

## Blockchain-Enabled Cybersecurity for Additive Manufacturing: A Framework for Data Integrity and Traceability

Laraib Khan <sup>a</sup>, Muhammad Arif Mahmood <sup>b</sup>, Todd Sparks <sup>c</sup>, Frank Liou <sup>a</sup>

<sup>a</sup> Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA

<sup>b</sup> Intelligent Systems Center, Missouri University of Science and Technology, Rolla, MO 65409, USA

<sup>c</sup> Product Innovation and Engineering (PINE) LLC, St. James, MO 65559, USA

### Abstract

Additive manufacturing (AM) proposals design freedom but also offer vulnerabilities in Industry 4.0 environments where machines are linked through IoT. Cyber-attacks on AM processes can compromise part quality, intellectual property (IP), and data integrity. To address these problems, this study proposes and implements a blockchain-enabled cybersecurity framework that combines Ethereum smart contracts, InterPlanetary File System (IPFS) storage, and Neo4j graph databases. The framework certifies immutable logging of design files, operating conditions, and sensor data, while smart contracts implement role-based access, version control, and automated verification. The experimental authentication between distributed nodes demonstrated an effective prevention of file tampering, recognition of unauthorized alterations, and dependable end-to-end traceability of CAD files as well as machine logs. In comparison with existing approaches, this work provides a scalable, tamper-resistant, and auditable solution that improves trustworthiness in AM digital supply chains.

**Keywords:** Additive Manufacturing; Secure Additive Manufacturing Systems; Blockchain; Smart contracts; IP theft.

### 1. Introduction

Additive Manufacturing (AM) has resulted in novel and decentralized manufacturing by permitting intricate geometries printing using layer-by-layer technique via digital computer-aided design (CAD) models [1]. As AM grows into data enriched cyber-physical systems linked by internet of things (IoT) platforms and digital twins (DTs), they have become more susceptible to cyber-attacks that may risk part quality, intellectual property (IP), and machine safety. Various studies have been carried out to identify the jeopardies in AM workflows. In Ref. [2], a Trojan-based malware was introduced which can change AM firmware, resulting in structural defects. Ultralightweight encryption was proposed to safeguard “STL” file transfers between slicer and IoT-connected AM printers, ensuring data integrity and privacy under limited computational resources [3].

Blockchain can be utilized as a cybersecurity layer in digital manufacturing. Recently, Blockchain has been identified as a capable technology to unalterably record parameters, toolpath, and post-processing logs, permitting traceable and tamper-proof workflow in AM [4]. In Ref. [5],

blockchain was applied to secure digital thread by integrating CAD/CAM, and sensing data into an auditable chain. Tao et al. (2021) established a blockchain-DT system to support real-time monitoring along with trustworthiness in a smart factory set up [6]. Smart contracts (SCs) are a critical part of blockchain supporting asset management. Recently, Cruz et al. (2018) established role-based access control in AM by utilizing SCs [7]. Besides, Wang et al. (2019) developed adaptive SCs to control log on policies for user and system states [8].

Process provenance (PP) is also essential in AM systems. Wala et al. (2020) proposed a blockchain-based PP system for AM-ed part history including design, sensing data, and post-inspection logs [9]. In another study by Chbaik et al. (2024), IoT-based sensing devices along with blockchain were introduced to monitor equipment status and anomalies before the part failure [10].

Blockchain has also been applied to certify a process. For instance, blockchain was explored by Yousef et al. (2025) to examine defects induced during casting process [11], resulting in a more secured and certified process. In another study by Westphal et al. (2022), blockchain was linked to ASTM 52900 framework, resulting in automated part approval by smart inspecting [12]. Ermyas et al. (2019) explored blockchain layer to monitor and track data exchange, resulting an immutable data log layer that can be extended to digital factories [13].

