IPFS-Enabled Blockchain Framework for Fair Payment Systems

Ilyas Khudhair[1,2], Hasimi Sallehudin[1], Azana Hafizah Mohd Aman[1] and Muntadher Saadoon[2]
[1]*Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia*
[2]*College of Computer Science and Information Technology, Wasit University, 52001 Kut, Wasit, Iraq*
p151734@siswa.ukm.edu.my, ilyas@uowasit.edu.iq, hasimi@ukm.edu.my, muntadher.saadoon@gmail.com

Keywords: Fair Payment Systems; Smart Contracts; IPFS (Interplanetary File System); Off-Chain Storage; Decentralized Storage.

Abstract: Blockchain-based fair payment systems have gained considerable attention due to their potential to facilitate transparent, secure, and automated transactions. However, managing large-scale digital assets directly on-chain is impractical due to blockchain scalability constraints, high transaction fees, and limited storage capacities. Existing solutions often employ centralized cloud-based storage services for off-chain data, introducing trust dependencies, censorship risks, and undermining core decentralization objectives. This paper presents a preliminary exploration into integrating the InterPlanetary File System (IPFS) a decentralized, peer-to-peer storage protocol with blockchain smart contracts to enable fully decentralized off-chain storage for fair payment systems. We propose a conceptual model leveraging IPFS to ensure verifiable data delivery, automated dispute resolution, and trustless payment settlements. Preliminary qualitative analysis and comparative evaluations highlight substantial improvements in decentralization, transparency, and resilience compared to traditional cloud-based approaches. Our results demonstrate the feasibility and potential advantages of employing IPFS in blockchain payment frameworks, paving the way for truly decentralized digital asset management and exchange systems.

1 INTRODUCTION

Blockchain technology has profoundly reshaped digital transactions, enabling trustless, transparent, and secure payment processes without relying on central authorities or intermediaries [1], [2]. Decentralized payment systems, leveraging blockchain smart contracts, have gained significant attention for their potential to automate financial exchanges, eliminate fraud, and enhance transparency in various digital economies. However, one persistent challenge in realizing truly decentralized blockchain-based applications (dApps) is efficiently managing the storage and distribution of substantial digital assets, as direct on-chain storage remains impractical due to inherent scalability constraints, limited block sizes, and high transaction costs [3], [4].

To work around on-chain limits, many "decentralized" payment systems quietly park their off-chain data in a regular cloud bucket think S3 or GCS while the chain stores only a pointer or hash [5]. It's hard to argue with the convenience: elastic storage, global CDNs, familiar APIs, and a bill that (usually) makes finance happy. But the trade-offs are familiar, too. A single account suspension, a region hiccup, or an ACL misconfiguration can take data offline. Takedown requests and policy changes can nudge toward de-facto censorship. And unless versioning and audit trails are enforced and actually monitored subtle tampering may slip by [6], [7]. All of that sits uneasily with what blockchains were supposed to buy us. These central points of control and failure appear to undercut resilience, fairness, and the neutrality we expect from a decentralized payment system. The approach works, yes – but it bends the architecture back toward the very trust assumptions it set out to avoid.

Decentralized off-chain storage most visibly IPFS has been gaining traction as an alternative to parking data in a single cloud account. IPFS runs over a peer-to-peer network and uses content addressing: every file is referenced by a CID (essentially the hash of its bytes), so if the content changes, the CID changes too. That setup makes tampering easy to spot and, in many cases, hard to get away with; it also appears to blunt

platform-level takedowns since copies can live on independent nodes rather than one provider [8], [9]. It isn't magic, though. Availability depends on pinning and who's actually hosting the data – teams often pin to a couple of providers and keep a local node as a backstop. Even with those caveats, recent work has used IPFS for decentralized sharing, archival records, and computation pipelines, reporting advantages over traditional cloud models in integrity checks and resilience to outages [10], [11]. However, few studies systematically integrate IPFS with blockchain-based payment protocols, particularly regarding automated payment settlement and dispute resolution, thus leaving critical gaps unresolved.

