Blockchain-Enabled Framework for Efficient and Interoperable Proteomic Data Management

Victor Ikenna Kanu, Josiah Isong, Simeon Okechukwu Ajakwe*, Taesoo Jun, Dong-Seong Kim
Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea
*ICT Convergence Research Centre, Kumoh National Institute of Technology, Gumi, South Korea
(kanuxavier, isongjosiah, simeonajlove)@gmail.com, (taesoo.jun, dskim)@kumoh.ac.kr

2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN) | 979-8-3315-2487-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICUFN65838.2025.11169963

***Abstract*—The management and integration of proteomic data remain challenging due to heterogeneous formats, reliance on centralized storage, and restricted interoperability across platforms. While databases like the Protein Data Bank (PDB) and UniProt enhance data organization and accessibility, they struggle with scalability, security, and decentralized data exchange. This study presents a blockchain-driven framework leveraging Hyperledger Fabric and InterPlanetary File System (IPFS) to overcome these limitations. The system employs microservices for API interactions, encryption, and access control, ensuring secure and interoperable bioinformatics data management. Performance assessments, including Locust load testing and Hyperledger Caliper benchmarking, showcased scalability, achieving peak throughputs of 197 TPS (1,000 TxNum) and 206.6 TPS (2,000 TxNum) with stable latency. Validating human insulin protein data demonstrated seamless synchronization between on-chain and off-chain storage, reinforcing data integrity and traceability. Future improvements will focus on real-time protein data uploads, improving scalability for large datasets, and expanding applications in drug discovery to support secure and scalable proteomics system automation.**

***Index Terms*—Blockchain, bioinformatics, data sharing, data storage, microservices, proteomics.**

I. INTRODUCTION

Proteomic data integration presents significant challenges due to the increasing volume and heterogeneity of datasets. Centralized repositories such as the PDB, ProteomicsDB [1], and iProX [2] facilitate data aggregation but suffer from issues like redundancy, restricted interoperability, and security weaknesses. PDB lacks uniform annotation standards, while ProteomicsDB faces computational limitations in real-time proteomic data analysis [1], [3], impeding cross-disciplinary research that requires seamless data integration.

Frameworks like jPOST and iProX, developed within the ProteomeXchange consortium, have enhanced metadata consistency and streamlined data submission processes [4]. However, they struggle with real-time synchronization and managing complex heterogeneous data sets [5]. These challenges underscore the need for a scalable and unified framework capable of securely integrating diverse proteomic platforms. Works on multi-omics data integration highlight the importance of harmonizing datasets and ensuring seamless accessibility across platforms [6] via tamper-proof and reliable security.

Blockchain technology (BT) provides a decentralized solution for proteomic data management, minimizing dependence on centralized repositories while improving security and interoperability [7], [8]. Advanced cryptographic techniques, such as AES-256 encryption, safeguard sensitive data during storage and transmission [9], while role-based access control [10] ensures fine-grained authorization for users. This study introduces a blockchain-based framework that integrates bioinformatics repositories for data ingestion, automates validation, and enforces secure access policies [11] thereby addressing lapses in the existing frameworks. By leveraging REST APIs, smart contracts, and authentication mechanisms like OAuth and HTTPS, the proposed system strengthens data integrity, enhances accessibility, and enables real-time synchronization, effectively addressing key challenges in proteomic research. To achieve this, this study made the following contributions.

Hence, the specific contributions of this study are:

- Developed a tamper-proof and latency-friendly approach capable of handling large proteomic data with high throughput.
- Designed an interoperable and efficient data-sharing mechanism for seamless data integration, authenticity, and reliable accessibility via metadata harmonization.
- Scalable performance evaluation through load testing and benchmarking to handle real-world scenarios and deployments.
- Established a transparent and innovative security architecture for advancing a cohesive blockchain-driven proteomic data ecosystem via ubiquitous computing.

The rest of this paper captures Section II as Related Work and Section III as System Design, while Section IV discusses the Results, and the Conclusion and Future Work is presented in Section V.

