International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0336
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2343 - 2348

Blockchain-Based Service Level Agreement Verification Framework

M A Anusuya[1], Suhas S[2], Vignesh D[3], M Samudhyatha[4], Adithi H R[5], Sanath P M [6]
[1] Professor, Dept. of CSE, JSS Science and Technology University, Mysuru, Karnataka, India
[2] Assistant Professor, Dept. of CSE, JSS Science and Technology University, Mysuru, Karnataka, India
[3,4,5,6] UG Scholar, Dept. of CSE, JSS Science and Technology University, Mysuru, Karnataka, India
Emails:*anusuya_ma@jssstuniv.in[1], suhas@jssstuniv.in[2], vignesh.d9123@gmail.com[3], ms04bly@gmail.com[4], adithihr23@gmail.com[5], sanathpm333@gmail.com[6]*

Abstract
Traditional Service Level Agreement (SLA) services depend on centralized architectures, where users must inherently trust a single provider for accurate data collection, storage, and reporting. This reliance limits transparency and prevents independent validation of monitoring results. To address these limitations, this paper introduces a decentralized framework for SLA verification that eliminates the need for trust in a central authority. The proposed protocol leverages geographically distributed validator nodes that independently perform service checks and submit signed observations. A quorum-based consensus mechanism is applied to aggregate validator responses, ensuring reliability and reducing the impact of faulty or malicious nodes. Monitoring results are recorded on-chain using the Solana programming model, enabling immutable storage and public verifiability. By utilizing idle computational resources from independent participants, the system forms an economically driven validation network that promotes decentralization and scalability. The design ensures that any external entity can audit and verify uptime records without depending on the protocol operator. Experimental evaluation and comparative analysis indicate that the proposed approach significantly enhances auditability, tamper resistance, and SLA validation capabilities compared to conventional centralized solutions, though it introduces additional system complexity and coordination overhead.

Keywords: *Blockchain, Solana, Decentralized Uptime Monitoring, Cryptographic Verification, Service Level Agreement*

1. Introduction
As the digital tools or websites support trade, talks, and essential networks, they must work most of the time. Modern digital services operate under guarantees known as Service Level Agreements (SLAs), which define the expected availability of a system or application. Common SLA commitments such as 99.9% uptime permit approximately nine hours of downtime annually, while stricter guarantees like 99.99% reduce allowable downtime to less than one hour per year. Traditionally, uptime verification has been performed by centralized monitoring providers including Pingdom and UptimeRobot. These platforms continuously monitor service availability and generate uptime statistics for clients. However, the monitoring process remains dependent on trust in the provider, since the same organization controls data collection, storage, and reporting. As a result, customers have limited ability to independently verify whether outage records are complete or whether uptime statistics have been altered. This centralized trust model creates concerns regarding transparency, data integrity, and the possibility of selective omission or manipulation of monitoring results. Blockchains such as Solana offer immutable, anti-tamper ledgers with quick programmatic execution and sub-cent transaction fees. The InterPlanetary File System (IPFS), introduced by Benet (2014), provides distributed storage where file identity is derived directly from content. Together, these technologies enable a monitoring architecture in which uptime observations are cryptographically signed by independent validators, aggregated into verifiable reports, and anchored on-chain in a manner any individual can independently reproduce and audit. This paper presents a Blockchain-Based SLA Verification Framework, including formal system architecture, detailed round lifecycle, cryptographic verification

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0336
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2343 - 2348

procedures, experimental results and economic security analysis.

2. Literature Review

Existing uptime monitoring services traditionally rely on centralized infrastructure, while blockchain research has been explored independently on tamper-proof data attestation and stake-based accountability. This section briefs on the relevant work across these areas and highlights the gaps in areas like monitoring data verifiability, immutable attestation, and accountability. The proposed protocol architectural frame work aims to address the above gaps by discussing in the below section. Prior work has explored using blockchains as tamper-evident logs for data attestation. Androulaki et al., (2018) describe Hyperledger Fabric's approach to permissioned ledger where all participants are known and accountable, this assumption that all participants are accountable does not hold in open, decentralized monitoring environments. The proposed protocol addresses this by operating on a permissionless public chain, where any individual can register as a validator using their idle computing resources, and the integrity is enforced by on-chain staking and cryptographic signatures as discussed by Bernstein et al., (2012). Buterin (2014) introduced Ethereum's smart contract enabling on-chain program execution and data storage but with low throughput. Solana, introduced by Yakovenko (2018), extends this paradigm with a high-throughput, low-latency architecture based on Proof of History (PoH), enabling theoretical throughput of 65,000 Transaction Per Second (TPS) with sub-second finality and sub-cent transaction fees. The proposed monitoring protocol thus leverages Solana due to its high throughput, making it well suited for high frequency anchoring operations