Motivated by these considerations, this paper presents a novel approach that integrates IPFS as decentralized off-chain storage with blockchain-based smart contracts to establish an end-to-end fair payment system. The proposed framework aims to ensure trustless verification, transparent settlement, and automated dispute management while eliminating reliance on third-party storage providers. Specifically, this research provides the following key contributions:

- A detailed conceptual model demonstrating how IPFS can be integrated seamlessly with blockchain-based payment mechanisms.
- Qualitative analysis highlighting the advantages, feasibility, and potential limitations of the proposed decentralized storage approach compared to traditional cloud-based solutions.
- A preliminary scenario illustrating dispute resolution processes within the IPFS-based fair payment system, supported by conceptual diagrams and comparative tables.

The remainder of the paper is structured as follows: Section II defines the problem context and motivation, Section III reviews related literature, Section IV describes the proposed architecture and methodology, Section V presents preliminary findings and analysis, and Section VI concludes with future directions.

2 PROBLEM DEFINITION

Blockchain-based payment systems have attracted considerable attention due to their potential to enable transparent, secure, and automated financial transactions without reliance on central intermediaries [1], [2]. Nevertheless, decentralized applications (dApps), particularly those designed for fair payment and digital asset management, frequently require storing and managing large-scale or complex data. Directly storing such data on-chain is impractical due to blockchain limitations such as restricted block sizes, limited scalability, and prohibitive transaction costs [3], [12]. As a result, existing decentralized payment systems typically utilize centralized cloud services for off-chain storage to alleviate these constraints and reduce blockchain-related expenses as depicted in Figure 1 [5] [13].

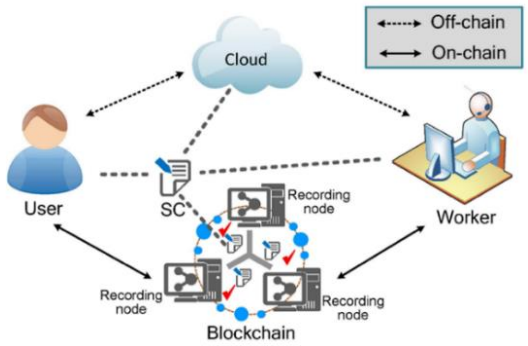

Figure 1: Decentralized payment systems relying on centralized cloud services for off-chain storage.

Cloud services scale nicely and are easy to use, but leaning on a single provider sits awkwardly with the goals of decentralization, transparency, and minimizing trust [14]. A suspended account, a misconfigured bucket policy, or a regional outage can knock data offline; takedown requests and shifting terms can amount to soft censorship [15]. You also inherit new trust anchors provider admins, access keys, internal ACLs the very stuff blockchains try to avoid. And while clouds promise high durability, they don't, by default, give you content-addressed proofs or independently reproducible audit trails; long-term integrity, availability, and verifiability become policy claims rather than cryptographic facts. That gap complicates disputes and appears to leave fair-payment flows exposed to manipulation missing files, swapped blobs, or back-dated uploads are hard to rebut without external evidence [16].

There's still a stubborn gap: an off-chain storage layer that preserves the same decentralization, censorship resistance, and minimal trust that blockchains promise on-chain. To chip away at that gap, we explore wiring IPFS a decentralized, peer-to-peer, content-addressed protocol directly into smart contracts [17], [18]. The idea is simple enough. Let the chain commit to CIDs while the actual bytes live off-chain. Verification then boils down to fetching the content for that CID and checking a signature, which

appears to give a practical, scalable, and easily verifiable path for settling payments without leaning on a single provider. Two caveats are worth flagging. Availability hinges on pinning and replication policies, and access control usually comes from encryption and key management rather than IPFS itself. Even with those reservations, the approach keeps the trust budget small and the system closer to the decentralization we set out to maintain.

3 RELATED WORK

Blockchain-based fair-payment schemes have become a go-to way to keep exchanges transparent and automated without leaning on a trusted middleman. The proposals cover a lot of ground: payment atomicity via conditional transfers or HTLC-style locks, stronger security with verifiable proofs, and fee cuts through batching or off-chain channels. A representative example is Zhang et al. [16]. Their protocol made the payment flow harder to break and, on paper, more dependable. Still, the design parked outputs in a centralized cloud bucket (think S3 or GCS) while the chain stored only references. That choice is understandable it's cheap and widely supported but it also nudges the system back toward a single provider and, by extension, appears to dilute the decentralization the blockchain layer is supposed to guarantee.