II. BACKGROUND AND RELATED WORKS

The exponential growth of bioinformatics data spanning genomics, proteomics, and clinical research has introduced significant obstacles to seamless integration and interoperability [12]. Data fragmentation, heterogeneous formats, and restricted access impede large-scale analytics and interdisciplinary collaboration. Traditional centralized data management frameworks often lack the scalability, security, and reliability to support evolving biomedical research demands. In response, solutions incorporating Semantic Web technologies [13], FAIR principles, and blockchain-based architectures have emerged to enhance data accessibility and security [14]. However, these approaches still face challenges in achieving full interoperability, computational efficiency, and metadata harmonization,

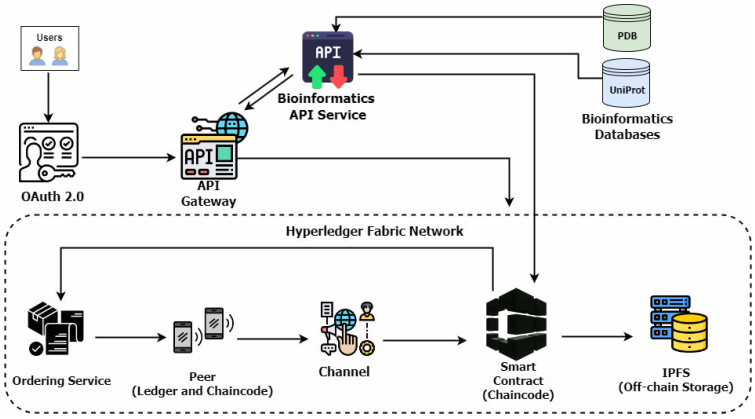

Fig. 1. Flow of interactions in a Hyperledger Fabric network, showcasing components like the Certificate Authority, Orderer, Peer Nodes, and Chaincode, along with their interactions to manage proteomic data securely and efficiently

highlighting the need for further advancements toward scalable and cohesive bioinformatics ecosystems.

Several studies have investigated strategies to overcome challenges in bioinformatics data integration. Precision omics frameworks [15] enhance interoperability by leveraging standardized ontologies for structured data sharing and analysis, though their adaptability across diverse domains remains a limitation. Likewise, Semantic Web technologies [13] facilitate large-scale biomedical data linkage via Linked Open Data (LOD) networks, improving cross-domain machine learning applications. However, issues such as semantic heterogeneity, complex SPARQL queries, and accessibility constraints hinder their broader adoption. Additionally, work on biomedical repositories [16] underscores challenges related to high storage costs, metadata complexity, and privacy concerns when managing large-scale proteomic and clinical datasets.

Though BT is a promising solution to guarantee tamper-proof and traceable data security in bioinformatics, its adoption and integration are relatively nascent, especially for cohesive proteomic data management [11]. Blockchain-enabled microbial databases by [17] demonstrated the ability of BT to decentralize storage, improve data provenance, and strengthen security. However, issues such as computational overhead, regulatory constraints, and privacy concerns persist in their approach. These limitations highlight the importance of a BT-driven bioinformatics framework that is cost-effective, highly efficient, interoperable, and can handle large-scale heterogeneous proteomic data in a decentralized manner. This work is an innovation to bridge these gaps.

III. SYSTEM DESIGN

The components of the proposed framework include Hyperledger Fabric (HF) as the underlying preferred BT, the microservices architecture, the testing parameters, etc., as described in Fig. 1. A detailed description of each component of the proposed framework is presented forthwith.

A. *HF Cryptographic Model*

HF utilizes cryptographic hash functions to create distinct fingerprints for each data entry, ensuring immutability. This mechanism safeguards data integrity by preventing unauthorized modifications. Equation (1) represents the hash function equation:

$$\mathtt{H}(\mathtt{x}) = \mathtt{y} \tag{1}$$

where, $\mathtt{x}$ is the input data (e.g., a proteomic dataset), $\mathtt{H}(\mathtt{x})$ is the hash output, and $\mathtt{y}$ is a fixed-length unique identifier $\mathtt{x}$.

For the proposed system, each proteomic dataset is hashed using a secure algorithm (e.g., AES-256), and the resulting $\mathtt{H}(\mathtt{x})$ is stored on-chain, while the raw dataset resides off-chain in IPFS. Data encryption ensures the confidentiality of proteomic datasets before being linked to the blockchain. Using the AES-256 algorithm as shown in equation (2):

$$\mathtt{C} = \mathtt{E}_{\mathtt{K}}(\mathtt{P}) \tag{2}$$

where, $\mathtt{P}$ is the plaintext (proteomic data), $\mathtt{H}(\mathtt{K})$ is encryption key (256 bits) and $\mathtt{C}$ is the ciphertext.

