

inMediaVeritas: A Decentralized News Verification Application Using zk-SNARKs and Blockchain Consensus

Jakub Iváček, Patrik Baran, Tomáš Tisovský, Vojtech Babinský, Tomáš Miština, Matúš Hrkáč,
Richard Gazdík, Ivan Homoliak
inMediaVeritas Group Project
STU FIIT, Slovakia Bratislava

Abstract—The proliferation of misinformation in digital media demands trustworthy, censorship-resistant verification mechanisms that operate without relying on any central authority. This paper presents inMediaVeritas, a blockchain-based decentralized application (DApp) designed to restore public trust through a three-tier architecture involving verified reporters, community witnesses, and news agencies. The system leverages zk-SNARKs for privacy-preserving location verification, IPFS for decentralized media storage, and on-chain consensus to collectively establish event truthfulness. Beyond the core verification pipeline, we describe the companion applications that complete the system: an identity wallet for reporters and witnesses, an authority application for credential issuers, and a news agency subscription interface. We detail the system architecture, cryptographic mechanisms, incentive design, implementation experience, and key limitations.

Index Terms—blockchain, zk-SNARKs, decentralized verification, misinformation, IPFS, Ethereum, digital identity, news provenance

I. INTRODUCTION

The rapid and largely unchecked spread of false information across digital platforms poses a systemic threat to democratic institutions, public health discourse, and social cohesion [1]. Traditional fact-checking pipelines depend on centralized intermediaries—editorial boards, platform algorithms, or designated verification services which introduce single points of failure, susceptibility to political or commercial bias, and a fundamental asymmetry of trust: users must accept the authority of the verifier without independent recourse.

Decentralized technologies offer a meaningful alternative. By distributing the verification burden across a network of economically incentivized participants whose behavior is governed by cryptographically enforced smart contracts, it becomes possible to construct a verification system that no single party can capture, censor, or manipulate. The blockchain’s immutability provides a permanent, auditable record of every submission, vote, and consensus outcome, while zero-knowledge proofs allow participants to assert facts about themselves (such as their physical proximity to an event) without exposing private data.

inMediaVeritas applies these ideas to the full news lifecycle. Reporters upload media with cryptographic proof of their location at the time of capture. Community members geographically

proximate to the event submit testimonials verified through zk-SNARKs. News agencies then pay subscription fees to access and license verified, blockchain-attested media. The result is a pipeline from raw media capture to verified, licensable content that requires no trusted intermediary to arbitrate truth.

This paper makes the following contributions:

- 1) A complete three-tier architecture for decentralized news verification combining zk-SNARKs, IPFS, and Ethereum smart contracts.
- 2) A credibility-weighted witness consensus protocol with slashing incentives for dishonest behavior.
- 3) Companion applications spanning the participant tiers: an implemented identity wallet for participants and an authority application for identity issuers, together with the design of a news agency subscription and access interface.
- 4) A proof-of-concept implementation of the on-chain voting contract and the zero-knowledge circuits, with a performance and gas-cost evaluation, and a discussion of the practical challenges encountered and the limitations of the current design.

II. RELATED WORK

A. Automated Fact-Checking

NLP-based systems such as ClaimBuster [2] automate claim detection and routing to fact-checkers, but are limited by language ambiguity, domain specificity, and the latency of human review cycles. Rule-based and machine-learning pipelines have been surveyed extensively [11], revealing a consistent gap: even high-performing models produce no verifiable provenance record and offer no resistance to adversarial rephrasing. More recent large-language-model approaches improve fluency and coverage but remain susceptible to confident hallucination [12]; their outputs cannot be audited or contested by third parties after the fact. inMediaVeritas sidesteps this class of limitation by grounding truth not in model confidence but in cryptographically attested, community-driven consensus.

B. Blockchain Provenance and Smart Contracts

The notion of a programmable, self-enforcing agreement on a shared ledger was introduced by Szabo [13] well before a practical substrate existed. Bitcoin [14] provided that

substrate, and Ethereum [15], [16] extended it with a Turing-complete execution environment whose state transitions are deterministically reproducible by any network participant.

