Vol. 7, No. 4, December 2025

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Empirical Evaluation of Decentralized Genomic Data Computation Using Bacalhau and IPFS

Bagas Triaji[1], Badiyanto[2], Justivan Intifadhah Afif[3]

[1,2,3]Information Technology Faculty, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia

Email: bagastriaji@utdi.ac.id[1], badiyanto@utdi.ac.id[2], justivan.intifadhah24@students.utdi.ac.id[3]

Received: Oct 23, 2025

Revised: Nov 5, 2025

Accepted: Nov 21, 2025

Published: Dec 18, 2025

Corresponding Author:

Author Name*:

Bagas Triaji1, Badiyanto

Email*:

bagastriaji@utdi.ac.id

DOI:

10.63158/journalisi.v7i4.1311

Abstract. Large-scale genomic analysis typically relies on centralized infrastructures, creating conflicts between collaborative needs and data sovereignty regulations. This study solves this dilemma by evaluating a decentralized architecture designed to facilitate secure, inter-institutional genomic computation without moving raw data. We integrated Bacalhau for orchestration and IPFS Cluster with CRDT consensus for storage, employing AES-256 encryption. A quantitative evaluation was conducted on AWS using five t3.medium nodes to simulate a resource-constrained hospital network. We tested three scenarios: a centralized baseline (SSH+SCP), an ideal decentralized workflow, and a "chaos" scenario involving active network fault injection. While the centralized baseline was the fastest (Mean=37.69s), the decentralized architecture incurred a manageable ~30% overhead under ideal conditions (Mean=49.22s, SD=1.58s). Critically, under chaos fault injection, although execution time increased to 90.67s (SD=17.84s), the system achieved a superior 100% job completion rate compared to the fragile baseline. This research quantifies the trade-off between execution speed and system resilience in a healthcare context. We demonstrate that this architecture prioritizes data sovereignty and high availability over raw speed, offering a proven model for privacy-critical Decentralized Science (DeSci) collaborations.

Keywords: Decentralized System, IPFS Cluster, Genomic, Bacalhau, Fault Tolerance

1. INTRODUCTION

The availability of public genomic data from initiatives such as the 1000 Genomes Project has become a cornerstone of modern biomedical research, enabling population-scale studies on genetic variation [1]. These datasets, commonly distributed in Variant Call Format (VCF), contain rich information about genomic variants but require complex, multi-stage computational workflows even for a single chromosome, such as Chromosome 22. Traditional analysis pipelines typically rely on centralized infrastructure or cloud-based systems [2], which, while scalable, introduce challenges when dealing with multi-institutional genomic collaborations.

When genomic data originate from multiple healthcare institutions, the problem extends beyond computational scalability to critical issues of data sovereignty, privacy, and governance [3]. In many jurisdictions, including Indonesia's Personal Data Protection (PDP) Law No. 27 of 2022, sensitive patient data cannot legally be transferred outside institutional boundaries. This legal restriction, combined with ethical and organizational barriers, often prevents centralized aggregation of genomic datasets [4]. This tension between the need for large-scale collaborative analysis and the mandate for data locality highlights a fundamental challenge in biomedical data processing.

To overcome these challenges and explicitly preserve institutional data sovereignty and security, the emerging paradigm of Decentralized Science (DeSci) proposes the Compute-over-Data (CoD) model, where computation is brought to where the data reside rather than moving the data themselves. Recent developments in decentralized technologies such as the InterPlanetary File System (IPFS) for distributed storage and Bacalhau as a decentralized job orchestration platform embody this CoD principle [5]. These systems aim to preserve data locality and minimize trust dependencies, offering robustness against single-point failures [6]. To further ensure confidentiality and compliance, the integration of cryptographic layers such as AES-256 encryption or other privacy-preserving methods like homomorphic encryption has been proposed [4], [7]. This encryption-based approach allows secure computation and storage without exposing raw genomic data, making it suitable for inter-institutional analyses.