Decryption is represented with equation (3):

$$\mathtt{P} = \mathtt{D}_{\mathtt{K}}(\mathtt{C}) \tag{3}$$

Role-Based Access Control (RBAC) mechanisms provide granular permission enforcement. The permissions $\mathtt{R}(\mathtt{u})$ assigned to a user $\mathtt{u}$ are evaluated against the requested dataset $\mathtt{D}$, as shown in Equation (4):

$$\mathtt{R}(\mathtt{u}) = \{\mathtt{p}_1, \mathtt{p}_2, \ldots, \mathtt{p}_n\} \tag{4}$$

Access is granted if the user's role satisfies the required permission, $\mathtt{T}(\mathtt{u}, \mathtt{D}) = \mathtt{Valid}$, expressed in Equation (5):

$$\mathtt{Access}(\mathtt{u}, \mathtt{D}) \equiv \mathtt{p}_i \in \mathtt{R}(\mathtt{u}) \land \mathtt{T}(\mathtt{u}, \mathtt{D}) = \text{Valid} \tag{5}$$

where the logical AND ($\land$) ensures both conditions are met, and the equivalence ($\equiv$) guarantees access is tightly controlled.

This secure and decentralized system also ensures seamless interoperability through smart contracts. These automate data validation and synchronization processes across repositories like PDB and UniProt. Synchronization is achieved by periodically verifying on-chain and off-chain data consistency, with the synchronization condition defined as $\mathtt{H}(\mathtt{D})_{\text{external}} \neq \mathtt{H}(\mathtt{D})_{\text{on-chain}}$. Discrepancies trigger updates to ensure real-time alignment.

The transaction throughput in HF can be calculated as shown in equation (6):

$$\mathtt{T}_{\text{throughput}} = \frac{\mathtt{n}_t}{\Delta \mathtt{t}} \tag{6}$$

where, n_t is the number of successfully committed transactions and Δt is the total elapsed time.

Scalability is assessed by increasing peer nodes and analyzing its effect on n_t, ensuring the system accommodates growing proteomic datasets efficiently. Digital identities issued by Certificate Authorities (CAs) authenticate network participants, while the ordering service structures transactions into blocks for validation and commitment. The Raft consensus mechanism enhances data immutability, traceability, and secure sharing, overcoming limitations of centralized systems.

B. Microservices Development

The development of microservices in this framework integrates Golang and Python to create modular, scalable components for API integration, encryption, and access control. Python, with its non-blocking I/O, is used for API services that interact with external bioinformatics databases like UniProt and PDB, enabling seamless retrieval of proteomic data in standardized formats such as FASTA and PDB. Additionally, Python handles encryption, utilizing AES-256 for securing sensitive datasets and RSA-2048 for key exchange to restrict decryption access to authorized users. A dedicated access control microservice enforces Role-Based Access Control (RBAC), validating user roles against predefined policies stored in a database to ensure secure authentication. Smart contracts, implemented in Go, automate blockchain operations, including data submission and retrieval, verifying user credentials before granting access to the hash or linked data [18], thereby ensuring data integrity and controlled accessibility. These contracts also facilitate data provenance tracking by immutably logging dataset modifications and securely managing encryption keys. The data ingestion pipeline follows a structured workflow where input data D is validated, encrypted, and stored in IPFS, with its cryptographic hash $H(D)$ recorded on the blockchain through smart contract invocation, ensuring data security, integrity, and traceability at every stage.