Blockchain-based provenance systems [3] build on this foundation to provide immutable audit trails for digital media, establishing that a given artifact existed at a given time and has not been subsequently altered. Advancements in the space include content-fingerprinting schemes that anchor SHA-256 hashes of media files on-chain [17], making post-hoc tampering immediately detectable by any verifier. However, these systems share a fundamental limitation: provenance confirms *what* was recorded, not whether the depicted events are truthful. *inMediaVeritas* addresses this gap by layering a community verification step on top of the provenance record.

C. Zero-Knowledge Proofs

Zero-knowledge proofs allow a prover to demonstrate knowledge of a secret witness satisfying a constraint system without revealing any information beyond the validity of the statement [18]. The Groth16 proving system [19] reduced the proof size to three group elements and verification to a constant number of pairing operations, making on-chain verification economically feasible. Zk-SNARKs were introduced to wide deployment through Zcash [4], and subsequent systems such as PLONK [5] removed the need for a circuit-specific trusted setup. The Circom language [6] and the Baby JubJub curve [9] optimized for arithmetic inside zk-SNARK circuits further lowered the barrier to practical circuit authorship. These advances collectively enable *inMediaVeritas* to prove location membership and credential validity on consumer hardware in under 300 ms (see Section VIII), a prerequisite for real-world usability by journalists in the field.

D. Decentralized Identity

The W3C Decentralized Identifiers (DID) specification [20] defines a globally resolvable, controller-owned identifier that requires no central registry. Verifiable Credentials [21], built atop DIDs, provide a standardized envelope for signed, machine-verifiable claims about a subject. The EU’s eIDAS 2.0 regulation and the associated EUDI Wallet architecture extend these concepts to government-issued identity documents, providing a blueprint for cross-border digital identity that *inMediaVeritas* adapts at the participant onboarding layer.

E. Decentralized Storage

IPFS [7] provides content-addressed, censorship-resistant media storage in which content is identified by its cryptographic hash, making any modification to a stored artifact immediately detectable. Filecoin [22] complements IPFS by adding a cryptoeconomic incentive layer that compensates storage providers for maintaining data availability over time; without such incentives, IPFS pinning depends on operator goodwill. Arweave [23] takes an alternative approach, encoding a one-time storage fee into a “blockweave” structure designed to guarantee permanent retention. For *inMediaVeritas*, long-term availability of the stored media is a critical requirement,

and these incentive mechanisms represent a natural extension to the current Kubo-based pinning approach.

F. Prior Decentralized Journalism Platforms

The Civil platform [8] combined IPFS storage with blockchain-based governance tokens for journalism, but lacked a robust community verification mechanism and ultimately collapsed due to adoption and token-economics challenges. The Po.et project [24] similarly anchored article metadata and licensing terms on the Bitcoin blockchain, providing provenance without verification. NewsGuard and related trust-rating services take a centralized editorial approach, reintroducing the single point of failure that decentralized architectures aim to eliminate. *inMediaVeritas* addresses the shared shortcomings of these prior systems by integrating cryptographically enforced witness voting with a self-sustaining protocol incentive layer, replacing editorial gatekeeping with on-chain consensus.

III. SYSTEM ARCHITECTURE

inMediaVeritas is organized around three interacting actor tiers. All persistent state is anchored to a public blockchain (Ethereum or a compatible L2 network), with media stored on IPFS and referenced by on-chain content identifier (CID) hashes.

A. Reporter Layer

Reporters are credentialed participants whose real-world identities are verified through a decentralized identity (DID) registry prior to their first submission. Each news submission bundles three elements committed atomically on-chain via a smart contract: a media artifact stored on IPFS and referenced by its CID, a precise timestamp, and a SHA-256 integrity hash of the raw media. Reporters stake reputation tokens at submission time; demonstrated misconduct triggers slashing, creating a direct economic disincentive for fabricated or misleading reports.

B. Witness Layer

Community witnesses are users who are geographically proximate to a reported event. Each witness submits a testimonial (confirm or deny) accompanied by a zk-SNARK proof that their location at the time of the event falls within the claimed event radius, without revealing their precise coordinates. Witness votes are weighted by each witness’s historical *credibility score*, which is derived from the alignment of their past votes with the eventual consensus outcomes of resolved events. This mechanism rewards accurate and honest engagement over time, and discourages both coordinated brigading and indiscriminate voting.