However, while decentralized infrastructures such as IPFS clusters and Bacalhau provide a promising foundation for secure and resilient collaborative computation, quantitative evaluations of their performance trade-offs remain scarce. Existing literature has focused primarily on conceptual frameworks and privacy mechanisms [8], [9]. However, comprehensive empirical studies that systematically assess the combined computational overhead of Bacalhau orchestration, IPFS Cluster CRDT, and AES-256 encryption remain limited. Consequently, quantitative data regarding their performance trade-offs in real-world scenarios is still scarce. Therefore, this study aims to contribute to this body of knowledge by providing a comprehensive end-to-end performance evaluation of a secure, decentralized genomic analysis workflow. Specifically, we quantify the trade-offs in speed and resilience by comparing the baseline centralized approach with the decentralized architecture (Bacalhau + IPFS Cluster CRDT + AES-256) under simulated network disruptions. The focus is on measuring latency, throughput, and resilience under network disruptions while maintaining strong data confidentiality guarantees.

2. METHODS

To empirically evaluate the performance, security, and resiliency of the proposed decentralized architecture, a multi-stage methodology was designed. This section details the components of this methodology, outlining: (1) the genomic data and analysis workflow used for the computation, (2) the design of the simulated inter-institutional test infrastructure, (3) the specific configuration of the Bacalhau and IPFS Cluster architecture, (4) the design of the three comparative test scenarios, and (5) the methods used for data collection and statistical analysis.

2.1. Data and Analysis Workflow

This studi use data source from Genomic data originated from the 1000 Genomes Project, filtered for the Chromosome 22 sample. The raw 11 GB VCF file was converted to Apache Parquet format to optimize I/O performance for locality-aware orchestration [10], resulting in a 12 MB analysis file named sample_chr22.parquet. The python script computational workflow implements a pipeline adapted for decentralized genomic environments [11], performing four key analyses: (1) QC Filtering, (2) Variant Type Aggregation, (3) Rare Variant Filtering, and (4) Ti/Tv Ratio.

The computational workflow, executed via a Python script, performs four analyses: Analysis 1: QC Filtering, Analysis 2: Variant Type Aggregation, Analysis 3: Rare Variant Filtering, Analysis 4: Ti/Tv Ratio [12], [13]. Details of this workflow can be seen in Figure 1.

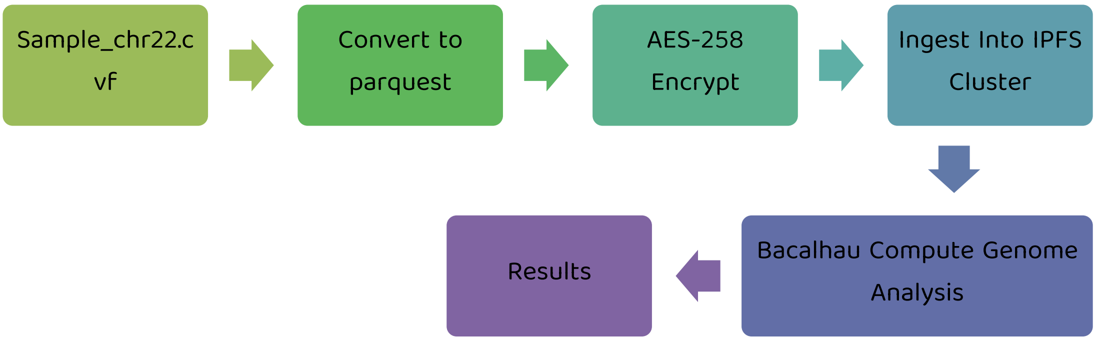

Figure 1. Genomic Analysis Workflow Diagram

2.2. Test Infrastructure Design: Simulating an Inter-Institutional Network

The experimental testbed was deployed on Amazon Web Services (AWS) using five EC2 instances. We specifically selected the t3.medium instance type (2 vCPU, 4 GiB Memory) to represent a standard, resource-constrained computational environment typical of edge nodes in hospital IT infrastructures, rather than high-performance computing (HPC) clusters. This choice ensures the baseline performance metrics remain conservative and realistic for widespread adoption.