Blockchain has been explored to resolve challenges in intricate manufacturing processes, including trust assessment, parts counterfeit restriction and anomalies identification. Blockchain was utilized in complex manufacturing system by Wang et al. (2022), and a micro-service design was proposed to improve handling efficiency [14]. Furthermore, a trust assessment model was developed using blockchain by Kurpuweit et al. (2021) for dispersed AM supply chain [15]. Besides, the AM-ed parts' counterfeiting as well as IP protection issues were resolved by Holland et al. (2017) [16]. A sign identity composite method was proposed by Geng et al. (2025) to control and manage the identity management in a distributed environment [17]. To identify the abnormalities in a control system, blockchain-based system was explored by Jadidi et al. (2020), incorporating log organization as well as anomaly identification [18]. The proposed system was able to limit the cyber-attack due to blockchain sandboxes incorporated within the framework. Blockchain is applied to control IP and manage manufacturing process in a distributed environment. Federated learning technique was utilized by Sun and Diao (2023), allowing model update in case of design without disclosing data [19]. Runtime devices' certification and access control has been proposed by Hang and Kim (2020) using blockchain in IoTs [20].

However, current efforts persist fragmented, concentrating on isolated aspects such as encryption of STL files, access control through smart contracts, or provenance tracking. None of these methods deliver a holistic solution that combines distributed storage, immutable logging, and relationship-based metadata management for AM workflows. This research addresses the gap mentioned above by developing a cohesive framework by leveraging blockchain, IPFS, and Neo4j to deliver end-to-end data safety and traceability within AM supply chain.

In this study, a framework has been proposed by integrating blockchain, InterPlanetary File System (IPFS)-based data storage system, and Neo4j graph database to manage data trustworthiness and cybersecurity in AM processes. Typically, AM is equipped with centralized storage, resulting in more susceptibility towards data interfering as well as unauthorized access.

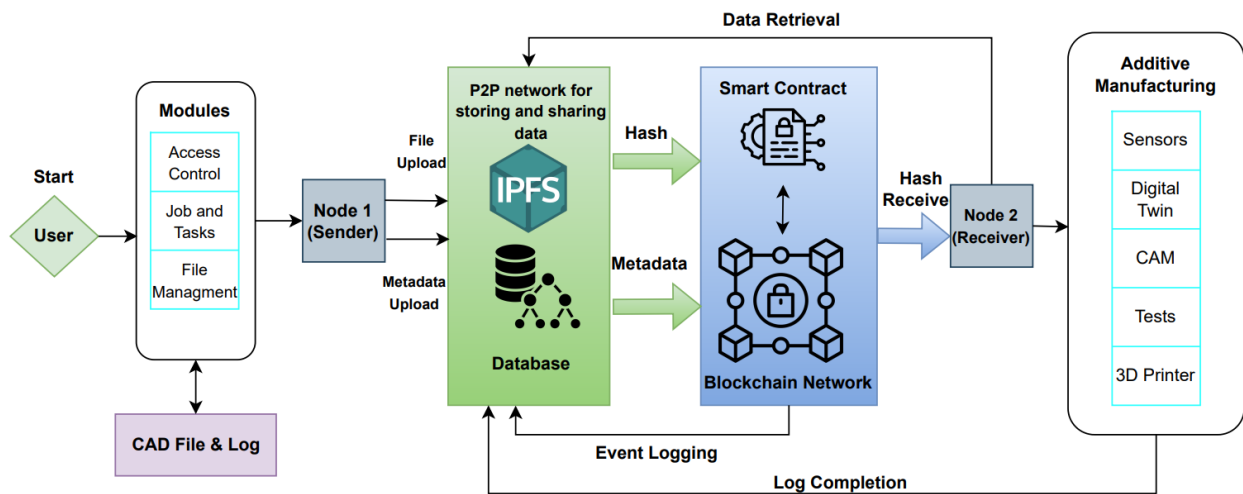
The proposed three-layered framework resolves the issue mentioned above and guarantees immutable data transactions and access events.

The main objective of this study is to recommend and validate a blockchain-enabled cybersecurity framework that leverages Ethereum-based smart contracts (SCs), IPFS-based decentralized storage, and Neo4j metadata. The framework is developed to secure AM design files and processing parameters, ensuring immutability, authenticity, and traceability in a distributed network.