To sidestep the chain's storage ceiling, a lot of recent work parks off-chain data in regular cloud services – think S3 buckets or GCS blobs while the blockchain keeps only a pointer or hash [16], [19]. It's convenient and cheap, sure, but it appears to smuggle trust back in: an account suspension, a regional outage, or a sloppy bucket policy can take data offline; takedown requests can edge into soft censorship; and "integrity" often boils down to provider logs rather than cryptographic facts [18]. These cracks suggest we still need to look harder at decentralized, durable off-chain storage that fits blockchain assumptions systems where availability comes from replication, integrity comes from content addressing, and long-term verifiability doesn't hinge on a single company's dashboard.

Recent works have explored decentralized storage technologies such as the IPFS and Swarm to mitigate these challenges. Kumar et al. [4] presented an in-depth analysis of IPFS and Swarm, highlighting their compatibility with blockchain systems, especially emphasizing immutability and secure storage. Similarly, Benisi et al. [17] conducted a comparative assessment of IPFS and Swarm, emphasizing blockchain interoperability and the ability to guarantee data integrity in decentralized contexts.

Researchers have started leaning on decentralized storage mainly IPFS and, in some settings, Swarm to get around the obvious limits of parking data with a single provider. Kumar et al. [4] take a close look at both systems and argue they line up well with blockchain assumptions: content addressing helps you spot tampering, and data can live on multiple nodes rather than one vendor. Benisi et al. [17] offer a side-by-side comparison and come to a similar place, stressing interoperability with on-chain logic and the ability to verify that what you fetched is exactly what was written. That said, neither system is plug-and-play perfection. Availability still depends on who pins or incentivizes storage, and real-world latency can wobble with network churn. Even so, these studies appear to suggest that IPFS and Swarm are a better fit for decentralized workflows than a lone cloud bucket, especially when integrity and auditability matter.

Casino et al. [7] take a hard look at what "immutability" really buys you in decentralized storage and where it can bite. Freezing content by hash doesn't stop abuse; it can, in some cases, make takedowns messy. Once a harmful CID circulates, anyone can repin it, gateways may block inconsistently, and legal accountability gets murky. The authors highlight issues like malicious content dissemination, "poisoned" hashes used for harassment or denial-of-service, and the awkward linkability of CIDs to sensitive material. They also sketch practical mitigations: encrypt data client-side and manage keys off-chain; apply clear pinning policies; maintain gateway deny-lists with transparent audit logs; and design incentives that discourage hosting flagged content on IPFS-style networks. The bigger lesson appears to be that pairing decentralized storage with blockchains doesn't make safety a given you still need to engineer for abuse cases, compliance, and the tension between permanence and the need to remove harmful material.

Moreover, not all next-generation peer-to-peer data networks have embraced blockchain integration, due to concerns such as scalability and latency. Naik and Keshavamurthy [20] provided a general review of modern peer-to-peer (P2P) networks, analyzing the evolution of classical systems (e.g., BitTorrent, Chord) and newer models, with particular attention to their performance characteristics under node churn conditions. Notably, their classification categorizes IPFS as a classic P2P network. However, other recent perspectives suggest IPFS and similar technologies have significantly evolved, incorporating incentive

mechanisms and decentralized data management techniques that extend beyond traditional P2P models [9].

In terms of incentive mechanisms specifically designed for decentralized storage and delivery, Tit-for-Token (2023) presented peer incentivization protocols for IPFS-based systems, introducing economic models to foster resource sharing and fair compensation among peers [11]. Similarly, FileInsurer (2022) proposed mechanisms layered atop IPFS to enhance reliability through economic guarantees and deposit-based compensation models for data availability [21].

Despite the breadth of existing work, comprehensive solutions integrating decentralized off-chain storage with automated, blockchain-based fair payment remain limited. Few studies explicitly tackle challenges around dispute resolution, verification mechanisms, and persistent data availability when employing decentralized storage like IPFS. Thus, a gap remains in systematically combining these critical elements to form robust, decentralized payment systems.