The witness contract enforces a minimum quorum before any event is resolved. If quorum is not reached within a configurable time window, the event is marked as unresolved and the reporter’s stake is returned. Resolved events are permanently recorded on-chain with the aggregate credibility-weighted vote outcome.

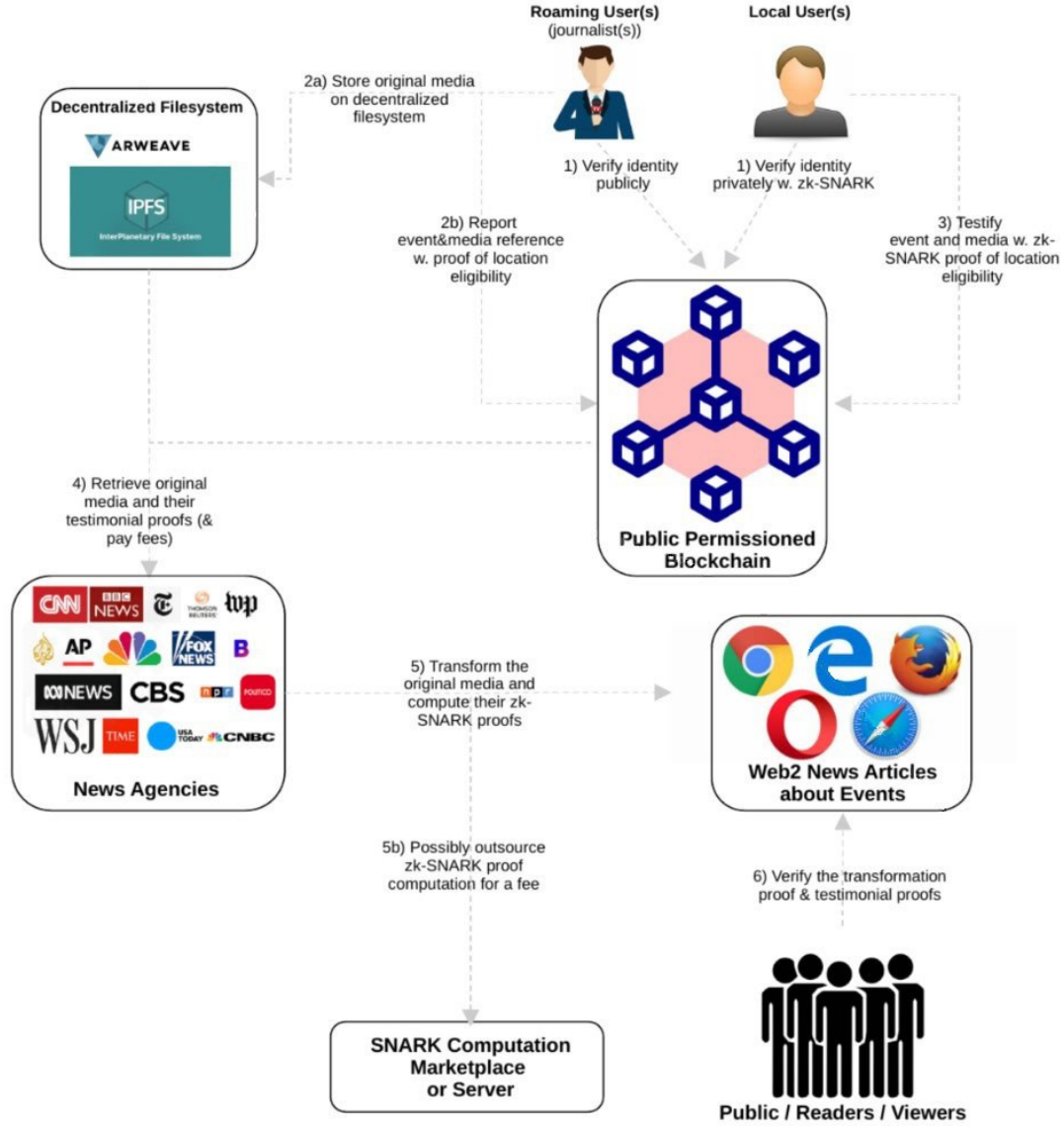


Fig. 1. Architecture overview of *inMediaVeritas*: the three actor tiers, IPFS/Arweave media storage, the permissioned blockchain, and the zk-SNARK-backed verification and licensing flow.