## 2. Methodology

Figure 1 describes a secure framework for managing and executing AM processes by integrating blockchain, IPFS, and Neo4j along with dispersed nodes. In this procedure, the user initiates the process through an interface including modules such as access control, task and file management, which is linked to the Node 1 (sender node). Let's assume that a CAD file has to be sent from Node 1 to Node 2 (receiver) responsible for printing the AM part. The CAD file from Node 1 will be stored to IPFS, while the metadata is also uploaded to Neo4j. IPFS will create a novel hash for the file, which will be logged on, align with metadata, to blockchain using SC. All transactions are recorded on Neo4j database. Node 2 will recover the hash using the blockchain and utilize it to attain corresponding CAD file from IPFS, which will be used in AM printer to manufacture part. The whole process results in a secure and tamper-proof data transfer.

This framework delivers secure, traceable, and tamper-proof transfer of AM digital files. The framework leverages blockchain, IPFS-based file storage, and Neo4j for metadata management. The logic flow is: (a) files uploaded to IPFS are hashed and logged onto the blockchain, (b) associated to metadata in Neo4j, and (c) verified by the receiver prior to usage.



**Figure 1:** Blockchain enabled cybersecurity for AM process flow and utilization.

The proposed framework has three key elements, including an Ethereum-based blockchain [21], an IPFS-based data storage system [22], and a Neo4j graph database [23]. Ethereum was chosen over Hyperledger owing to its flexibility in SC development and scalability within

distributed networks. IPFS was selected for peer-to-peer (P2P) and content-addressable storage that avoids single points of failure and supports large file handling through chunking. Neo4j was nominated for its capability to model and request complex relationships, making it efficient for auditing file lineage and access events. These components deliver an integrated cybersecurity framework. The elements of this framework are explained below.

## **2.1. Blockchain Infrastructure**

An Ethereum-based blockchain network was established to log file transmission between two nodes, such as Node 1 is sender while Node 2 is receiver. Practically, Node 1, with permission, may upload and register CAD files. Node 2, with restricted permissions, may request to retrieve and verify files but cannot modify them. Role-based access control is required by the SC, confirming each actor accomplishes the authorized operations. A SC was formed and employed within the network that manages the file log, hash generation and verification, as well as file access control. The SC was responsible for recording the IPFS hash and corresponding metadata for every file upload prompted by the source (sender). To ensure data reliability, each file transaction was signed “cryptographically” and recorded in an immutable registry.

## **2.2. IPFS Storage**

The IPFS was utilized for P2P (peer-to-peer) file sharing protocol to deposit files digitally using a dispersed environment. The sender pushed files to IPFS by utilizing a localized IPFS node, generating a hash for individual file that was communicated to the SC on the blockchain and serve as the reference point to retrieve the file.

## **2.3. Neo4j Database**

To facilitate metadata organization and activity logging, a Neo4j database was implemented. Metadata, such as hash generation, sending and receiving nodes addresses, and time labelling, was mined and mapped as graph nodes and connections. The database schematic was proposed to combine consumers, blockchain activities, and IPFS resources, allowing effective exploration as well as conception of data source and access behaviors.

The implemented prototype used IPFS along with SHA-256 hashing to create content identifiers (CIDs). Files up to 2.0 GB were supported involving an automated chunking of larger sized files. The SC, written in Solidity, followed a simple logic: file registration triggers hash generation as well as storage, whereas the retrieval requests validate the hash against blockchain records. The unauthorized access attempts failed automatically. Metadata was uploaded in Neo4j as nodes (file, user, transaction) and edges (uploaded by, retrieved by), allowing enquiries such as file lineage trailing, recovery history, and irregularity detection.