We tackle the gap with a single, end-to-end path: wire IPFS directly into smart contracts so the chain commits to CIDs while the bytes live off-chain. That way, anyone can check what was stored (and when), payments settle under rules you can actually audit, and disputes run as code timeouts, bonds, compact proofs instead of going through a help desk. The aim is straightforward: keep trust out of the critical path, scale with rising job counts, and stay resilient when parts of the network wobble. It won't solve everything pinning policies and key management still matter but this unified approach appears to move fair-payment systems closer to the decentralization they promise, with verifiable storage, transparent settlement, and automated dispute handling that's practical in the real world.

4 IPFS AS OFF-CHAIN STORAGE

Much of the recent literature OBFP [13] and EFP [5] are typical keeps the bytes in a cloud bucket (think S3 or GCS) and writes only a hash or pointer to the chain to dodge storage fees. Pragmatic? Absolutely. It scales, ops teams know how to run it, and the bill is predictable. But the fit with blockchain's goals is shaky. Handing availability to a single provider quietly brings back a trusted third party: account suspensions, region outages, or an ACL slip can make data vanish, and takedown requests can act like soft censorship. Integrity, too, becomes a matter of provider logs and policies rather than end-to-end cryptographic proofs. That appears to undercut the very properties these payment systems claim to prioritize trustlessness, transparency, and resilience. None of this means clouds are "wrong" in every case; a hybrid setup (e.g., dual-cloud storage with periodic on-chain anchoring) may ease operations. Still, for fair-payment designs that aim to be genuinely decentralized, reliance on centralized infrastructure looks like a step back, not forward.

To tackle this limitation, we use IPFS as the off-chain storage layer, following the direction of Rani et al. [22], and Ma et al. [23]. The goal is straightforward: keep storage trustless and hard to censor while still scaling to real job sizes. With content addressing, the chain commits to a CID and the bytes live off-chain; if the file changes by even a single bit, the CID flips, which makes tampering obvious. Practically, this appears to cut our reliance on any single provider and helps with availability when multiple parties pin the same data (e.g., worker, client, and a small pinning service). It also shifts weight off the blockchain CIDs and small receipts on-chain, large artifacts on the P2P network which is likely to lower fees and keep state growth in check. Cloud buckets (S3/GCS) can be faster in some regions and are undeniably convenient, but they reintroduce a trust anchor we're trying to avoid. There are a few knobs to get right. IPFS doesn't do access control by itself, so sensitive outputs are typically encrypted client-side and keys managed off-chain. Pinning and replication policies matter, too; without them, availability can drift. Even with those caveats, replacing centralized off-chain storage with IPFS strengthens resilience and censorship resistance, and by distributing data across peers improves scalability and cost efficiency without bloating the chain.

The contribution of integrating IPFS into decentralized payment systems is a principled model for off-chain data management that advances trustlessness, transparency, and system autonomy. Conventional models often handle only the payment process on-chain while depending on centralized or semi-centralized repositories for off-chain data, introducing single points of failure, potential censorship, and weaker integrity guarantees. In the proposed design, IPFS serves as the off-chain storage layer that complements smart contracts managing fair payments. When a provider generates digital content or service outputs, the data are uploaded to IPFS, which fragments and distributes them across the peer-to-peer network. IPFS assigns a unique content hash (CID) that acts as an immutable, verifiable identifier; the provider then submits this CID to an on-chain

smart contract. The client can independently retrieve and verify the content from IPFS, ensuring it has not been altered. Figure 2 illustrates data availability in IPFS; reliable access depends on file pinning by one or more network nodes. Overall, the proposed approach not only mitigates the weaknesses of traditional off-chain storage models but also enhances the reliability, security, and efficiency of decentralized payment systems for outsourced service providers.