3. System Architecture

Layer by layer, the setup stacks up to five levels, each one having specific duties as described below: The proposed system consists of validator nodes that independently monitor service availability, a hub verifier that aggregates validator submissions, a smart contract deployed on the Solana blockchain for immutable record storage, an IPFS layer for storing detailed uptime reports, and an off-chain application layer containing the backend API and frontend interface for user interaction.

3.1.Validator Node Architecture

Each validator node operates as an independently managed containerized process deployed on unused computers across different geographic regions. Every validator maintains a persistent Ed25519 public-private key pair (skv, pkv).As shown in Algorithm 1 specifies the validator submission procedure.

Algorithm 1 Validator Submission Procedure

Input: skv, pkv, target_url,
round_timestamp, timeout τ
Output: Signed submission payload

```
t_start ← current_time()
response ← HTTP_HEAD(target_url, τ)
latency ← current_time() − t_start
if response.status ∈ [200, 299] then
    is_up ← 1
else
    is_up ← 0
end if
payload ← {
   target_id: SHA256(normalize(target_url)),
   round_ts: round_timestamp,
   is_up:    is_up,
   latency:  latency,
   pubkey:   pkv
}
σ ← Sign(skv, serialize(payload))
return { payload, signature: σ }
```

3.2.Hub Verifier Architecture

The hub verifier H is the coordinator of the proposed protocol. Upon receiving a validator submission, the hub executes the verification pipeline defined in Algorithm 2. Region-wise aggregates are computed by the hub upon receiving the submission by mapping the source IP address to a continent using a GeoIP database as documented by IP-API (2022). Region is stored in the IPFS report but not anchored on-chain to minimize chain state. The hub triggers round finalization when either of two conditions is met: the count of accepted submissions for a round reaches the

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0336
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2343 - 2348

quorum threshold Q (configurable, default Q = 3), or the round timeout Tround elapses (default Tround = 120 seconds). Algorithm 3 specifies the round finalization procedure.

Algorithm 2 Validator Submission Verification Pipeline

Input: submission S = (payload, σ)
Output: ACCEPT | REJECT

1: $pk_v \leftarrow$ payload.pubkey
2: // Check 1: Signature validity
3: if ¬Verify(pk_v, serialize(payload), σ)
4: return REJECT
5: round ← rounds[payload.target_id+payload.round_ts]
6: // Check 2: Validator is registered
7: if $pk_v \notin$ round.expected_validators
8: return REJECT
9: // Check 3: No duplicate for this round
10: if $pk_v \in$ round.submissions
11: return REJECT
12: // Check 4: Round is still open
13: if round.state = CLOSED
14: return REJECT
15: round.submissions.add(S)
16: return ACCEPT

Algorithm 3 Round Finalization Procedure

```
Input:  round_id, submissions S = {s_1,...,s_n}
Output: IPFS CID, on-chain tx_hash
uptime ← (Σ s_i.is_up) / |S| × 100
latencies ← sort([s_i.latency for s_i ∈ S])
median_lat ← latencies[⌊|S|/2⌋]
report ← build_json(S, uptime, median_lat)
cid ← IPFS_upload(report)
report_hash ← cid
tx ← build_solana_tx(
    instruction: submit_round,
    target_id:   round.target_id,
    round_ts:    round.timestamp,
    uptime_pct:  uptime,
    median_lat:  median_lat,
    report_hash: report_hash
)
tx_hash ← Solana_send(tx)
distribute_rewards(S)
return cid, tx_hash
```

3.3.On-Chain Program (Solana)

The Solana program manages multiple account types for handling targets, validators, staking, rewards, and finalized monitoring rounds. Target accounts store the normalized target identifier and creation metadata, while validator accounts maintain the validator public key, staked lamports, and validator status. Separate accounts are used for staking pools and validator reward distribution. Round summary accounts store finalized monitoring results including the target identifier, round timestamp, uptime percentage, median latency, and SHA-256 hash, discussed by National Institute of Standards and Technology (2015), of the uploaded report. These accounts are created using Solana Program Derived Addresses (PDAs), allowing secure program-controlled account management without requiring private keys.