### 3. Results

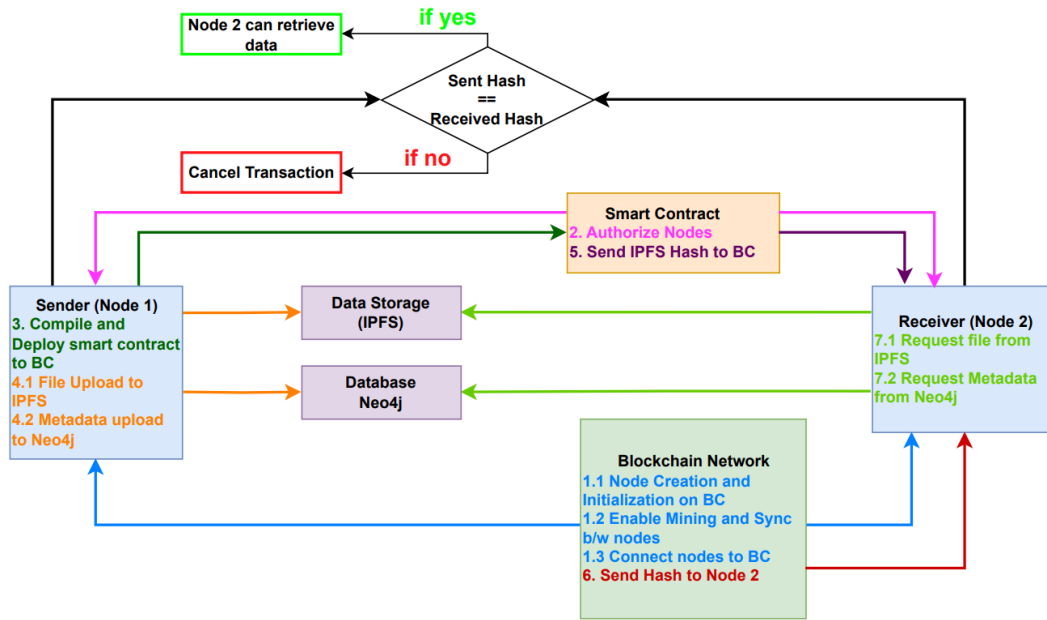
#### 3.1. Proposed Framework Workflow

Digital assets, including CAD models as well as machine-logs, must be securely transferred from AM systems for execution. Ensuring the safety, traceability, and integrity of the operations mentioned above becomes vital as AM moves into a distributed ecosystem. Principally in industries where part quality and provenance are critical, cyber-attacks including data manipulation, illegitimate access, and IP theft exhibit serious issues. To address these challenges, a framework has been proposed in Figure 1. The principal objectives of this framework include safeguarding the digital transfer of files against tampering, ensuring immutable and traceable records via blockchain, and managing metadata effectively. Figure 2 describes the process flow of the integrated system architecture.

The process initiates with the formation of a private Ethereum blockchain network, with several nodes for the AM supply chain. To certify that each action is cryptographically logged and verified, the subsequent activities are conducted: (a) each contributor starts the blockchain node and links with the P2P network; (b) mining is allowed to support each transaction authentication and block formation; and (c) nodes preserve a constant ledger, safeguarding a history of all data-related activities within the distributed system.

To facilitate metadata anchoring and secure access control, a Solidity-developed smart contract (SC) is deployed on the blockchain. The SC ensures that only pre-approved and verified nodes can access resources. Once the file is uploaded, it securely stores the IPFS hash on the blockchain, creating a tamper-proof link between the data file and its provenance.

The file-transfer process starts by the sender (Node 1) after endorsement. The SC is assembled and employed to the localized blockchain by the sender. After that, the digital file is uploaded to the IPFS, and a unique content-addressable hash that denotes the file is generated. Furthermore, a matching metadata log is made within Neo4j, which comprises information such as file type, name, timestamp, the Node 1 address, and file status. After the file is uploaded, the generated hash is sent to the recipient (Node 2) using blockchain by SC. This transaction helps in resulting a safe and unalterable reference of the uploaded file for receiver. The receiver end utilizes the provided hash to recover the file from the IPFS network. Besides, the file-related metadata is attained via Neo4j, which records every single recovery event to enable auditability and recognize efforts of unauthorized access. The receiver end authenticates the file's hash against the original hash saved on the blockchain to approve the file's authenticity. It also verifies that the file has not been interfered with during transmission and is safe to utilize. If the process is completed in a negative way, the transaction is automatically terminated, reducing the likelihood of data tampering.



**Figure 2:** Detailed file sharing and retrieving workflow.

### 3.2. Blockchain Integration Results

The proposed secured framework is developed by integrating blockchain, decentralized storage, and graphical metadata tracking systems. This section compiles the experimental results obtained using the developed prototype. The system consists of two Ethereum-based nodes representing sender and receiving stakeholders in a distributed AM environment. The results have been validated at SC deployment, file registry, logging, verification, and file retrieval. **Figures 3-6** illustrate key milestones in the system’s implementation and serve as evidence for the feasibility, tamper-resistance, and traceability features of our architecture.