C. News Agency Layer

News agencies access verified media and the associated on-chain attestation records by purchasing weekly, monthly, or yearly subscription plans. Subscription fees flow into a shared protocol pool that funds reporter submissions and witness votes, ensuring the system is economically self-sustaining without external grants or centralized governance. Agencies also receive cryptographic proofs, generated on demand, confirming that any published artifact is unaltered and that the associated on-chain record is authentic.

The news agency interface provides subscribing organizations with a dashboard for discovering, filtering, and licensing verified content. Agencies can search the on-chain event registry by location, time range, topic tags, and consensus confidence

score. For each event, the interface displays the associated media, the full witness vote breakdown, and a downloadable cryptographic proof of authenticity suitable for inclusion in published reporting. Subscription management, including plan selection and renewal, is handled through a straightforward web front-end that interacts with the subscription smart contract.

IV. COMPANION APPLICATIONS

In addition to the core smart contract infrastructure, companion applications were developed to make the system accessible to non-technical participants and to complete the end-to-end user flow. The two applications described below—the identity wallet and the authority application—were implemented; the

news agency interface outlined in Section III-C remains at the design stage.

A. Identity Wallet

The identity wallet is a mobile-first application through which reporters and witnesses manage their decentralized identities and interact with the *inMediaVeritas* protocol. It is inspired by the EUDI/eIDAS standard for identity wallets in the EU. No suitable existing solution was found, thus it was decided to create a custom tailored application that suits the needs of this project. Its primary functions are:

- **Credential storage:** Holding verifiable credentials issued by the authority application (described below), which attest to the user’s registration status as a reporter or witness.
- **Submission and voting:** Allowing reporters to compose and submit news events capturing media, attaching a location commitment, and signing the transaction and allowing witnesses to cast credibility-weighted votes with automatically generated zk-SNARK location proofs.
- **Biometric-Protected Signing:** Rather than requiring users to manually manage 256-bit private keys, the app utilizes *react-native-biometrics*. This allows users to sign protocol messages (reports, votes, and registrations) using keys stored in the device’s Secure Enclave or Keystore, triggered by FaceID or Fingerprint.

B. Authority Application

The authority application is a simplified web interface for credential issuers (such as governmental bodies or trusted news organizations) who are responsible for verifying the real-world identities of reporters and witnesses before admitting them to the protocol. Its functions include:

- **Identity verification:** Reviewing submitted government-issued credential proofs and approving or rejecting applicants.
- **Credential issuance:** Issuing signed verifiable credentials to approved participants, which are then stored in their identity wallets and presented when required by the smart contract.
- **Audit log:** Maintaining a tamper-evident log of all issuance actions.

The authority application is deliberately minimal by design: its role is to bootstrap trust at the identity layer without becoming a bottleneck or a central point of control over content decisions. In the future multiple authority applications can be used, (each government can have its own solution) but they all have to support API for communication with the ID app.

V. CRYPTOGRAPHIC MECHANISMS

A. Zero-Knowledge Location Proofs

The core privacy primitive is a zk-SNARK circuit that accepts a witness’s location and their DID-linked identity commitment as private inputs, and proves that (i) the location falls within the declared event radius, and (ii) the identity commitment corresponds to a registered participant, without revealing either the exact location or the underlying identity.

We describe this circuit in the Circom language [6] and prove it using the Groth16 proving system over the BN254 (alt_bn128) curve [19], for which Ethereum provides native precompiles enabling efficient on-chain verification. The Baby JubJub elliptic curve [9]—optimized for arithmetic inside zk-SNARK circuits—is used only for the elliptic-curve operations embedded within the circuit, not as the proving curve. Proof generation completes in a few seconds on consumer hardware (see Table I); on-chain verification is performed by a Solidity verifier contract generated from the circuit’s verification key (Table III).

B. Media Integrity

All uploaded media is SHA-256 hashed client-side before upload and the hash is committed on-chain at submission time, creating an immutable fingerprint of the original content. The IPFS CID independently encodes the content hash of the stored artifact. Together, these two independent integrity checks ensure that any tampering with the media after submission is immediately detectable by any party with access to the original artifact.