The experimental setup simulated an inter-institutional collaborative network through the logical isolation of nodes representing distinct entities (e.g., Hospital A, Research Center B). Within this baseline framework, nodes were hosted in the same Virtual Private Cloud (VPC) and communicated via private IP addresses. This approach minimized external network jitter, creating a controlled environment to rigorously isolate the computational overhead generated by the Bacalhau orchestration, IPFS Cluster consensus, and AES-256 encryption layers. The operating system used was Ubuntu 22.04 LTS with Docker installed to containerize the Bacalhau and IPFS services.

2.3. Architecture Configuration

The decentralized storage layer utilizes the IPFS Cluster with Merkle-CRDT (Conflict-free Replicated Data Type) consensus [14]. Unlike Raft consensus which prioritizes strong consistency, CRDT was chosen for its partition tolerance and high availability—critical features for loose federations of health institutions where network reliability cannot be guaranteed. The CRDT mechanism ensures eventual consistency; if a network partition occurs, updates are tracked locally and merged automatically ($S_{merged} = S_A \cup S_B$) once connectivity is restored, preventing data loss.

For data security, we implemented AES-256 (Advanced Encryption Standard with 256-bit keys) in CBC mode. As genomic data falls under strict privacy regulations (e.g., GDPR, UU PDP), AES-256 serves as the industry-standard symmetric encryption layer to guarantee confidentiality before any data leaves the trusted institutional boundary (prior to IPFS ingress). This ensures that even if the encrypted chunks on the public IPFS network are intercepted, the raw genomic information remains inaccessible without the private key. Operationally, to guarantee data sovereignty, the decryption keys are never transmitted across the network; they are held exclusively by the data-owning institution's node, ensuring that access control remains strictly local.

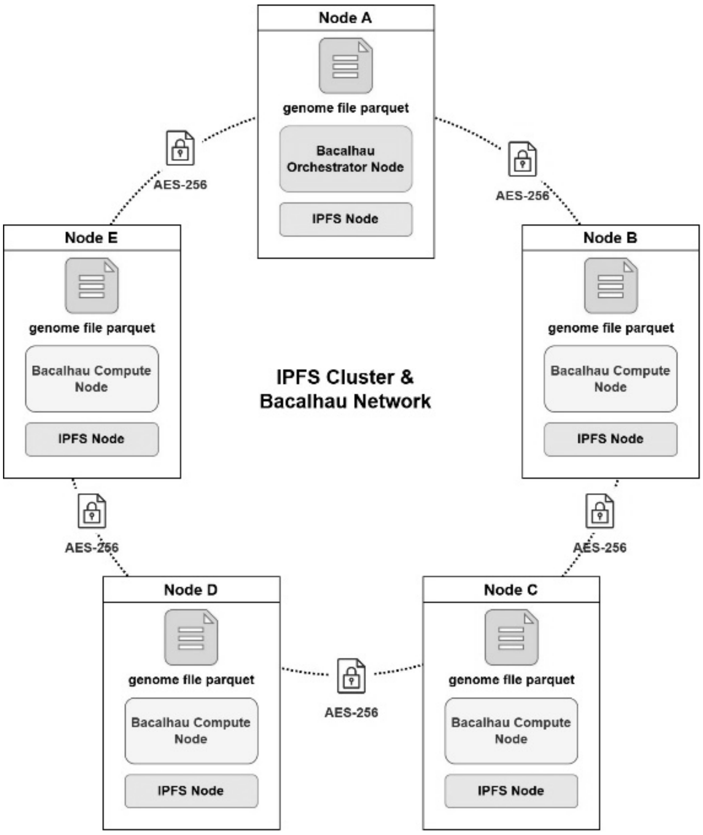

Figure 2. 5-Node System Architecture Diagram. Each node simulates one participating Health Institution

Figure 2 illustrates the decentralized infrastructure designed for collaborative genomic analysis. This system comprises five nodes (A-E), where each node represents a different health institution.