C. API Integration with UniProt and PDB

The API integration with bioinformatics systems (UniProt and PDB), serves as a critical microservice in our framework, facilitating real-time data retrieval, validation, and synchronization for efficient proteomic data management. Built with Python, this service uses RESTful endpoints to fetch protein sequences, structural data, and metadata, validating them against standard formats like FASTA and PDB. As depicted in Fig. 2, the integration workflow begins with requests to verify or update on-chain data based on external sources. Data retrieved via unique identifiers undergoes schema validation, ensuring consistency through a function $V(D)$, which must return `True` for further processing. A cryptographic hash (SHA-256) is generated for validated datasets, enabling comparison with on-chain records to detect discrepancies. If mismatches arise, encrypted updates via AES-256 secure off-chain storage in IPFS, maintaining data integrity. A synchronization script periodically recalibrates hashes, triggering updates

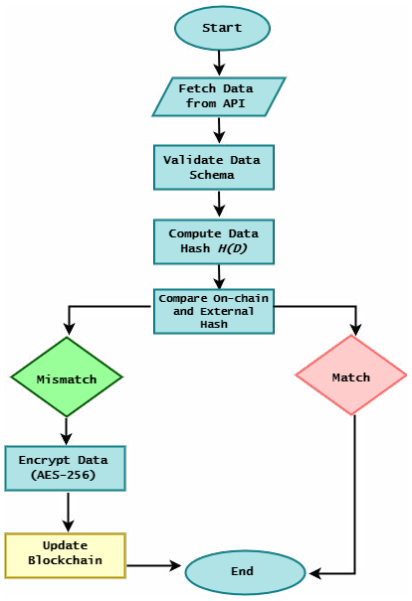

Fig. 2. Workflow for API Integration and Synchronization

when $H(D)_{external} \neq H(D)_{on\text{-}chain}$, ensuring accurate, up-to-date records. This robust integration enhances the system's interoperability, reliability, and security, providing a seamless, high-throughput platform for proteomic data storage and retrieval. Finally, the blockchain is updated with a computed hash and off-chain data reference, ensuring data integrity, security, and real-time synchronization. This process keeps blockchain records aligned with external sources, enabling researchers to access the most recent, verified proteomic data while maintaining immutability and reliability.

D. Front-end Authentication

The front end of the framework was developed using React.js, facilitating dynamic and responsive interactions with backend microservices and the blockchain network. Designed for seamless proteomic data upload, querying, and visualization, the interface adopts a modular component-based structure to enhance reusability and efficiency. React components—including file download modules, dashboards, and authentication forms—communicate with the backend via RESTful APIs, enabling real-time updates with minimal latency. State management is handled using Redux, ensuring session consistency and data integrity under high traffic loads. Secure authentication and access control are enforced through OAuth 2.0, integrating external providers like Google for token-based login and authorization. The system follows an authorization code grant flow, securely managing authentication tokens to maintain access security. Additionally, a dedicated microservice enforces role-based access control (RBAC), restricting sensitive operations like data uploads and retrievals to authorized users. Algorithm 1 outlines the authentication workflow, from user login to token validation. This layered security model ensures a robust yet efficient user experience for researchers interacting with the system.

Algorithm 1: Authentication and Access Control

Data: `User Credentials` : (C); `Resource Request` (R)
Result: `Access Token` : T_{access}; `Authorization Decision` :
Initialize (C), (R)
Begin:
while *True* **do**
  Redirect `User` to OAuth provider's authentication
  `User` authenticates and consents to share identity information
  Receive Authentication Code C_{auth}
  Exchange C_{auth} for Access Token T_{access} via backend server
  Validate T_{access} with OAuth provider
  Forward T_{access} to Access Control Microservice
  Verify User Role $R(U)$ against Requested Resource R
  if $R(U)$ `==` `R` **then**
    Grant `Access`
  end
  else
    Deny `Access`
  end
end

E. *Testing and Security Audits*

Testing and security audits in this proposed framework involved rigorous evaluation of individual components and system-wide performance to ensure functional accuracy, data integrity, and security. Each microservice underwent unit testing to validate functionality across diverse scenarios, while scalability and performance were assessed using Locust, a Python-based load testing tool that stressed the API layer, a critical intermediary for data ingestion and retrieval between the framework and external databases such as PDB and UniProt. Docker containerization played a key role in isolating microservices, ensuring consistent, reproducible testing environments, and enabling real-time resource monitoring for CPU and memory usage optimization. For blockchain components, Hyperledger Caliper was employed to benchmark transaction throughput, latency, and resource utilization under varied workloads, validating the system's efficiency in handling high transaction volumes and low-latency queries for proteomic data storage. The network configuration, detailed in Table I, included two organizations, each with one peer, one orderer, and resource constraints of 1 vCPU and 2 GB RAM. Security audits extended to authentication and access control, with RBAC simulations ensuring unauthorized users could not access sensitive datasets and OAuth 2.0 testing verifying secure token expiration and refresh handling. Data integration tests employed hash-based comparisons to detect discrepancies between synchronized datasets from external bioinformatics repositories and on-chain records, reinforcing system reliability and consistency.