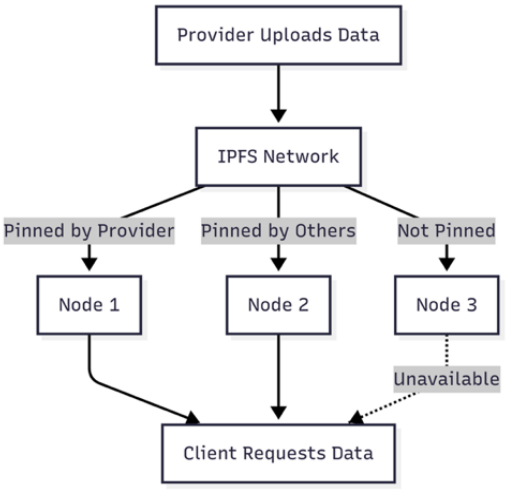

Figure 2: Illustration of data availability in IPFS that upload by service provider.

5 RESULTS AND DISCUSSION

This study presents a preliminary exploration of integrating IPFS as an off-chain storage solution within decentralized fair payment systems. Given the conceptual and prototype nature of the work, the results focus on the qualitative assessment of system behavior, feasibility, and anticipated advantages over traditional cloud-based approaches.

5.1 Feasibility and System Behavior

Through a prototype implementation and scenario analysis, the proposed model demonstrates several critical improvements. By leveraging the content-addressed architecture of IPFS, the system ensures data integrity and tamper-evidence without the need for trusted third parties. In the illustrative scenario of digital asset exchange, the provider uploads content to IPFS and submits the resulting hash to a blockchain smart contract. The client retrieves the file directly from IPFS using the hash, verifies its authenticity, and only then triggers payment release through the contract. This workflow depicted in Figure 3 confirms the model's capacity to automate fair settlement and transparent data verification.

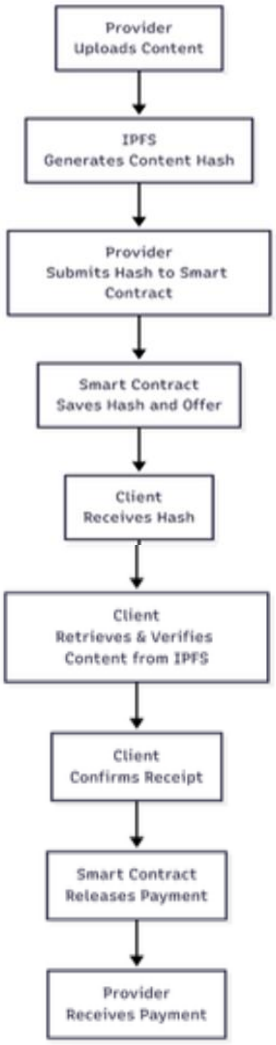

Figure 3: Automated fair settlement and transparent data verification workflow using IPFS and blockchain smart contracts.

5.2 Comparative Analysis

A qualitative comparison with conventional cloud-based off-chain storage models is presented in Table 1. The analysis highlights that, unlike centralized solutions, the IPFS-based model eliminates single points of failure and enhances censorship resistance. Furthermore, payment settlement and content verification are entirely automated and transparent via the smart contract,

significantly reducing operational risks. However, challenges remain regarding persistent data availability on IPFS, which depends on effective pinning and incentivization strategies. Retrieval latency was also identified as a variable factor, suggesting the need for further optimization in production-scale deployments.

Table 1: A qualitative comparison with conventional cloud-based off-chain storage models is presented and IPFS-Based Model.

Feature/Aspect	IPFS-Based Model (Proposed)	Cloud-Based Off-Chain Storage
Decentralization	Fully decentralized, peer-to-peer	Centralized (single/multi-cloud)
Data Integrity	Guaranteed by content hash (immutable)	Depends on provider
Payment Settlement	Automated via smart contract, trustless	Relies on third party logic
Censorship Resistance	High (difficult to remove data)	Low (provider can censor/delete)
Scalability	High, distributed across many nodes	Limited by provider infrastructure
Storage Cost	Minimal on-chain cost, pay for pinning	Pay-per-use or subscription
Data Availability	Requires persistent pinning/incentives	Guaranteed by provider SLAs
Transparency/Auditability	Fully transparent on blockchain	Limited to provider's reporting

5.3 Conceptual Findings

The conceptual results indicate that the integration of IPFS addresses key limitations of previous architectures. Specifically, the system:

- Reduces reliance on centralized infrastructure and trusted intermediaries.
- Guarantees data authenticity through content-addressed storage.
- Supports automated, trustless payment and dispute resolution.
- Provides robust transparency and auditability for all transactions.