1) Node A (Institution A): Functions as the Bacalhau Orchestrator and Client Node. It initiates the workflow by encrypting input data (AES-256 Encryption) and submitting the compute job.
2) Nodes B, C, D, E (Institutions B-E): Act as Bacalhau Compute Nodes.
3) IPFS Cluster Peers (CRDT): Each node participates as a peer in the IPFS Cluster, using CRDT consensus. This ensures encrypted data added by Node A is replicated across all participating institutions, providing high availability and eventual consistency even if some nodes are temporarily disconnected.
4) NATS Communication: Bacalhau uses NATS to orchestrate job requests between the Orchestrator (A) and Compute Nodes (B-E).
5) Workflow Execution: When a job is submitted (e.g., to Node B), the Compute Node retrieves the necessary encrypted data replica from its local IPFS peer, performs AES-256 Decryption, and executes the genomic analysis workflow inside a Docker Container. Results are then generated.

2.4. Test Scenario Design

Scenario 1: Traditional Baseline (SSH+SCP)

1) Workflow: Manual scp of the parquet file to Node B → ssh into Node B → docker run the analysis.
2) Purpose: To measure the fastest possible performance without orchestration overhead, simulating analysis on a single institution's data in isolation.

Scenario 2: Bacalhau with AES-256 Encryption (No Disruption)

1) Workflow*:* scp .parquet file to Node A → AES-256 Encrypt on Node A → ipfs-cluster-ctl add encrypted file → bacalhau docker run job (which includes decryption and analysis).
2) Purpose*:* To measure the collaborative architecture's performance, including the combined overhead of orchestration, IPFS, and encryption, under ideal conditions.

Scenario 3: Bacalhau Chaos Test (With Encryption & Disruption)

1) Workflow: Identical to Scenario 2, but with simulated network disruption.

2) Disruption Simulation: An automation script runs in parallel. Every 5 seconds, it randomly selects a target compute node n from the set $\{B, C, D, E\}$ using Uniform Random Selection with probability $P(n)=0.25$. The script then blocks NATS/IPFS ports on the selected node using iptables, simulating transient network partitions. This technique implements the principles of Network Fault Injection [15] and is commonly applied in chaos engineering experiments [16],[17]. This method follows the standard fault-injection approach for assessing system resilience in distributed environments, where random or probabilistic node disruptions are used to evaluate fault tolerance and recovery mechanisms [18][19].
3) Purpose: To measure the resilience of the encrypted collaborative system during simulated network disruptions between institutions.

Data Collection and Analysis Execution time and Total Workflow Time were recorded in seconds. Data (N=5 per scenario) from the *.csv files were analyzed for Mean and Standard Deviation (SD).

3. RESULTS AND DISCUSSION

3.1. Quantitative Performance Results

The quantitative evaluation of the system was conducted across three scenarios. The results of each scenario are presented in the Tables 1. Table 1 presents the baseline performance using the traditional centralized workflow (SSH/SCP).

Table 1. Traditional workflow test results (baseline)

Run	SCP Upload	Compute	Total Workflow
1	12.23	24.18	36.41
2	11.95	24.9	36.85
3	12.52	28.62	41.14
4	12.26	24.39	36.65
5	12.2	25.22	37.42

Table 2 details the execution time for the decentralized architecture under ideal network conditions, broken down by phase (Encryption, IPFS Ingest, Compute).

Table 2. Decentralized architecture test results without disruption

Run	SCP Upload	AES-256 Encryption	Ingest IPFS	Bacalhau Compute	Overhead	Total Workflow
1	13.12	3.28	3.82	27.26	3.13	50.61
2	12.09	3.25	4.96	25.22	3.09	48.61
3	12.12	3.15	3.46	27.28	3.38	49.39
4	12.23	3.24	3.6	24.51	3.27	46.85
5	12.44	3.42	3.56	27.73	3.50	50.65

Table 3 presents the results under "Chaos" conditions (Scenario 3), where network disruptions were introduced.