C. Verifiable Credentials and Identity

Reporter and witness identities are anchored to Identity Credentials, each linked to a government-issued credential via a zero-knowledge credential proof. This proof demonstrates that the holder possesses a valid government-issued identity document without revealing the document’s contents or the holder’s name or address. Credentials are issued by registered authority nodes and stored in participants’ identity wallets. The smart contract verifies credential validity on-chain before accepting submissions or votes.

VI. INCENTIVE MODEL

inMediaVeritas is designed to be economically self-sustaining. News agency subscription fees flow into a shared protocol pool, from which: (i) reporters receive a gas reimbursement and a small reward upon successful event resolution, (ii) witnesses receive a proportional share of the resolution reward based on their credibility-weighted contribution to the correct consensus, and (iii) stakes are slashed from dishonest participants and redistributed to the pool.

This design creates a virtuous cycle: agency demand for verified content directly funds the participation of reporters and witnesses, who are in turn incentivized to report accurately and vote honestly. No external funding source, governance token, or centralized treasury is required to sustain the system.

A key design choice is that reporters and witnesses are not required to provide their own capital. The protocol pool pre-funds participation costs, lowering the barrier to entry for individual contributors. Slashing applies only to pre-funded stakes, not to personal funds, preventing punitive outcomes for good-faith participants who encounter edge cases.

VII. IMPLEMENTATION

The system was implemented as a proof-of-concept targeting an Ethereum-compatible network (the Sepolia testnet or a comparable L2). Smart contracts were written in Solidity and developed using the Foundry toolchain; gas costs were measured separately on the Remix JavaScript VM as described in Section VIII. The zk-SNARK circuits were implemented using the Circom language. On mobile, witness calculation used `circom-witnesscalc`—which compiles circuits into graph operations for efficient on-device computation—and proofs were generated with `react-native-rapidsnark`. On desktop, witness generation and Groth16 proving used the SnarkJS toolchain (`powers-of-tau` setup, `groth16` setup, and `groth16` prove). The Solidity verifier contract was generated from each circuit’s verification key via `snarkjs zkey export solidityverifier`. IPFS storage was integrated via a local Kubo node, accessed over its HTTP RPC API.

The identity wallet was built as a React Native application, targeting deployment on Android smartphones. The authority application was implemented as a standard React web application. Both front-ends sign and broadcast blockchain transactions directly via `ethers.js`. Media uploads and retrieval are routed through a lightweight backend service that relays content to a local Kubo IPFS node (its HTTP RPC for uploads and its gateway for reads). This service holds no trust authority: because IPFS content identifiers are self-verifying content hashes and all consensus state lives on-chain, the backend can neither tamper with stored media nor influence verification outcomes, and could be replaced by a public IPFS gateway or an in-browser node without affecting the protocol’s trust model.

VIII. EVALUATION

We evaluated the key performance characteristics of the *inMediaVeritas* proof-of-concept. On-chain gas costs were measured on a local EVM (the Remix JavaScript VM), which executes identical bytecode and charges identical gas to a public network such as Sepolia; the measured values are therefore representative of mainnet/testnet execution. zk-SNARK proof generation was measured using the SnarkJS CLI.

A. Experimental Setup

On-chain gas measurements were performed on the Remix VM (Solidity 0.8.13, optimizer enabled, `runs=200`, `viaIR=true`) [10]. zk-SNARK proof generation was measured with `react-native-circom-witnesscalc` (0.0.1-beta.1) and `react-native-rapidsnark` (0.0.1-beta.2) using the Groth16 proving system on the following devices:

- Pixel 6 Pro
- Samsung S23 Ultra

Additionally it was also measured with SnarkJS v0.7.6 (Node.js 20.17.0):

- **Desktop:** AMD Ryzen 5 7600X (6 cores / 12 threads, 4.7 GHz), 32 GB RAM, Windows 11 with WSL 2 (Ubuntu).

TABLE I
ZK-SNARK TIMING

Device / Circuit	Witness gen. (ms)	Proof gen. (ms)	Total (ms)
Pixel Identity	10.09 ± 3.95	157.74 ± 8.28	276.66 ± 56.13
Pixel Location	8.46 ± 2.10	155.09 ± 9.28	254.31 ± 8.30
Samsung Identity	8.90 ± 2.46	90.65 ± 11.87	186.53 ± 32.06
Samsung Location	9.28 ± 1.92	105.25 ± 23.61	205.63 ± 26.66
Desktop Identity	1 121	1 454	2 575
Desktop Location	951	1 411	2 362

The proving curve is BN254 (alt_bn128); the Baby JubJub curve is used only as the embedded curve inside the circuits.