5.4 Dispute Scenarios

The proposed IPFS-based fair payment model anticipates several principal dispute scenarios and incorporates dedicated resolution mechanisms to address each case, as detailed in Table 1. For instance, if data is not accessible on IPFS, the smart contract is designed to withhold payment and automatically trigger a refund or invoke timeout logic to protect the client. In cases where the content hash does not match the expected data, payment is withheld, and the client may explicitly reject the transaction or initiate a dispute process [18]. The system also addresses scenarios where a user falsely claims non-delivery by requiring explicit confirmation of receipt, with provisions for auditor intervention when necessary (Table 2), [24]. Additionally, slow or intermittent data retrieval is managed through timeout policies, allowing users to retry or abort the transaction as appropriate. Figure 4 show Dispute resolution process Payment is only released upon client confirmation of data retrieval, with timeouts and optional auditor intervention for unresolved cases. Finally, any attempt by the provider to submit a fake or malicious hash is counteracted by robust content hash verification within the contract, ensuring transparency and auditability using the algorithm shown in Figure 5. Collectively, these mechanisms provide an automated, fair, and trustless approach to dispute resolution, significantly enhancing the security and reliability of the decentralized payment framework.

Table 2: Common dispute scenarios and resolution mechanisms in the proposed IPFS-based fair payment model [18], [24].

Dispute Scenario	Model's Resolution Mechanism
Data not accessible on IPFS	No payment released; refund or timeout logic triggers
Content hash does not match expected data	Payment withheld; client can reject or trigger dispute
User falsely claims non-delivery	Requires explicit confirmation; possible auditor check
Data retrieval is slow or intermittent	Timeout policy; user may retry or abort transaction
Provider submits fake or malicious hash	Content hash verification prevents payment; transparency for auditing

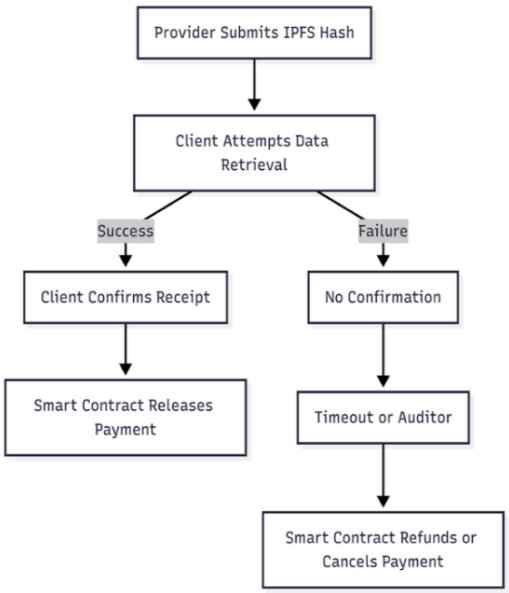

Figure 4: Workflow of dispute resolution process in the IPFS-based fair payment model.

```
Algorithm 1: Dispute Resolution in IPFS-
Based Fair Payment System

Input:
    ipfs_hash      // Content hash provided
by Provider
    timeout        // Maximum time for client
confirmation

1:  Provider submits ipfs_hash and initiates
payment via SmartContract
2:  Client attempts to retrieve and verify
data from IPFS using ipfs_hash

3:  if Client confirms successful retrieval
and verification before timeout then
4:      SmartContract releases payment to
Provider
5:  else
6:      if Auditor or Oracle is available
then
7:          Auditor checks data availability
on IPFS using ipfs_hash
8:          if Auditor confirms data is
available then
9:              SmartContract releases
payment to Provider
10:         else
11:             SmartContract refunds
payment to Client
12:         end if
13:     else
14:         SmartContract refunds payment to
Client after timeout
15:     end if
16: end if

17: SmartContract logs dispute and
resolution status on-chain
```

Figure 5: Pseudocode Representation of the Dispute Resolution Mechanism in the Proposed IPFS-Based Model.