Table 3. Decentralized architecture test results with disruption

Run	SCP Upload	AES-256 Encryption	Ingest IPFS	Bacalhau Compute	Overhead	Total Workflow
1	12.63	3.27	47.63	25.90	3.23	92.66
2	12.78	3.64	14.55	25.32	3.63	59.92
3	12.22	3.23	40.25	39.41	3.38	98.49
4	12.42	3.23	51.79	34.87	3.49	105.80
5	12.29	3.43	50.92	26.57	3.28	96.49

A consolidated statistical comparison of the total workflow performance across all scenarios is summarized in Table 4.

Table 4. Statistical comparison of total workflow performance

No	Test Scenarios	Methods	Encryption	Disruption	Avg. Total Time (sec.)	Standard Deviation (sec.)
1	Baseline	SSH & SCP	No	No	37.69	1.96
2	Bacalhau AES-256	Bacalhau & IPFS	Yes	No	49.22	1.58

No	Test Scenarios	Methods	Encryption	Disruption	Avg. Total Time (sec.)	Standard Deviation (sec.)
3	Chaos Test AES-256	Bacalhau & IPFS	Yes	Yes	90.67	17.84

3.2. Visual Performance Analysis

To allow for a direct comparison of the scenarios summarized in Table 4, Figure 3 illustrates the mean execution times. While the baseline SSH approach demonstrates the lowest latency, the decentralized scenarios introduce measurable overhead.

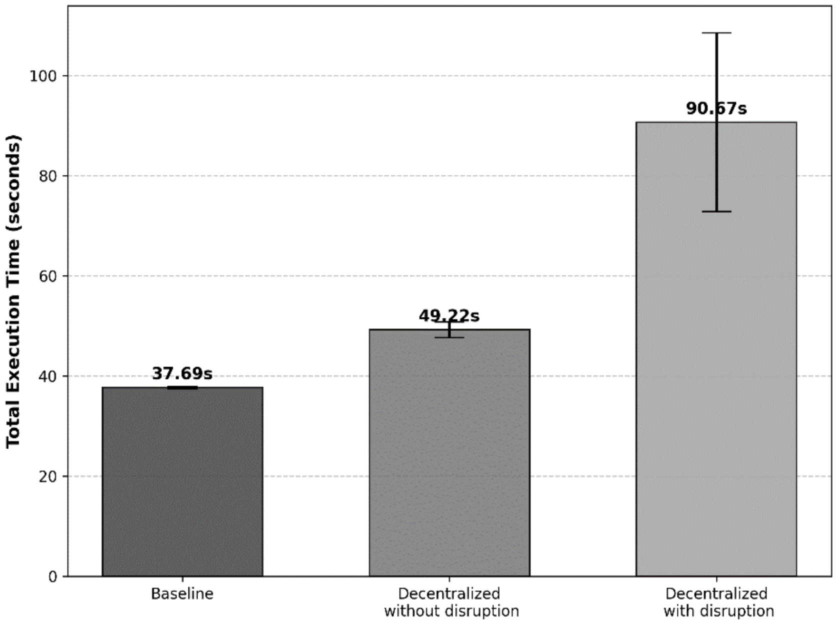

Figure 3. Comparison of Total Workflow Execution Time across Three Scenarios.

To visually analyze these performance differences, Figure 3 illustrates the mean execution times. As observed in the chart, the significantly larger variance (indicated by the wider error bars) in the Chaos scenario is directly attributed to the network fault injection, which forced the system into variable retry loops. To further investigate the source of the latency in the decentralized scenarios, Figure 4 breaks down the time components. This visualization explicitly addresses the impact of network disruption on specific workflow stages.