The first step in secured data sharing process is to establish a SC on blockchain. Figure 3 illustrates the terminal output in real-time during the deployment via a “Truffle” Suite. The SC was formed in Solidity and compiled into bytecode and transferred to the blockchain via geth. The primary features shown in **Figure 3** include endorsement that the contract was mined and spread across the network, the computational cost to deploy the SC that is considered a prime metric when approximating scalability, and an exclusive Ethereum address to which the contract is linked. The installed SC serves as the “cryptographic” backbone to manage access, authentication, and transaction log. It stipulates the conditions in which file registration and recovery are permitted, yielding a resilient system against illegitimate operations, guaranteeing accountability for transactions. This step creates the framework for an absolute, dispersed authentication mechanism that will regulate the files as well as the metadata activities.

```
laraib@pop-os:~/Desktop/Blockchain_global_installation_implementation$ truffle
migrate --config truffle-config.cjs --network node1 --reset

Compiling your contracts...
=====

Starting migrations...
=====
> Network name: 'node1'
> Network id: 1234

1_deploy_contracts.js
=====

Deploying 'FileTransfer'
-----
> transaction hash: 0x7bfa036304c0c501fe12fd2abc486e29b5763d67ffe2591bc1711d472d705b7d
> Blocks: 0 Seconds: 0
> contract address: 0x96ea47B188ad8F529181fb83270591D318914649
> block number: 12301
> account: 0xbCB3600fcD00516D0E95b0F965173BFd061Fc57c
> balance: 999999.391605659655199152
> gas price: 10 gwei
> value sent: 0 ETH
> total cost: 0.01035539 ETH

> Saving artifacts
-----
> Total cost: 0.01038539 ETH

Summary
=====
> Total deployments: 1
> Final cost: 0.01038539 ETH
```

**Figure 3:** Deployment of smart contract on the Ethereum blockchain.

Figure 4 displays the system interacting with IPFS and Neo4j components, which handle metadata recording as well as file storage, respectively. A localized IPFS daemon is applied to ascribe the file to the distributed file network. The system outcome is a content identifier (CID), a SHA-256 hash based on the file contents. The CID is a fingerprint for a given file, any modifications to the file results in a new CID. Using IPFS compared to consolidated storage confirms the accessibility and resilience to single-point failures.

Once the CID is generated, the system stores the metadata within Neo4j, comprising the file name, file type, sender’s address, CID, timestamp, and transaction hash from SC. It is important to mention here that the Neo4j was selected owing to its ability for relationship-based enquiries, enabling quick recoveries of data, usage behavior and file associations. For AM supply chain, this permits the graphical and effective tracking of part alterations, approvals, and certification stages. The dual-layered technique outcomes in secure, verifiable, and extensible solutions for digital asset sharing in AM.

```

laraib@pop-os:~/Desktop/Blockchain_global_installation_implementation$ node client/sender.cjs
Do you want to upload a file or a folder? (file/folder): file
Enter the path to the file:
/home/laraib/Desktop/Blockchain_global_installation_implementation/node1/output_files/received_data.txt
Starting upload and registration process...
Contract Address: 0x96ea47B188ad8F529181fb83270591D318914649
Sending Transaction Hash: 0x7bfa036304c0c501fe12fd2abc486e29b5763d67ffe2591bc1711d472d705b7d
Uploading file to IPFS...
File uploaded to IPFS: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG
IPFS Hash: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG
Checking if the file already exists...
Adding metadata to Neo4j...
Logging transfer from 0xbCB3600fcD00516D0E95b0F9651738Fd061Fc57c (Node 1) to 0x627b825FCbeA6a11a0fd5e0e6e98b54fd49A6c9C (Node 2)...
Relationship successfully added to Neo4j.
Metadata successfully added to Neo4j.
Registering file in the smart contract...
File registered with the smart contract. IPFS Hash: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG

```

**Figure 4:** File upload to IPFS and metadata to Neo4j.

To test the system’s confrontation to tampering, a controlled alteration of a file was conducted after CID generation and tried to re-register it by utilizing identical transaction flow. The results are depicted in Figure 5. It can be observed that the blockchain transaction stops due to a discrepancy between the newly generated CID and prior CID within SC. The SC guarantees that the hash given upon recovery matches the one registered during the first upload. The attempted manipulation stops, representing the efficiency of integrity protection. This example can be compared with the real-world cyber-attack in AM. A compromised STL or G-code file can result in faulty parts. Our blockchain-based CID validation confirms that only unaltered and approved files are recognized for AM printing.

```

laraib@pop-os:~/Desktop/Blockchain_global_installation_implementation$ node client/receiver.cjs
Enter the IPFS hash of the file you want to retrieve: QmVjKT4ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG

Receiving IPFS Hash != Sending IPFS Hash

Contract Address: 0x96ea47B188ad8F529181fb83270591D318914649
Sending Transaction Hash: 0x7bfa036304c0c501fe12fd2abc486e29b5763d67ffe2591bc1711d472d705b7d
Receiving Transaction Hash: 0x7bfa036304c0c501fe12fd2abc486e29b5763d67ffe2591bc1711d472d705b7d

Receiving Transaction Hash == Sending Transaction Hash

Checking Metadata with received Hash to ensure Different file or Modified file
Fetching metadata from Node 1...

File metadata retrieved from Neo4j: {
  owner: 'Laraib Khan',
  filename: 'received_data.txt',
  size: 406208,
  ipfsHash: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG,
  timestamp: '06/13/2025, 11:48:25 PM',
  status: 'Sent (Node 1) , Receive (Node 2) '
}
** Metadata Matched (File has been modified while Transaction) **

RECEIVING PROCESS ABORTED . . .

```

**Figure 5:** Failed transaction due to file modification during transaction.

Within the proposed framework, the final step is to recover the stored data. Figure 6 exhibits an effective fetch process, which recovers the file from IPFS utilizing the CID. It can be seen that the metadata is retrieved from Neo4j using graph-based queries based on file name, sender address, or CID, and data validity is carried out by comparing the IPFS CID with the SC record. In real-world manufacturing, it can be correlated with a supplier recovering design files for manufacturing or a certification agency retrieving part histories. The capability to automate authorization (by SC

along with CID match), tracking (using Neo4j), and recovery (via IPFS) validates the proposed architecture's efficiency in protecting digital supply chains.

```
lارايب@pop-os:~/Desktop/Blockchain_global_installation_implementation$ node client/receiver.cjs
Enter the IPFS hash of the file you want to retrieve: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG

Receiving IPFS Hash == Sending IPFS Hash

Fetching metadata from Node 1...
Contract Address: 0x96ea47B188ad8F529181fb83270591D318914649
Sending Transaction Hash: 0x7bfa036304c0c501fe12fd2abc486e29b5763d67ffe2591bc1711d472d705b7d
Receiving Transaction Hash: 0x7bfa036304c0c501fe12fd2abc486e29b5763d67ffe2591bc1711d472d705b7d

Receiving Transaction Hash == Sending Transaction Hash

File metadata retrieved from Neo4j: {
  owner: 'Lارايب Khan',
  filename: 'received_data.txt',
  size: 406208,
  ipfsHash: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG,
  timestamp: '06/13/2025, 11:48:25 PM',
  status: 'Sent (Node 1) , Receive (Node 2) '
}
Downloading file from IPFS...
Retrieving file from IPFS: QmVjKT5ewKbBhqcDr5hAMGeRRH66Xsnix1Nr4zjGVZATjG
File downloaded successfully: received_data.txt
Storing metadata on Node 2...
Logging transfer from 0xbCB3600fcD00516D0E95b0F965173BFd061Fc57c (Node 1) to 0x627b825FCbeA6a11a0fd5e0e6e98b54fd49A6c9C (Node 2)...
Relationship successfully added to Neo4j.
Relationship created in Neo4j!
Metadata successfully stored in Neo4j on Node 2.
```

Figure 6: Retrieval of data from IPFS and Neo4j.

