bamforth (1951) state to be the nature of successful socio-technical systems.

A culture of resilience through design DRaaC promotes learning how to respond to near-misses and recovery simulations, which will become a daily routine operation. Quarterly chaos engineering, popularised by Netflix and implemented by large FinTech companies, is consistent with the principles of DRaaC. With the deliberate introduction of failures, organisations can test their recovery code and make gains in trust in automation.

VI. CHALLENGES AND FUTURE RESEARCH

Despite the fact that DRaaC promises to bring a lot of benefits in terms of resilience, it also brings forth new challenges that have to be addressed by academicians and practitioners alike. Model risk is one. The observability and decision-making machine learning algorithms can misunderstand the signals and cause unnecessary or harmful recovery measures (Vankayalapati & Pandugula, 2022). To ensure trust, continuous mechanisms of validation and explainability are necessary.

Furthermore, there is a limitation with the quality of data. Predictive recovery involves full and unsaturated telemetry; a shortfall of the data can affect the performance and reliability (Padamati, 2023). Also, Duvvada (2025) and Syed (2025) argue that strategic risk is vendor lock-in. Several FinTech companies rely on proprietary cloud services; thus, they find it hard to migrate or transfer DRaaC systems between providers. This can be alleviated by research on cross-cloud orchestration and open recovery standards.

Finally, the policy must be innovated due to the ethical and legal concerns, including the appropriate degree of human control, the transparency of audit, and the responsibility of the automated decisions. The future directions of explainable resilience automation should be based on a framework of researching how to use AI ethics and regulatory design.

VII. CONCLUSION

To FinTech executives and system architects, the Disaster Recovery as Code (DRaaC) offers a concrete way to achieve greater operational resilience. Recovery process automation reduces the cost of downtime and compliance risks, but the real effectiveness is determined by the integration of these systems between teams. The initial one is to formalise key recovery processes into versioned templates and incorporate automated validation tests into CI/CD pipelines. With every iteration, reliability is enhanced and manual intervention is minimised.

Also, compliance-as-code should be embedded at the beginning to make sure that all automated recoveries are in line with financial regulations regarding data privacy, encryption, and geographic sovereignty. This integration transforms regulatory requirements into proactive elements of resilience as opposed to post-hocs.

Furthermore, leaders must understand that DRaaC is a cultural change as much as a technical change. The cooperation of DevOps, compliance, and risk management teams creates a learning environment in which resilience is tested, refined, and verified.

Finally, DRaaC redefines disaster recovery as a smart, dynamic system. It substitutes fixed contingency plans with ongoing, code-based assurance. Combining Infrastructure as Code, AI-based observability, and compliance automation, FinTech organisations can find a balance between efficiency and trust. Resilience is not a response to failure but a natural aspect of day-to-day operations-systems that recover, adapt, and get better as naturally as they are deployed.

REFERENCES

[1] Abieba, O. A., Alozie, C. E., & Ajayi, O. O. (2025). Enhancing disaster recovery and business continuity in cloud environments through infrastructure as code. *Journal of Engineering Research and Reports*, *27*(3), 127–136. https://doi.org/10.9734/jerr/2025/v27i31423

[2] Amazon Web Services. (2023). *AWS Resilience Hub now available in the AWS GovCloud (US) Regions.* https://aws.amazon.com/about-aws/whats-new/2023/10/aws-resilience-hub-govcloud-regions/

[3] Amazon Web Services. (2023a, October 19). AWS Resilience Hub now available in the AWS GovCloud (US) Regions. Retrieved from https://aws.amazon.com/about-aws/whats-new/2023/10/aws-resilience-hub-govcloud-regions/

[4] Amazon Web Services. (2025). *AWS Well-Architected Framework*. https://docs.aws.amazon.com/pdfs/wellarchitected/latest/framework/wellarchitected-framework.pdf

[5] Bhattacharjee, I., Srivastava, N., Mishra, A., Adhav, S., & Singh, N. (2024). The rise of FinTech: Disrupting traditional financial services. *Educational Administration: Theory*

and Practice, *30*(4), 89–97. https://doi.org/10.53555/kuey.v30i4.1408

[6] Duvvada, V. (2025). Automated business continuity and disaster recovery in hybrid cloud enterprise architectures. *European Modern Studies Journal*, *9*(3), 439–448. https://doi.org/10.59573/emsj.9(3).2025.39

[7] Google Cloud. (2024). *PCI-DSS compliance overview*. https://cloud.google.com/security/compliance/pci-dss

[8] Hollnagel, E. (2016). Resilience engineering: A new understanding of safety. *Journal of the Ergonomics Society of Korea*, *35*(3), 185–191. https://doi.org/10.5143/JESK.2016.35.3.185

[9] Nangunori, S. K. (2024). Leveraging AI in disaster recovery: The future of business continuity. *International Journal of Research in Computer Applications and Information Technology,* 7(2), 1675–1687. https://ijrcait.com/index.php/home/article/view/IJRCAIT_07_02_130

[10] Padamati, J. R. (2023). AI-driven self-healing infrastructure: The next frontier in scalable cloud deployments. *International Journal for Advanced Research in Science & Technology*, *13*, 506–517.

[11] PCI Security Standards Council. (2013). *PCI DSS cloud computing guidelines*. https://www.pcisecuritystandards.org/pdfs/PCI_DSS_v2_Cloud_Guidelines.pdf

[12] Stamenkov, G. (2024). Cloud service models, business continuity and disaster recovery plans, and responsibilities. *International Journal of Organizational Analysis*. Advance online publication. https://doi.org/10.1108/ijoa-12-2023-4127

[13] Steinmann, F., Tobi, H., & van Voorn, G. (2024). Resilience metrics for socio-ecological and socio-technical systems: *A scoping review. Systems,* 12(9), 357. https://doi.org/10.3390/systems12090357

[14] Syed, A. (2024). Disaster recovery and data backup optimization: Exploring next-gen storage and backup strategies in multi-cloud architectures. *International Journal of Emerging Research in Engineering and Technology*, *5*(3), 32–42. https://doi.org/10.63282/3050-922X.IJERET-V5I3P104

[15] Thota, P. K. (2025). Demystifying self-healing cloud architectures: Building resilient systems for modern applications. *World Journal of Advanced Engineering Technology and Sciences*, *15*, 2211–2218. https://doi.org/10.30574/wjaets.2025.15.1.0426

[16] Trist, E., & Bamforth, K. (1951). Some social and psychological consequences of the longwall method of coal-getting. *Human Relations*, *4*(1), 3–38. https://doi.org/10.1177/001872675100400101

[17] Vankayalapati, R. K., & Pandugula, C. (2022). AI-powered self-healing cloud infrastructures: A paradigm for autonomous fault recovery. *Migration Letters*, *19*(6), 1173–1187. https://doi.org/10.2139/ssrn.5052024