3.4.IPFS Storage Layer

The monitoring report stored in IPFS follows a structured JSON format containing validator submissions and aggregated round results.

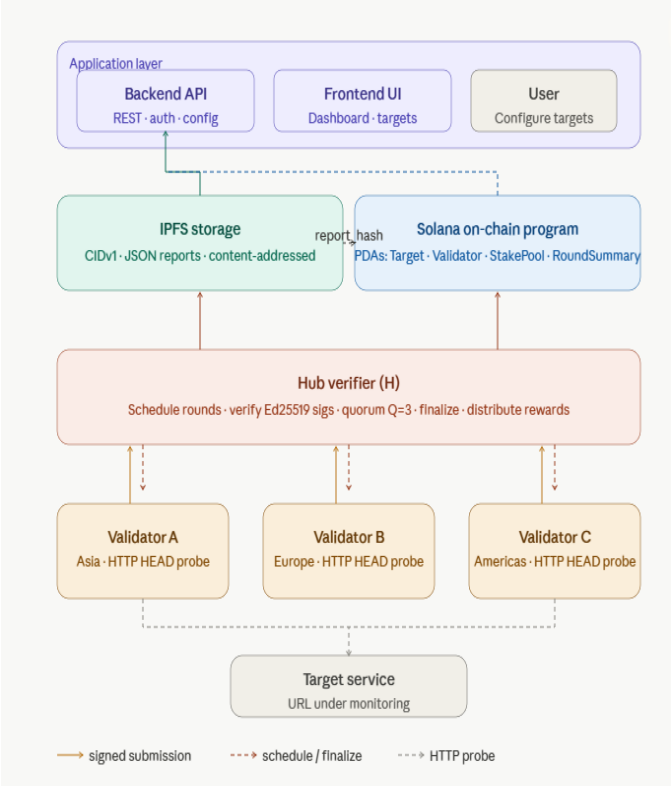

Figure 1 System architecture of the decentralized SLA verification protocol

Each validator entry includes the validator public key, geographic region, measured latency in milliseconds, service availability status, and a cryptographic signature for verification. The

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0336
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2343 - 2348

aggregation section stores the finalized uptime percentage and the median latency calculated from all validator responses within the monitoring round. An independent verifier can verify that the fetched IPFS document matches the on-chain commitment by comparing the report_hash on-chain and the IPFS document hash, eliminating the need to trust the IPFS gateway.Shown in Figure 1 System architecture of the decentralized SLA verification protocol Figure 1 illustrates the five-layer system architecture, comprising geographically distributed validator nodes, a central hub verifier, the Solana on-chain program, IPFS storage, and the application layer, with arrows denoting the flow of signed submissions, round scheduling, and finalized report anchoring.

4. Methodology

The proposed uptime monitoring protocol operates through a coordinated workflow involving validator nodes, the hub verifier, decentralized storage, and on-chain verification. Before monitoring begins, submitted target URLs are normalized to ensure that logically identical URLs generate the same identifier. The normalized URL is hashed using SHA-256, producing a unique target identifier that is later used for scheduling and on-chain record management. Monitoring rounds are scheduled at fixed time intervals depending on active user subscriptions which can be one of {5, 15, 30, 60} minutes. If multiple users are subscribed to the same target with different check intervals, the system schedules monitoring rounds using the minimum subscribed interval. For example, if one user selects a 5-minute interval while another selects a 30-minute interval, the target is monitored every 5 minutes. Users with larger intervals only view the rounds relevant to their subscription period through client-side filtering. For every round, the hub verifier assigns monitoring tasks to validators distributed across different regions. To prevent duplicate participation within a monitoring round, validators authenticate themselves using timestamp-based signed requests generated through Ed25519 cryptography. Once authenticated, validators perform availability checks by sending HTTP requests to the target service and measuring the response latency. A service is considered operational only if a valid response is received within the configured timeout period. Each validator then submits a signed observation containing its public key, measured latency, uptime status, and regional information. After collecting sufficient validator responses, the hub verifier computes the final uptime percentage and median latency for the round. The complete monitoring report is serialized into a deterministic JSON structure and uploaded to IPFS. A SHA-256 hash of the serialized report is then committed on-chain through the Solana program, creating an immutable reference to the monitoring results. This design ensures that any modification to the report content produces a different hash value, making tampering detectable. Clients can independently verify monitoring data without relying entirely on the provider. Using the target URL and round timestamp, a client can reconstruct the expected on-chain account, retrieve the stored report hash, recompute the hash from the IPFS report, and verify validator signatures to confirm that the published monitoring results are accurate and unmodified.