TABLE I
DEFAULTS CONFIGURATIONS FOR ALL EXPERIMENTS

Parameters	Values
BatchSize	
Max Message Count	10.
AbsoluteMaxBytes	99 MB.
PreferredMaxBytes	512 KB.
Number of Channels	1.
State Database	levelDB.
Network Configuration	2 organizations, each with 1 peer.
Peer Resources	1 vCPU (2.3GHz), 2 GB RAM.
Endorsement Policy	AND.
Block Timeout	2 sec.
Number of Orderers	1.

IV. SIMULATION RESULT AND DISCUSSION

This section evaluates the proposed framework through simulations and benchmarking, assessing blockchain performance using Hyperledger Caliper for transaction throughput, latency, and fault tolerance. API and functionality tests using a PDB code to validate data synchronization and retrieval accuracy. The analysis highlights system strengths, addresses limitations, and suggests optimizations for bioinformatics applications. A prototype integrating these technologies was developed to demonstrate the operations of this framework, which supports data ingestion, access control, and retrieval, with its documentation publicly available at [19].

A. *BT Performance Evaluation*

The blockchain performance metrics, evaluated using Hyperledger Caliper as shown in Table II, highlight the system's efficiency in handling store and query operations under varying transaction send rates (100–500 TPS). For the store operations, the 1000-TxNum configuration showed increasing maximum latency as the transaction rate scaled, peaking at 2.79s at 500 TPS, while average latency remained within 2.11s at peak load. Throughput performance ranged from 99.8 TPS at 100 TPS to a maximum of 197 TPS at 500 TPS, indicating the system's capability to handle higher transaction loads effectively. The 2000-TxNum store configuration demonstrated higher computational demand, with max latency rising to 7.42s at 500 TPS and a corresponding increase in average latency (5.77s at peak). Despite this, the throughput remained strong, peaking at 198.3 TPS, reflecting the framework's scalability under increased transaction volumes.

For query operations, performance results showcased a lower computational overhead compared to store operations. The 1000-TxNum query setup exhibited maximum latency peaks at 3.43s at 500 TPS, while the 2000-TxNum setup recorded slightly higher delays, reaching 7.92s at peak send rates. However, query operations maintained relatively stable average latencies across configurations, with the 1000-TxNum setup averaging 1.97s and the 2000-TxNum setup reaching 4.23s at peak load. Throughput remained competitive, with

TABLE II
SUMMARY OF BLOCKCHAIN PERFORMANCE METRICS TESTED USING HYPERLEDGER CALIPER

Name	No. of TXs	Succ	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
Store	1000.	1000.	100, 200, 300, 400, 500	0.39, 2.68, 2.79, 3.29, 2.94	0.03, 0.21, 0.34, 0.56, 0.36	0.10, 1.68, 2.11, 2.38, 1.96	99.8, 160.6, 191.1, 168.7, 197.0
Store(2)	2000	2000	100, 200, 300, 400, 500	0.31, 3.37, 5.50, 4.96, 7.42	0.03, 0.35, 0.42, 0.48, 0.65	0.10, 2.37, 4.07, 3.60, 5.77	99.7, 172.6, 198.3, 206.6, 171.5
Query	1000.	1000.	100, 200, 300, 400, 500	0.46, 1.22, 2.75, 2.54, 3.43	0.04, 0.30, 0.63, 0.54, 0.56	0.11, 0.81, 1.97, 1.93, 2.20	99.6, 185.2, 184.4, 196.3, 193.5
Query(2)	2000	2000	100, 200, 300, 400, 500	0.69, 1.05, 6.39, 5.67, 7.92	0.04, 0.16, 0.22, 0.56, 0.49	0.15, 0.67, 4.61, 4.23, 6.06	99.8, 194.3, 182.6, 194.3, 167.3

the 1000-TxNum setup peaking at 196.3 TPS, while the 2000-TxNum setup sustained a maximum of 194.3 TPS. These results indicate that query operations scale efficiently under higher transaction loads but may require further optimization to mitigate latency increases.