We deployed a minimal escrow contract on Ethereum Sepolia and integrated it with IPFS for off-chain asset addressing. Using the CID QmPtUE…K44j, contract deployment consumed 978,042 gas (5.57 s), while the createDeal and confirmAndPay functions required 186,418 gas (10.85 s) and 61,843 gas (11.51 s), respectively. Complementary local IPFS measurements for ~1 MB assets showed sub-0.03 s upload and ~0.005 s retrieval times with SHA-256 verification. These results, summarized in Table 3, demonstrate efficient content-addressed storage and verifiable retrieval, supporting the feasibility of automating fair settlement conditioned on IPFS-based data verification, with on-chain costs concentrated in the initial deployment and escrow creation steps.

Table 3: Experimental results of escrow contract deployment and IPFS integration.

Operation	Resource/Asset	Gas Used	Time (s)
Contract Deployment	Escrow contract on Sepolia	978,042	5.57
CreateDeal	CID QmPtUE…K44j	186,418	10.85
ConfirmAndPay	CID QmPtUE…K44j	61,843	11.51
IPFS Upload	~1 MB asset	–	0.03
IPFS Retrieval	~1 MB asset	–	0.005

Figure 6. End-to-end prototype: (1) client uploads asset to IPFS and obtains CID; (2) escrow smart contract records CID and funds; (3) client verifies asset off-chain; (4) on successful verification, confirmAndPay releases payment; otherwise, a dispute/timeout path halts payout.

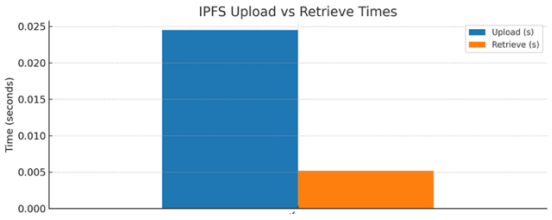

Figure 6: End-to-end prototype of upload and retrieve times.

Nevertheless, it is acknowledged that practical deployment will require additional mechanisms to ensure reliable long-term storage and efficient access, particularly for large or highly dynamic datasets. In summary, these findings support the conclusion that integrating IPFS with blockchain-based payment logic is a promising pathway for building scalable,

transparent, and fully decentralized fair payment systems. Future work will focus on quantitative performance benchmarking, incentive designs for storage reliability, and broader evaluation across diverse application scenarios.

6 CONCLUSIONS

This paper has presented a preliminary exploration of integrating the IPFS as a decentralized off-chain storage solution within blockchain-based fair payment systems. By addressing the limitations of centralized cloud storage, the proposed model leverages IPFS's content-addressed, peer-to-peer architecture to enhance data integrity, transparency, and censorship resistance. The integration of blockchain smart contracts with IPFS enables automated, trustless settlement and robust dispute resolution without reliance on third-party intermediaries. Qualitative analysis and scenario-based evaluation demonstrate that the model effectively mitigates common dispute scenarios and delivers substantial improvements in decentralization and security compared to conventional approaches. The study demonstrates the important and effective role of IPFS as off-chain storage approach in scalable the reach of blockchain applications to make them fully decentralized and trustless. Future work will focus on quantitative performance evaluation, large-scale prototype deployment, and the development of advanced incentive and persistence strategies to further strengthen reliability and practical applicability in real-world decentralized payment systems.

REFRENCSES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] V. Buterin, "A next-generation smart contract and decentralized application platform," White Pap., vol. 3, no. 37, pp. 2-1, 2014.

[3] X. Xu, Q. Lu, Y. Liu, L. Zhu, H. Yao, and A. V. Vasilakos, "Designing blockchain-based applications a case study for imported product traceability," Future Gener. Comput. Syst., vol. 92, pp. 399-406, 2019.

[4] S. Kumar, A. K. Bharti, and R. Amin, "Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions," Secur. Priv., vol. 4, no. 5, p. e162, 2021.