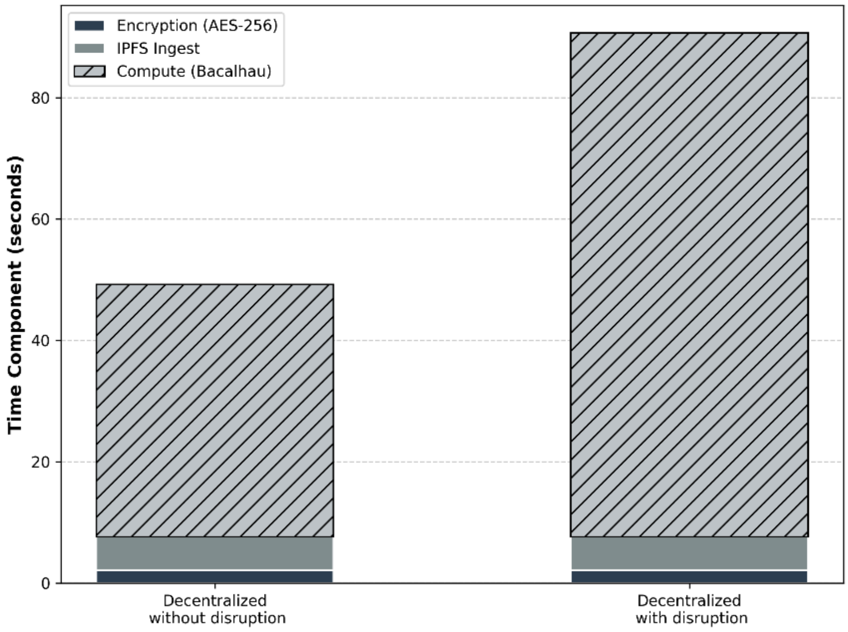

Figure 4. Component analysis of latency overhead

As shown in Figure 4, the stacked bar chart highlights that while encryption and IPFS ingest times remain stable, the Compute Phase (striped area) increases drastically during the Chaos Test. This implies that the delay is not caused by the storage layer, but specifically due to orchestration retries and timeout mechanisms in the compute layer when nodes become unreachable.

3.3. Verification of Job Execution

Following the performance tests, the system's ability to successfully complete the genomic analysis job was verified. The following log output from the Bacalhau compute phase confirms that the containerized job executed successfully even under the decentralized constraints:

```
--- ANALYSIS 1 RESULTS (QC Filter) ---
Number of high-quality variants (QUAL > 50): 1,103,547
--- ANALYSIS 2 RESULTS (Variant Types) ---
Total SNPs: 1,059,727
Total Indels: 43,820
--- ANALYSIS 3 RESULTS (Rare Variants) ---
Number of rare variants (AF < 0.01): 905,197
```

```
--- ANALYSIS 4 RESULTS (Ti/Tv Ratio) ---
Transitions (Ti): 747,927
Transversions (Tv): 311,800
Ti/Tv Ratio: 2.3987
```

3.4. Discussion

The quantitative results highlight a distinct trade-off between speed and system resilience. The baseline centralized workflow (Scenario 1) is approximately 23% faster than the ideal decentralized workflow (Scenario 2). As shown in Figure 4, this expected baseline overhead is primarily driven by the IPFS Ingest phase (~5.5s) and Encryption (~2.1s). This aligns with findings on distributed orchestration costs, where service coordination inherently adds latency compared to monolithic execution [20].

However, Scenario 3 (Chaos Test) incurs a significantly higher performance penalty (Mean=90.67s). The component analysis in Figure 4 reveals that Bacalhau Compute time is the specific area most impacted. This is a structural consequence of the resilience mechanism: when network ports are blocked, the orchestration layer triggers retry loops to re-establish connectivity via NATS [20], and the IPFS Cluster attempts to heal the CRDT state [21]. This observation aligns with recent comparative studies [22], which confirm that while decentralized protocols offer superior data integrity, they inherently incur higher I/O latency during network instability compared to centralized systems. Despite this delay, the system achieved a 100% completion rate (0% failure), proving that the architecture successfully prioritized task completion over speed.

Addressing scalability, while this study utilized five nodes, the architecture is designed to be horizontally scalable. For larger inter-institutional deployments, we recommend a tiered topology to manage consensus traffic. Future implementations could also explore integrating blockchain-based access logs for enhanced auditability [23].