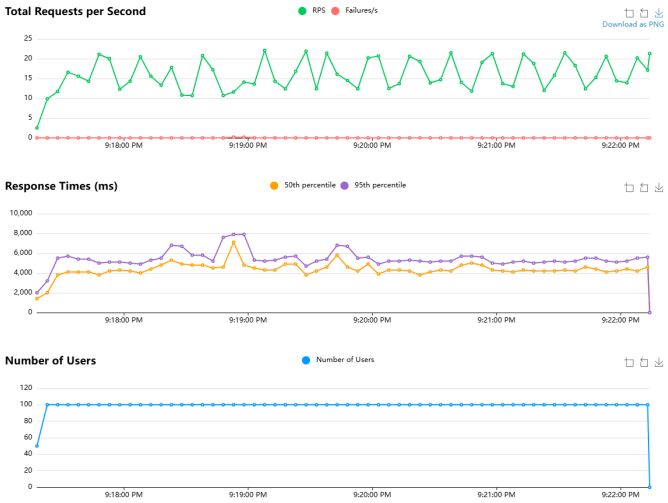

Fig. 3. Configuration of Load Test Parameters in Locust: Setting 100 Concurrent Users with a Ramp-Up Rate of 10 Users Per Second.

B. Locust Load Performance

Two separate tests were conducted to simulate different levels of load executed in a Python environment by sending requests to the API endpoints. The first test simulated 100 users with a ramp-up rate of 10 users per second (Load Test) for 5 minutes, followed by a second test scaling up to a load test of 100 users with a ramp-up rate of 10 users per second for the same duration. The final test pushed the system further by simulating a load test of 1000 users with a ramp-up rate of 100 users per second over 5 minutes.

The Locust load testing results highlight the performance of the system under varying traffic loads. At Load100, API requests to (`/api/v1/protein/1HNY`) and (`/api/v1/protein/P12345`) response times increased significantly to 3685.08 ms and 4575.11 ms, with a small failure rate, as shown in Fig. 3. Load1000 exhibited the highest response times, peaking at 47273.2 ms for aggregated requests,

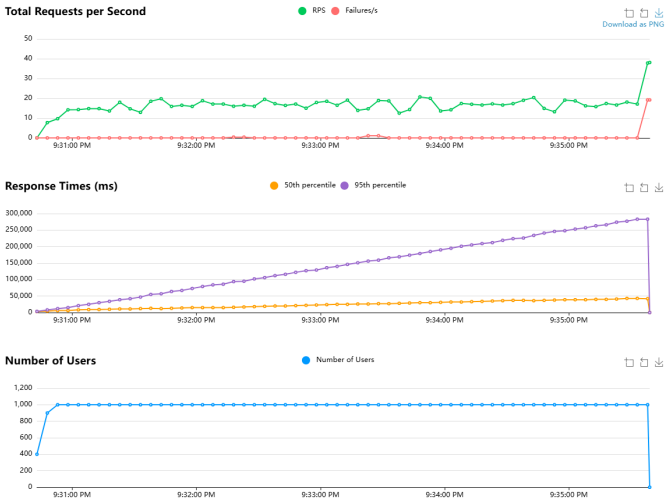

Fig. 4. Configuration of Load Test Parameters in Locust: Setting 1000 Concurrent Users with a Ramp-Up Rate of 100 User Per Second.

with failure rates rising to 1.26% as shown in Fig. 4. The results indicate that while the system scales, response times and failure rates increase under higher loads, emphasizing the need for performance optimizations.