B. zk-SNARK Proof Generation

The identity circuit is compiled with the Groth16 proving scheme. Testing on desktop with `snarkjs` required approximately 7 680 arithmetic constraints, while compiled circuits on mobile devices used 4 505 constraints.

Table I reports witness generation and proof generation times, with the total also including function setup. Results for mobile devices include average of 10 runs, including standard deviation. The final Groth16 proof is 192 bytes and is submitted on-chain as part of the `voteOnMediaWithProof` transaction.

On mobile devices, witness generation and proof generation are performed client-side using `circom-witnesscalc` and `rapidsnark`. The Circom circuits are compiled into graph operations that enable efficient witness calculation on resource-constrained devices. With the 42% reduction in constraint count compared to desktop (4,505 vs. 7,680 constraints) and a different framework, mobile proof generation remains substantially faster, with end-to-end times roughly 10× quicker than their desktop counterparts. On a Pixel 6 Pro, the identity circuit completes in 277 ms; on a Samsung Galaxy S23 Ultra, it completes in 187 ms. Witness calculation accounts for under 10 ms on all tested devices, with proof generation as the dominant cost. This client-side proving approach avoids server load and latency.

The reported desktop times correspond to a cold run; the witness generation time ($\approx 1\,121$ ms) includes Node.js startup overhead of approximately 900 ms, with the cryptographic work itself accounting for the remainder. The two circuits exhibit similar costs, consistent with their comparable constraint counts; the location circuit is marginally faster despite producing a larger public input vector. These desktop figures reflect the `snarkjs` proving path on a high-performance CPU, where a cold proof already costs on the order of seconds and would be slower still on laptop-class hardware. The mobile path—the system’s primary deployment target—is roughly an order of magnitude faster (under 300 ms) despite running on far weaker hardware, because the native `circom-witnesscalc/rapidsnark` pipeline is considerably more efficient than `snarkjs`. On-device usability is therefore governed by the choice of an optimized native proving framework rather than by raw hardware performance, which is precisely why mobile clients do not rely on the `snarkjs` path used for desktop measurement.

TABLE II
GAS CONSUMPTION PER PROTOCOL OPERATION
(REMIX VM, SOLIDITY 0.8.13, OPTIMIZER ON)

Operation	Gas (units)	ETH at 10 gwei
Contract deployment	1 649 455	0.016495
createEvent (0 media)	280 774	0.002808
createEvent (1 media)	450 615	0.004506
createEvent (3 media)	735 659	0.007357
addMediaItem	212 567	0.002126
voteOnEvent (first, local)	95 533	0.000955
voteOnEvent (subsequent, local)	61 333	0.000613
voteOnMedia (first, non-local)	76 190	0.000762
voteOnMedia (subsequent, local)	81 307	0.000813

Circuit setup is performed only once per circuit (depending on its size, powers of tau might be reused); subsequent proof requires only witness calculation and proving.

C. On-Chain Gas Costs

Table II reports the gas consumed by each state-changing protocol operation. At a reference gas price of 10 gwei the corresponding ETH cost per call is also shown. View functions (e.g. `getEvent`, `getMediaItem`) are omitted, as they incur no gas when called off-chain.

Three effects are visible in the data. First, the cost of `createEvent` scales approximately linearly with the number of attached media items ($\approx 150\text{--}170\text{k}$ gas per item), reflecting the on-chain storage of media metadata strings. Second, voting cost depends on storage warmth: the first vote on a given event initialises its counters from zero (cold $0 \rightarrow 1$ storage writes, $\approx 20\text{k}$ gas each), whereas a subsequent vote on the same event updates already-nonzero counters at a reduced cost, lowering the per-vote cost from $\approx 95\text{k}$ to $\approx 61\text{k}$ gas. Third, the `isLocalVoter` flag adds the cost of one additional counter update: when set, the vote increments a separate local-tally counter, and initialising that counter from zero adds a further $\approx 20\text{k}$ gas. This explains why the two `voteOnMedia` measurements differ in the opposite direction to `voteOnEvent`: the first `voteOnMedia` call was cast with `isLocalVoter` disabled (76 190 gas), while the second enabled it, incurring the additional cold local-counter write (81 307 gas).