Regarding the trade-offs, this decentralized model is not suitable for critical real-time healthcare applications, such as emergency monitoring, where milliseconds count. Instead, it offers a strategic alternative for collaborative genomic research, where Data Sovereignty and Privacy outweigh the need for immediate processing speed [24], [25]. The additional latency observed in Scenario 3 is an acceptable architectural cost for ensuring high availability while maintaining strict data confidentiality. Although encrypted data chunks are distributed across the cluster for resilience, the raw genomic information remains mathematically inaccessible to unauthorized peers, ensuring that data sovereignty is preserved via cryptographic control rather than strict physical isolation.

4. CONCLUSION

This study empirically validated a collaborative decentralized architecture for genomic analysis using IPFS Cluster and Bacalhau. The quantitative evaluation demonstrates that while this distributed approach introduces a performance overhead—approximately 1.3x slower under ideal conditions and 2.4x slower under network disruptions—it offers superior resilience (100% job completion) compared to traditional workflows. By utilizing ubiquitous data replication combined with AES-256 encryption, the system ensures that data remains available and confidentially secure even during inter-institutional connection failures.

However, this study has limitations. The evaluation was restricted to a homogenous five-node cluster within a single Virtual Private Cloud (VPC) environment using private IP addresses. Consequently, the measured latency represents a best-case scenario, and real-world inter-institutional deployments over public WANs would likely experience higher network latency. Despite this constraint, the findings have significant practical implications beyond genomics. The validated architecture is highly adaptable for other data-intensive healthcare domains, such as Medical Imaging (e.g., MRI/CT scans) and Electronic Health Record (EHR) analytics, where establishing a centralized data lake is often hindered by regulatory or trust barriers.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Direktorat Penelitian dan Pengabdian Masyarakat, Direktorat Jenderal Riset dan Pengembangan, Kementerian Pendidikan Tinggi, Sains dan Teknologi Indonesia for providing research funding through the Penelitian Dosen Pemula (PDP) grant scheme in the 2025 fiscal year.

REFERENCES

[1] M. Bourgey *et al.*, "GenPipes: An open-source framework for distributed and scalable genomic analyses," *Gigascience*, vol. 8, no. 6, Jun. 2019, doi: 10.1093/gigascience/giz037.

[2] B. Liu *et al.*, "Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses," *J Biomed Inform*, vol. 49, pp. 119–133, 2014, doi: 10.1016/j.jbi.2014.01.005.

[3] M. Beyene, P. A. Toussaint, S. Thiebes, M. Schlesner, B. Brors, and A. Sunyaev, "A scoping review of distributed ledger technology in genomics: Thematic analysis and directions for future research," Aug. 01, 2022, *Oxford University Press*. doi: 10.1093/jamia/ocac077.

[4] T. Zhao, F. Wang, R. Mott, J. Dekkers, and H. Cheng, "Using encrypted genotypes and phenotypes for collaborative genomic analyses to maintain data confidentiality," *Genetics*, vol. 226, no. 3, Mar. 2024, doi: 10.1093/genetics/iyad210.

[5] P. Kang, W. Yang, and J. Zheng, "Blockchain Private File Storage-Sharing Method Based on IPFS," *Sensors*, vol. 22, no. 14, Jul. 2022, doi: 10.3390/s22145100.

[6] Y. Zhang, M. Zhong, X. Zhao, C. Curtis, X. Li, and C. Chen, "Enabling privacy-preserving sharing of genomic data for GWASs in decentralized networks," in *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, Association for Computing Machinery, Inc, Jan. 2019, pp. 204–212. doi: 10.1145/3289600.3290983.

[7] T. T. Kuo *et al.*, "iDASH secure genome analysis competition 2018: blockchain genomic data access logging, homomorphic encryption on GWAS, and DNA segment searching," Jul. 21, 2020, *BioMed Central Ltd*. doi: 10.1186/s12920-020-0715-0.

[8] D. Copeland and A. Taylor, "A novel encryption protocol for facilitating de-identification of genomics health data," *Int J Popul Data Sci*, vol. 9, no. 5, Sep. 2024, doi: 10.23889/ijpds.v9i5.2907.