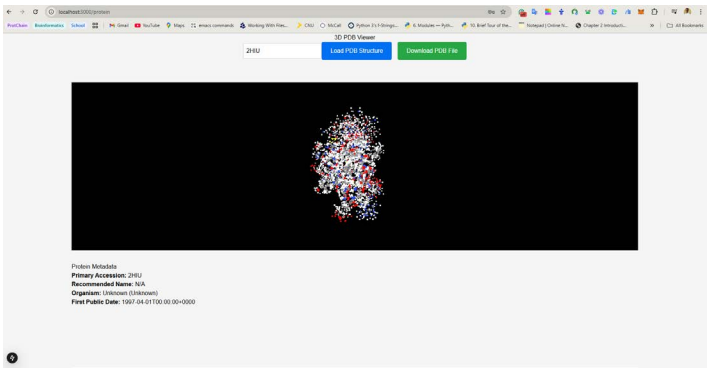

Fig. 5. Querying the 2HIU protein, with an option to download

C. Functionality Performance Using Human Insulin (2HIU)

The framework was tested using the human insulin protein (PDB code: 2HIU) to validate its secure and decentralized proteomic data management capabilities. User authentication

ensures only authorized access, maintaining data security. After logging in, users queried the 2HIU protein by entering its PDB code, retrieving its 3D molecular structure and metadata, as illustrated in Fig. 5. The system also provided a direct PDB file download option, facilitating external analysis. These seamless features highlight its efficiency in bioinformatics, integrating secure authentication, visualization, and decentralized data retrieval for enhanced proteomic research.

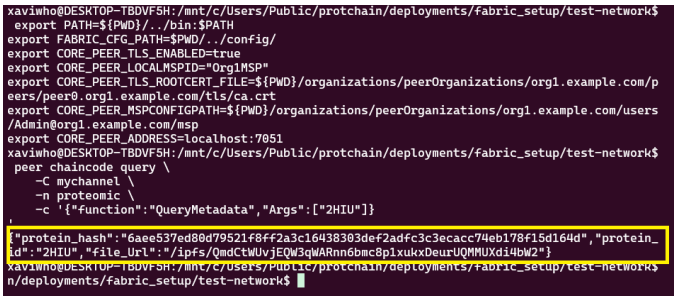

Fig. 6. Blockchain-IPFS synchronization for the 2HIU protein query. The protein hash is securely stored on-chain, with a reference link to its metadata and data stored offchain on IPFS

Finally, the proposed framework's backend operations integrate blockchain with decentralized storage via IPFS to ensure efficient and secure proteomic data management. When querying 2HIU, the platform generated a unique reference ID stored securely on IPFS, maintaining data integrity while handling large datasets. The metadata was verified for immutability and accessibility through IPFS. Additionally, the blockchain synchronized the protein hash stored on-chain with its IPFS reference (Fig. 6), ensuring consistency between both storage layers. This hybrid approach optimizes scalability, security, and accessibility, demonstrating our framework's effectiveness in decentralized bioinformatics data management.

V. CONCLUSION AND FUTURE WORK

This study proposed a blockchain-based framework designed for secure, interoperable, and scalable proteomic data management. Using Hyperledger Fabric and IPFS, it ensures data immutability, traceability, and controlled access while seamlessly integrating with bioinformatics databases. Performance evaluations demonstrate its capability to efficiently handle proteomic data, highlighting its modular and decentralized design. Future improvements will improve scalability, enabling real-time data uploads from researchers, and supporting larger datasets with reduced latency. These advancements will make our framework more suitable for bioinformatics applications, including drug development, ultimately establishing a new standard for decentralized and secure proteomic data management.

ACKNOWLEDGMENT

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korean government (MSIT) (IITP-2025-RS-2020-II201612, 25%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 25%) and by the MSIT, Korea, under the ITRC support program (IITP-2025-RS-2024-00438430, 25%), by the IITP (Institute of Information & Communications Technology Planning & Evaluation)-ICAN (ICT Challenge and Advanced Network of HRD) grant funded by the Korean government (Ministry of Science and ICT) (IITP-2025-RS-2022-00156394, 25%).

REFERENCES

[1] P. Samaras, T. Schmidt, M. Frejno, S. Gessulat, M. Reinecke, A. Jarzab, J. Zecha, J. Mergner, P. Giansanti, H.-C. Ehrlich *et al.*, "Proteomicsdb: a multi-omics and multi-organism resource for life science research," *Nucleic acids research*, vol. 48, no. D1, pp. D1153–D1163, 2020.