[5] X. Zou, P. Zeng, and H. Cheng, "EFPB: efficient fair payment based on blockchain for outsourcing services in cloud computing," IEEE Access, vol. 11, pp. 30118-30128, 2023.

[6] M. Al-Bassam, "SCPKI: A smart contract-based PKI and identity system," presented at the Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, pp. 35-40, 2017.

[7] F. Casino, E. Politou, E. Alepis, and C. Patsakis, "Immutability and decentralized storage: An analysis of emerging threats," IEEE Access, vol. 8, pp. 4737-4744, 2019.

[8] J. Benet, "Ipfs-content addressed, versioned, p2p file system," ArXiv Prepr. ArXiv14073561, 2014.

[9] E. Daniel and F. Tschorsch, "IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks," IEEE Commun. Surv. Tutor., vol. 24, no. 1, pp. 31-52, 2022.

[10] M. M. Merlec and H. P. In, "Blockchain-based decentralized storage systems for sustainable data self-sovereignty: A comparative study," Sustainability, vol. 16, no. 17, p. 7671, 2024.

[11] V. H. Lakhani, A. Babaei, L. Jehl, G. Ishmaev, and V. Estrada-Galiñanes, "Tit-for-Token: Understanding Fairness when Forwarding Data by Incentivized Peers in Decentralized Storage Networks," ArXiv Prepr. ArXiv230702231, 2023.

[12] C. Rahalkar and D. Gujar, "Content addressed P2P file system for the web with blockchain-based meta-data integrity," presented at the 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), IEEE, pp. 1-4, 2019.

[13] C. Lin, D. He, X. Huang, and K.-K. R. Choo, "OBFP: Optimized blockchain-based fair payment for outsourcing computations in cloud computing," IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 3241-3253, 2021.

[14] D. Manikandan, C. Valliyammai, and R. Karthika, "Blockchain-based secure big data storage on cloud," Int. J. Recent Technol. Eng. IJRTE, vol. 9, no. 4, pp. 37-45, 2020.

[15] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," Inf. Sci., vol. 462, pp. 262-277, 2018.

[16] A. Khanna, A. Sah, V. Bolshev, A. Burgio, V. Panchenko, and M. Jasiński, "Blockchain–cloud integration: A survey," Sensors, vol. 22, no. 14, p. 5238, 2022.

[17] N. Z. Benisi, M. Aminian, and B. Javadi, "Blockchain-based decentralized storage networks: A survey," J. Netw. Comput. Appl., vol. 162, p. 102656, 2020.

[18] C.-L. Chen, Y.-M. Zheng, D.-C. Huang, L.-C. Liu, and H.-C. Chen, "A blockchain and ipfs-based anticounterfeit traceable functionality of car insurance claims system," Sensors, vol. 23, no. 23, p. 9577, 2023.

[19] H. Eren, Ö. Karaduman, and M. T. Gençoğlu, "Security challenges and performance trade-offs in on-chain and off-chain blockchain storage: A comprehensive review," Appl. Sci., vol. 15, no. 6, p. 3225, 2025.

[20] A. R. Naik and B. N. Keshavamurthy, "Next level peer-to-peer overlay networks under high churns: a survey," Peer-Peer Netw. Appl., vol. 13, no. 3, pp. 905-931, 2020.

[21] H. Chen, Y. Lu, and Y. Cheng, "FileInsurer: A scalable and reliable protocol for decentralized file storage in blockchain," presented at the 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), IEEE, pp. 168-179, 2022.

[22] D. Rani, et al., "A secure digital evidence preservation system for an iot-enabled smart environment using ipfs, blockchain, and smart contracts," Peer-Peer Netw. Appl., vol. 18, no. 2, p. 5, 2025.

[23] S. Ma and X. Zhang, "Integrating blockchain and ZK-ROLLUP for efficient healthcare data privacy protection system via IPFS," Sci. Rep., vol. 14, no. 1, p. 11746, 2024.

[24] K. Ahmadi, M. Esmaili, and S. Khorsandi, "A P2P file sharing market based on blockchain and ipfs with dispute resolution mechanism," presented at the 2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings), IEEE, pp. 1-5, 2023.