[9] M. Shabani, "Blockchain-based platforms for genomic data sharing: a de-centralized approach in response to the governance problems?," Jan. 01, 2019, *Oxford University Press*. doi: 10.1093/jamia/ocy149.

[10] A. A. Corodescu *et al.*, "Locality-aware workflow orchestration for big data," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Nov. 2021, pp. 62–70. doi: 10.1145/3444757.3485106.

[11] G. Gürsoy, C. M. Brannon, E. Ni, S. Wagner, A. Khanna, and M. Gerstein, "Storing and analyzing a genome on a blockchain," *Genome Biol*, vol. 23, no. 1, Dec. 2022, doi: 10.1186/s13059-022-02699-7.

[12] R. P. Adelson *et al.*, "Empirical design of a variant quality control pipeline for whole genome sequencing data using replicate discordance," *Sci Rep*, vol. 9, no. 1, Dec. 2019, doi: 10.1038/s41598-019-52614-7.

[13] S. N. Kobren *et al.*, "Commonalities across computational workflows for uncovering explanatory variants in undiagnosed cases," *Genetics in Medicine*, vol. 23, no. 6, pp. 1075–1085, Jun. 2021, doi: 10.1038/s41436-020-01084-8.

[14] P. S. Almeida, "Approaches to Conflict-free Replicated Data Types," *ACM Comput Surv*, vol. 57, no. 2, Nov. 2024, doi: 10.1145/3695249.

[15] D. Cotroneo, L. De Simone, and R. Natella, "ThorFI: a Novel Approach for Network Fault Injection as a Service," *Journal of Network and Computer Applications*, vol. 201, May 2022, doi: 10.1016/j.jnca.2022.103334.

[16] A. Basiri *et al.*, "Chaos Engineering," *IEEE Softw*, vol. 33, no. 3, pp. 35–41, May 2016, doi: 10.1109/MS.2016.60.

[17] C. Diekmann, L. Hupel, J. Michaelis, M. Haslbeck, and G. Carle, "Verified iptables Firewall Analysis and Verification," *J Autom Reason*, vol. 61, no. 1–4, pp. 191–242, Jun. 2018, doi: 10.1007/s10817-017-9445-1.

[18] W. Hoarau, S. Tixeuil, and F. Vauchelles, "Fault Injection in Distributed Java Applications."

[19] R. Chandra, R. M. Lefever, K. Joshi, M. Cukier, and W. H. Sanders, "A Global-State-Triggered Fault Injector for Distributed System Evaluation *."

[20] P. Singhal, "Orchestration Workflows in Distributed Systems: A Systematic Analysis of Efficiency Optimization and Service Coordination." [Online]. Available: www.ijfmr.com

[21] D. Trautwein *et al.*, "Design and evaluation of ipfs: A storage layer for the decentralizedweb," in *SIGCOMM 2022 - Proceedings of the ACM SIGCOMM 2022 Conference*, Association for Computing Machinery, Inc, Aug. 2022, pp. 739–752. doi: 10.1145/3544216.3544232.

[22] O. A. Lajam and T. A. Helmy, "Performance evaluation of IPFS in private networks," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Feb. 2021, pp. 77–84. doi: 10.1145/3456146.3456159.

[23] S. Ma, Y. Cao, and L. Xiong, "Efficient logging and querying for blockchain-based cross-site genomic dataset access audit," *BMC Med Genomics*, vol. 13, Jul. 2020, doi: 10.1186/s12920-020-0725-y.

[24] R. Hariharan, "Resilience Engineering in Distributed Cloud Architectures," *International Journal of Engineering and Architecture*, vol. 2, no. 1, pp. 39–75, May 2025, doi: 10.58425/ijea.v2i1.355.

[25] G. Mandinyenya and V. Malele, "Comparative Security and Performance Evaluation of IPFS and Filecoin for Off-chain Blockchain Storage," *The Indonesian Journal of Computer Science*, vol. 14, no. 4, Aug. 2025, doi: 10.33022/ijcs.v14i4.4968.