[2] J. Ma, T. Chen, S. Wu, C. Yang, M. Bai, K. Shu, K. Li, G. Zhang, Z. Jin, F. He *et al.*, "iprox: an integrated proteome resource," *Nucleic acids research*, vol. 47, no. D1, pp. D1211–D1217, 2019.

[3] S. D. Appasamy, J. Berrisford, R. Gaborova, S. Nair, S. Anyango, S. Grudinin, M. Deshpande, D. Armstrong, I. Pidruchna, J. I. Ellaway *et al.*, "Annotating macromolecular complexes in the protein data bank: Improving the fairness of structure data," *Scientific data*, vol. 10, no. 1, p. 853, 2023.

[4] Y. Moriya, S. Kawano, S. Okuda, Y. Watanabe, M. Matsumoto, T. Takami, D. Kobayashi, Y. Yamanouchi, N. Araki, A. C. Yoshizawa *et al.*, "The jpost environment: an integrated proteomics data repository and database," *Nucleic acids research*, vol. 47, no. D1, pp. D1218–D1224, 2019.

[5] M. Kang, E. Ko, and T. B. Mersha, "A roadmap for multi-omics data integration using deep learning," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab454, 2022.

[6] M. L. Heacock, A. R. Lopez, S. M. Amolegbe, D. J. Carlin, H. F. Henry, B. A. Trottier, M. L. Velasco, and W. A. Suk, "Enhancing data integration, interoperability, and reuse to address complex and emerging environmental health problems," *Environmental science & technology*, vol. 56, no. 12, pp. 7544–7552, 2022.

[7] S. O. Ajakwe, I. I. Saviour, V. U. Ihekoronye, O. U. Nwankwo, M. A. Dini, I. U. Uchechi, D.-S. Kim, and J. M. Lee, "Medical iot record security and blockchain: Systematic review of milieu, milestones, and momentum," *Big Data and Cognitive Computing*, vol. 8, no. 9, p. 121, 2024.

[8] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *Acm Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–41, 2021.

[9] A. R. Akhatov, M. Sabharwal, F. M. Nazarov, and A. Rashidov, "Application of cryptographic methods to blockchain technology to increase data reliability," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2022, pp. 642–647.

[10] P. Kamboj, S. Khare, and S. Pal, "User authentication using blockchain based smart contract in role-based access control," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 2961–2976, 2021.

[11] V. I. Kanu, S. O. Ajakwe, J. M. Lee, and D.-S. Kim, "Blockchain technology in protein folding: Enhancing data sharing and collaboration," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2024, pp. 1913–1918.

[12] M. Masseroli, B. Mons, E. Bongcam-Rudloff, S. Ceri, A. Kel, F. Rechenmann, F. Lisacek, and P. Romano, "Integrated bio-search: challenges and trends for the integration, search and comprehensive processing of biological information," *BMC bioinformatics*, vol. 15, pp. 1–15, 2014.

[13] M. R. Kamdar, J. D. Fernández, A. Polleres, T. Tudorache, and M. A. Musen, "Enabling web-scale data integration in biomedicine through linked open data," *NPJ digital medicine*, vol. 2, no. 1, p. 90, 2019.

[14] I. U. Ajakwe, S. O. Ajakwe, J. M. Lee, and D.-S. Kim, "Iot-blockchain frameworks in environmental pollution monitoring and data management," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2024, pp. 2147–2151.

[15] Z. Wang and Y. He, "Precision omics data integration and analysis with interoperable ontologies and their application for covid-19 research," *Briefings in Functional Genomics*, vol. 20, no. 4, pp. 235–248, 2021.

[16] S. Fillinger, L. de la Garza, A. Peltzer, O. Kohlbacher, and S. Nahnsen, "Challenges of big data integration in the life sciences," *Analytical and Bioanalytical Chemistry*, vol. 411, 08 2019.

[17] F. Mohammadipanah and H. Sajedi, "Potential of blockchain approach on development and security of microbial databases," *Biological Procedures Online*, vol. 23, pp. 1–8, 2021.

[18] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," *Peer-to-peer Networking and Applications*, vol. 14, pp. 2901–2925, 2021.

[19] ProtChain-Repo. (2024, 10) An innovative framework utilising blockchain technology to achieve seamless interoperability in protein data systems. [Online]. Available: https://github.com/prot-chain/