5. Experimental Results

The proposed protocol was deployed on Solana devnet and evaluated on anchoring latency, transaction cost, quorum consensus, and independent verification. This section presents quantitative results demonstrating that the proposed system is accurate, cost efficient, and fully auditable without relying on a trusted backend. The deployment consisted of five geographically distributed validator nodes. Monitoring was carried out continuously for 7 days on 6 different web services (including a few test services) using a fixed 5-minute observation interval. With this configuration, each target generated slightly more than 2000 monitoring rounds during the testing period. During the 7-day deployment, validator agreement was achieved in 99.7% of monitoring rounds for high-availability targets. Only a small number of disagreement events occurred, mainly due to temporary regional network instability rather than protocol failure. Controlled outages were intentionally introduced on test services during the experiment. The protocol successfully detected each outage within one or two monitoring intervals. Since the final result depends on majority consensus,

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0336
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2343 - 2348

isolated regional failures did not incorrectly classify active services as offline. This shows that decentralized validation provides more stable and globally representative uptime measurements. The protocol also showed low operational overhead during testing. The average on-chain anchoring latency remained below four seconds, while the average transaction fee per finalized round was approximately 0.000025 SOL on the Solana devnet. Experimental observations further showed that increasing the monitoring interval reduced the total anchoring cost because fewer monitoring rounds were generated.

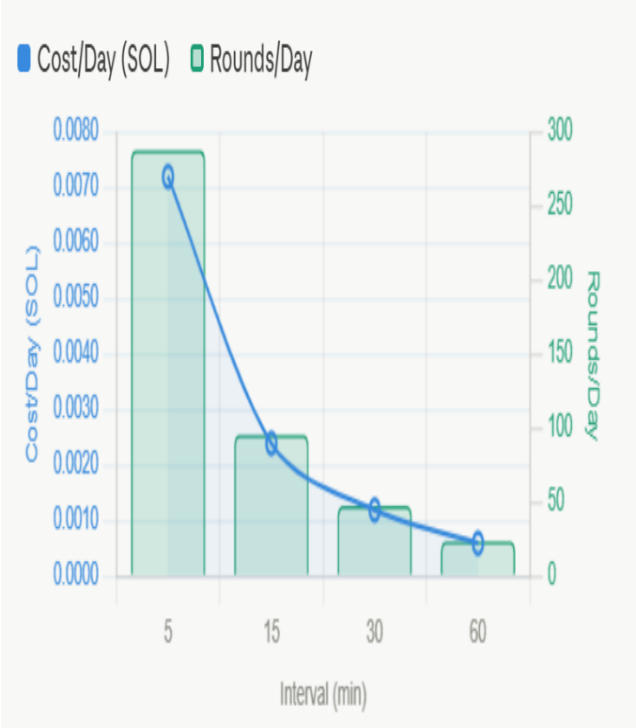

Figure 2 Round count and SOL anchoring cost across subscription intervals

Figure 2 illustrates how increasing the monitoring interval inversely reduces both the number of daily anchoring rounds and the corresponding SOL cost, demonstrating that less frequent checks yield proportionally lower on-chain transaction costs. Independent verification experiments confirmed that monitoring data could be audited without relying on the provider infrastructure. Historical rounds were successfully verified by recomputing report hashes stored through IPFS and validating Ed25519 signatures. Verification time increased linearly with the number of validator submissions, matching the expected computational complexity.

6. Comparison Between Centralized and Proposed Decentralised Monitoring Services

The following table summarizes the comparison between centralized monitoring services and our proposed decentralized monitoring protocol.