### 3.3. Framework Evaluation and Future Directions

While the prototype demonstrates the probability of integrating blockchain, IPFS, and Neo4j to secure AM workflows, the system performance evaluation remains essential. To provide a clear perception, this section summarizes the current observations, detects missing measurements, and highlights future outlook to improve the framework.

#### 3.3.1. Performance Metrics

The current prototype validates the functional behavior but does not provide the detailed numerical benchmarks. Future work will estimate transaction times, system throughput, and Ethereum gas costs. The initial observations propose that SC deployment incurs gas fees (a few USD), while CID creation and retrieval happen within seconds.

#### 3.3.2. Comparison with Other Approaches

Compared to centralized storage, the anticipated framework removes single points of failure and delivers tamper detection through content-addressable hashing. In this study, Ethereum was chosen for its flexible SC development and scalability testing. Traditional data integrity approaches secure confidentiality but lack attribution and auditability, which is presented in current frameworks.

#### 3.3.3. Stress Testing

Robust testing under various nodes, concurrent transactions, various file sizes, and simulated network interruptions have not yet been performed. The future work will assess scalability, concurrence handling, and fault acceptance to authenticate performance in realistic AM environments.

### **3.3.4 Resource Utilization**

The present study did not record CPU load, memory utilization, and disk activity during Ethereum SC execution, IPFS file handling, and Neo4j metadata queries. Future evaluations will include these metrics to calculate computational overhead and optimize deployment.

## **4. Conclusions**

In this study, a distributed blockchain-enabled system has been proposed to address cybersecurity concerns in AM. The proposed system includes Ethereum-based SCs to manage access, IPFS for distributed file storage, and Neo4j for metadata administration, providing a tamper-proof, verifiable, and scalable solution designed for digital manufacturing. The implementation and testing of the proposed framework, as shown using end-to-end file transfer between two nodes, proves the effectiveness of the proposed approach. The results have been validated via safe file recording, absolute data logging, and automated file trustworthiness verification using SC logic. Besides, Neo4j assists in dynamic querying and connection analysis within intricate AM workflows. The proposed approach not only advances the cybersecurity of AM systems, but it also promotes transparency as well as decentralization, which are critical for Industry 4.0. This study bridges the gap between blockchain and manufacturing process control, creating a solid platform to deploy digital twins, decentralized manufacturing networks, and traceable supply chains.

## **Acknowledgments**

The authors would like to acknowledge the Product Innovation and Engineering (PINE), LLC and NAVAIR for their support in carrying out the research.

## **References**

1. I. Gibson, D. Rosen, and B. Stucker, *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing, second edition*, 2nd ed. Springer New York, 2015. doi: 10.1007/978-1-4939-2113-3/COVER.
2. H. Pearce, K. Yanamandra, N. Gupta, and R. Karri, "FLAW3D: A Trojan-Based Cyber Attack on the Physical Outcomes of Additive Manufacturing," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5361–5370, Dec. 2022, doi: 10.1109/TMECH.2022.3179713.