D. zk-Enabled Voting Cost

The figures in Table II correspond to the non-zk (demonstration) voting mode, in which the voter hash and locality flag are supplied directly. In the zk-enabled mode, a vote is submitted via `voteOnMediaWithProof`, whose cost is the sum of the storage-update cost measured above ($\approx 76\text{k}$ gas) and the cost of on-chain Groth16 proof verification performed by the verifier contract.

The verification cost was measured by deploying the SnarkJS-generated Solidity verifier (via `snarkjs zkkey export solidityverifier`) for each circuit and submitting a real proof produced by the proving pipeline. Both proofs verified successfully on-chain. Table III reports the measured verification cost; the figures correspond to the EVM execution

TABLE III
ON-CHAIN GROTH16 PROOF VERIFICATION COST
(REMIX VM, BN254 CURVE)

Circuit	Public inputs	verifyProof gas
Identity	5	214 787
Location	7	227 978

cost, which is the relevant quantity when `verifyProof` is invoked internally by `voteOnMediaWithProof` rather than as a standalone transaction.

As expected for Groth16 on the BN254 curve, verification is dominated by the three elliptic-curve pairing operations and is largely independent of the circuit’s internal size; the small difference between the two circuits ($\approx 13\text{k}$ gas) reflects the two additional public inputs of the location circuit, each of which adds one scalar multiplication to the linear combination of the verification key ($\approx 6.5\text{k}$ gas per input).

Combining these results, the end-to-end cost of a zk-enabled media vote is approximately $214\,787 + 76\,190 \approx 291\,000$ gas for the identity circuit (and $227\,978 + 76\,190 \approx 304\,000$ gas for the location circuit), of which proof verification accounts for roughly 74–75%. On-chain proof verification, rather than state storage, is therefore the dominant cost in the privacy-preserving voting path. At a reference gas price of 10 gwei this corresponds to approximately 0.0029 ETH per zk-enabled vote.

IX. DISCUSSION AND LIMITATIONS

inMediaVeritas demonstrates that a fully decentralized, privacy-preserving news verification pipeline is technically feasible. However, several limitations merit acknowledgment.

Sparse witness coverage. The system’s correctness depends on a sufficient number of geographically proximate witnesses being present and willing to vote. In low-density areas, rural regions, or for events occurring at unusual times, quorum may be difficult to reach. Mechanisms for incentivizing proactive witness registration in underserved areas remain an open problem.

Authority centralization risk. While multiple authorities can operate in parallel, the credential issuance step reintroduces a degree of centralization at the identity layer. A malicious or captured authority could exclude legitimate reporters or admit Sybil accounts. Federated authority governance and cross-authority credential recognition are mitigations under consideration.

Backend relay for media transport. The current implementation routes IPFS uploads and reads through a backend relay rather than having clients contact the storage node directly. This relay is a deployment convenience and not a trust anchor—it cannot forge content (CIDs are self-verifying) or alter on-chain state—but it remains an availability dependency for media transport. Pointing clients at a public IPFS gateway, or embedding an in-browser node, would remove this dependency without changing the protocol’s trust assumptions.

Legal and regulatory uncertainty. The legal landscape for decentralized media licensing is nascent, and smart contract fee enforcement may conflict with jurisdictional copyright, data protection, or media liability frameworks. Production deployment would require careful legal analysis across target jurisdictions.

Scalability. The current implementation targets a single Ethereum L1 or L2 deployment. High-volume news cycles breaking events with thousands of concurrent submissions would stress both the contract gas costs and the IPFS pinning infrastructure. Layer-2 rollups and sharded IPFS pinning clusters are natural directions for scaling.

X. CONCLUSION

This paper presented *inMediaVeritas*, a decentralized application addressing the misinformation crisis through cryptographic verification, community consensus, and transparent provenance. By combining zk-SNARK location proofs, IPFS media storage, blockchain immutability, credibility-weighted witness voting, and a carefully designed self-sustaining incentive layer, *inMediaVeritas* enables a trustless pipeline from raw media capture to verified, licensable content without relying on any central authority to arbitrate truth.