Table 1 Comparison of Centralized Services vs. Proposed Protocol

Feature	Centralized Monitoring	Proposed Protocol
Trust Model	Provider trust	Cryptographic verification
Data Storage	Private database	On-chain + IPFS
Tamper Resistance	No	Yes
Validator Transparency	Hidden	Public identities
Economic Incentives	None	Stake-backed rewards
Independent Verification	Not supported	Fully supported
SLA Auditability	Weak	Strong (on-chain)
Sybil Resistance	None	Capital cost per identity
Operational Complexity	Low	High (validator coordination, Solana integration, IPFS pinning)

Table 1 demonstrates that the proposed decentralized monitoring protocol introduces higher operational complexity, by providing significant advantages over centralized services in verifiability, tamper resistance, transparency, and accountability. These results were compared and tabulated when experimented centralized over the decentralized system architectures.

7. Conclusions And Future Enhancements

The work presented in this paper explored a

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0336
e ISSN: 2584-2854
Volume: 04 Issue: 06
June 2026
Page No: 2343 - 2348

blockchain-based approach for transparent SLA verification framework using the Solana ecosystem. Unlike traditional monitoring platforms where uptime records are controlled entirely by a single organization, the proposed protocol distributes observation and verification across multiple independent validators. Monitoring results are digitally signed using Ed25519 cryptography, stored through IPFS, and anchored on-chain using immutable hash commitments. This combination enables any client to independently validate the authenticity and correctness of historical uptime records. Although the current implementation achieves decentralized verification, some areas remain open for improvement. The protocol presently relies on a single hub verifier responsible for collecting validator submissions and finalizing monitoring rounds. Future versions may introduce multiple hub verifiers that cross-validate monitoring results before blockchain commitment, further reducing trust assumptions. In addition, current on-chain storage optimization techniques remove older round accounts after a limited duration to reclaim rent costs. More advanced approaches, such as Merkle root commitments, discussed by Liu, Z. et al., (2023), could preserve historical verification data on-chain more efficiently. Hence our protocol proposes a strong verifiable fact anchored for public and decentralized system that serves as neutral, audit-ready infrastructure for enterprise SLA compliance adopted for service availability attestation proved cryptographically.

References

[1]. Benet, J. (2014). IPFS – Content addressed, versioned, P2P file system, arXiv:1407.3561.
[2]. Androulaki, E. et al., (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. Proceedings of the 13th EuroSys Conference, Porto, Portugal, 1–15. doi: 10.1145/3190508.3190538.
[3]. Bernstein, D.J., Duif, N., Lange, T., Schwabe, P. and Yang, B.-Y. (2011). High-speed high-security signatures. Cryptographic Hardware and Embedded Systems – CHES 2011, Lecture Notes in Computer Science, Vol. 6917, Springer, 124–142. doi: 10.1007/s13389-012-0027-1.
[4]. Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. Ethereum Whitepaper. [Online]. Available: https://ethereum.org/en/whitepaper/.
[5]. Yakovenko, A. (2017). Solana: A new architecture for a high performance blockchain. Solana Whitepaper. [Online]. Available: https://solana.com/solana-whitepaper.pdf.
[6]. Nazarov, S. and Ellis, S. (2017). Chainlink: A decentralized oracle network. Chainlink Whitepaper. [Online]. Available: https://chain.link/whitepaper.
[7]. IP-API (2022). IP API documentation. [Online]. Available: https://ip-api.com/docs.
[8]. National Institute of Standards and Technology (2015). Secure Hash Standard (SHS). FIPS PUB 180-4, U.S. Department of Commerce, Washington, D.C. doi: 10.6028/NIST.FIPS.180-4
[9]. Liu, Z. et al., (2023). Data integrity audit scheme based on quad Merkle tree and blockchain. IEEE Access, Vol. 11, 59263–59273.doi:10.1109/ACCESS.2023.3240066
[10]. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf. Pingdom: https://en.wikipedia.org/wiki/Pingdom.
[11]. Rahul Singh. What is UptimeRobot and use cases of UptimeRobot, 30 January 2024. URL: https://devopsschool.com/blog/what-is-uptimerobot-and-use-cases-of-uptimerobot/.