3. N. Yasmin and R. Gupta, "Ultra-Lightweight Encryption for STL Files in IoT-based 3D Printing," *International Journal of Safety & Security Engineering*, vol. 13, no. 4, pp. 657–664, 2023.
4. R. Abe, S. Suzuki, K. Saito, H. Tanaka, O. Nakamura, and J. Murai, "Fabchain: Managing Audit-able 3D Print Job over Blockchain," *IEEE International Conference on Blockchain and Cryptocurrency, ICBC*, 2022, doi: 10.1109/ICBC54727.2022.9805519.
5. A. Banerjee, "Blockchain Technology: Supply Chain Insights from ERP," *Advances in Computers*, vol. 111, pp. 69–98, Jan. 2018, doi: 10.1016/bs.adcom.2018.03.007.
6. F. Tao *et al.*, "Digital twin and blockchain enhanced smart manufacturing service collaboration and management," *J Manuf Syst*, vol. 62, pp. 903–914, Jan. 2022, doi: 10.1016/J.JMSY.2020.11.008.
7. J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, Mar. 2018, doi: 10.1109/ACCESS.2018.2812844.
8. S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," *IEEE Trans Syst Man Cybern Syst*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019, doi: 10.1109/TSMC.2019.2895123.
9. W. Alkhader, N. Alkaabi, K. Salah, R. Jayaraman, J. Arshad, and M. Omar, "Blockchain-based traceability and management for additive manufacturing," *IEEE Access*, vol. 8, pp. 188363–188377, 2020, doi: 10.1109/ACCESS.2020.3031536.
10. N. Chbaik, A. Khiat, A. Bahnasse, and H. Ouajji, "Blockchain-Assisted IoT Wireless Framework for Equipment Monitoring in Smart Supply Chain: A Focus on Temperature and Humidity Sensing," *IEEE Access*, vol. 12, pp. 117504–117522, 2024, doi: 10.1109/ACCESS.2024.3449253.
11. N. Yousef, A. Sata, M. Shukla, S. Jarboui, and D. Mobarsa, "Blockchain-integrated IoT device for advanced inspection of casting defects," *Sci Rep*, vol. 15, no. 1, pp. 1–15, Dec. 2025, doi: 10.1038/S41598-025-86777-3.
12. E. Westphal, B. Leiding, and H. Seitz, "Blockchain-based quality management for a digital additive manufacturing part record," *J Ind Inf Integr*, vol. 35, p. 100517, Oct. 2023, doi: 10.1016/J.JII.2023.100517.
13. E. Abebe *et al.*, "Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer (industry track)," *Middleware Industry 2019 - Proceedings of the 2019 20th International Middleware Conference Industrial Track, Part of Middleware 2019*, pp. 29–35, Dec. 2019, doi: 10.1145/3366626.3368129.
14. Y. Wang, S. Li, H. Liu, H. Zhang, and B. Pan, "A Reference Architecture for Blockchain-based Traceability Systems Using Domain-Driven Design and Microservices," *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, vol. 2022-December, pp. 269–278, 2022, doi: 10.1109/APSEC57359.2022.00039.

15. S. Kurpjuweit, C. G. Schmidt, M. Klöckner, and S. M. Wagner, “Blockchain in Additive Manufacturing and its Impact on Supply Chains,” *Journal of Business Logistics*, vol. 42, no. 1, pp. 46–70, Mar. 2021, doi: 10.1111/JBL.12231.
16. M. Holland, C. Nigischer, and J. Stjepandic, “Copyright protection in additive manufacturing with blockchain approach,” in *Advances in Transdisciplinary Engineering*, IOS Press BV, 2017, pp. 914–921. doi: 10.3233/978-1-61499-779-5-914.
17. C. Geng, Y. Zhang, X. Xu, Y. Yao, C. Lu, and Z. Zhi, “Blockchain-based identity authentication and data interaction scheme for Industrial Internet of Things,” *Computers and Electrical Engineering*, vol. 123, p. 110143, Apr. 2025, doi: 10.1016/J.COMPELECENG.2025.110143.
18. Z. Jadidi, A. Dorri, R. Jurdak, and C. Fidge, “Securing manufacturing using blockchain,” *Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020*, pp. 1920–1925, Dec. 2020, doi: 10.1109/TRUSTCOM50675.2020.00262.
19. X. Song, Z. Liu, F. Wei, F. Sun, and Z. Diao, “Federated Learning and Blockchain-Enabled Intelligent Manufacturing for Sustainable Energy Production in Industry 4.0,” *Processes*, vol. 11, no. 5, p. 1482, May 2023, doi: 10.3390/PR11051482.
20. L. Hang and D. H. Kim, “Reliable Task Management Based on a Smart Contract for Runtime Verification of Sensing and Actuating Tasks in IoT Environments,” *Sensors*, vol. 20, no. 4, p. 1207, Feb. 2020, doi: 10.3390/S20041207.
21. “Blockchain | Ethereum.” Accessed: Jun. 01, 2025. [Online]. Available: <https://www.blockchain.com/learning-portal/tokens/ethereum-explained>
22. “IPFS Documentation | IPFS Docs.” Accessed: Jun. 01, 2025. [Online]. Available: <https://docs.ipfs.tech/>
23. “Neo4j Graph Database & Analytics | Graph Database Management System.” Accessed: Jun. 01, 2025. [Online]. Available: <https://neo4j.com/>