The identity wallet and authority application demonstrate that the core protocol can be made accessible to non-technical participants, and the news agency interface completes the design across all three tiers of the system. While significant engineering and sociotechnical challenges remain before production deployment, the implemented components—the on-chain voting contract, the zero-knowledge location and identity circuits, and their performance characterization—validate the feasibility of the core verification pipeline and identify a concrete roadmap for the remaining incentive and consensus mechanisms.

REFERENCES

- [1] D. M. J. Lazer *et al.*, “The science of fake news,” *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.
- [2] N. Hassan *et al.*, “ClaimBuster: The first-ever end-to-end fact-checking system,” *Proc. VLDB Endow.*, vol. 10, no. 12, 2017.
- [3] H. R. Hasan and K. Salah, “Combating deepfake videos using blockchain and smart contracts,” *IEEE Access*, vol. 7, 2019.
- [4] E. Ben-Sasson *et al.*, “Zerocash: Decentralized anonymous payments from Bitcoin,” in *Proc. IEEE Symp. Security and Privacy (S&P)*, 2014.
- [5] A. Gabizon *et al.*, “PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge,” *IACR Cryptol. ePrint Arch.*, Rep. 2019/953, 2019.
- [6] M. Bellés-Muñoz, M. Isabel, J. L. Muñoz-Tapia, A. Rubio, and J. Baylina, “Circom: A circuit description language for building zero-knowledge applications,” *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 6, pp. 4733–4751, 2023, doi: 10.1109/TDSC.2022.3232813.
- [7] J. Benet, “IPFS – Content addressed, versioned, P2P file system,” 2014, arXiv:1407.3561.
- [8] Civil Media Company, “The Civil cryptoeconomic whitepaper,” 2018.
- [9] M. Bellés-Muñoz *et al.*, “Twisted Edwards elliptic curves for zero-knowledge circuits,” *Mathematics*, vol. 9, no. 23, 2021.
- [10] Ethereum Foundation, “Remix – Ethereum IDE,” [Online]. Available: <https://remix.ethereum.org>. Accessed: Jun. 8, 2026.
- [11] T. Guo *et al.*, “A survey on automated fact-checking,” *Trans. Assoc. Comput. Linguist.*, vol. 10, pp. 178–206, 2022.
- [12] I. Augenstein *et al.*, “Factuality challenges in the era of large language models,” *arXiv preprint*, arXiv:2310.05189, 2023.
- [13] N. Szabo, “Formalizing and securing relationships on public networks,” *First Monday*, vol. 2, no. 9, 1997.
- [14] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [15] V. Buterin, “Ethereum: A next-generation smart contract and decentralized application platform,” 2014. [Online]. Available: <https://ethereum.org/en/whitepaper/>
- [16] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum Project Yellow Paper, 2014.
- [17] N. Nizamuddin *et al.*, “Decentralized document version control using Ethereum blockchain and IPFS,” *Comput. Electr. Eng.*, vol. 76, pp. 368–384, 2019.
- [18] S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems,” *SIAM J. Comput.*, vol. 18, no. 1, pp. 186–208, 1989.
- [19] J. Groth, “On the size of pairing-based non-interactive arguments,” in *Proc. EUROCRYPT*, 2016, pp. 305–326.
- [20] M. Sporny *et al.*, “Decentralized identifiers (DIDs) v1.0,” W3C Recommendation, Jul. 2022. [Online]. Available: <https://www.w3.org/TR/did-core/>
- [21] M. Sporny *et al.*, “Verifiable credentials data model v1.1,” W3C Recommendation, Mar. 2022. [Online]. Available: <https://www.w3.org/TR/vc-data-model/>
- [22] Protocol Labs, “Filecoin: A decentralized storage network,” 2017. [Online]. Available: <https://filecoin.io/filecoin.pdf>
- [23] S. Williams *et al.*, “Arweave: A protocol for economically sustainable information permanence,” 2019. [Online]. Available: <https://www.arweave.org/yellow-paper.pdf>
- [24] M. Coddington, “Blockchain in journalism,” Tow Center for Digital Journalism, Columbia Journalism Review, 2018. [Online]. Available: https://www.cjr.org/tow_center_reports/blockchain-in-journalism.php