



Blockchain-Enabled Drug Traceability in Healthcare Supply Chains Using Ethereum Smart Contracts and Location-Aware Tracking

Salveru Samay¹, Ganji Manoj², Vemula Saitharun³

^{1,2,3}Department of Computer Science and Engineering, Anurag University, Hyderabad, Telangana, India.

To Cite this Article: Salveru Samay¹, Ganji Manoj², Vemula Saitharun³, "Blockchain-Enabled Drug Traceability in Healthcare Supply Chains Using Ethereum Smart Contracts and Location-Aware Tracking", Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 1021-1028.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by-nc-nd/4.0/); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Counterfeit and inadequately traced pharmaceutical products remain a persistent supply-chain risk because multiple independent organizations handle the same product while keeping separate, centrally administered records. This paper presents a blockchain-enabled drug traceability prototype that maintains a verifiable chain of custody from raw-material handling through manufacturing, distribution, retail sale, and consumer verification. A Solidity smart contract deployed on the Ethereum blockchain restricts stakeholder registration to an administrator, restricts drug creation and ownership transfer to registered participants, and permits transfer only by the account currently recorded as the drug's holder. A Django-based web application layers stakeholder dashboards, an order-generation workflow, and a public product-ID verification portal on top of the contract, with the Google Maps API used to visualize the location recorded at each custody transfer. The system was validated using four documented application-level functional scenarios — administrator authentication, order creation, end-to-end drug tracking, and rejection of an invalid product identifier — all of which produced the expected result, and an end-to-end walkthrough across all five stakeholder roles confirmed that the map-linked traceability report reflects the actual sequence of on-chain transfers. The smart contract itself was further exercised with an automated 14-scenario test suite in a Hardhat Ethereum sandbox, covering authorized and unauthorized registration, drug creation, ownership transfer across every supply-chain stage, and consumer verification; thirteen of the fourteen scenarios produced the expected result, and the remaining scenario empirically confirmed the contract's documented lack of batch-uniqueness enforcement rather than exposing an unexpected defect. A 40-transaction benchmark on the same sandbox additionally measured average gas consumption and confirmation latency for each core contract operation. Rather than reproducing the artificial-intelligence and Internet-of-Things capabilities described in the original project concept without implementation evidence, this paper restricts its claims to the blockchain-tracking, stakeholder-management, and location-visualization functions that the prototype actually demonstrates, and it examines the security properties and residual limitations of the smart-contract design. The results show that a compact, role-restricted smart contract can provide auditable custody transitions and consumer-facing provenance at prototype scale, while highlighting the engineering controls — batch-uniqueness enforcement, role-sequence validation, physical-product binding, and independent security auditing — required before production deployment.

Keywords: Blockchain; Drug Traceability; Ethereum; Smart Contracts; Pharmaceutical Supply Chain; Healthcare.

I. INTRODUCTION

Pharmaceutical traceability is fundamentally a provenance problem rather than a simple inventory problem. A drug package can pass through suppliers, manufacturers, distributors, retailers, and consumers, and every handoff creates an opportunity for records to become inconsistent, incomplete, or deliberately altered. The World Health Organization reports that at least one in ten medicines circulating in low- and middle-income countries is substandard or falsified [1]. Regulatory initiatives increasingly require interoperable electronic tracing at the package level: in the United States, the Drug Supply Chain Security Act requires mechanisms for identifying and tracing certain prescription drugs as they move through the supply chain [2]. Meeting these requirements calls for an architecture in which every custody change is recorded in a way that is difficult to alter after the fact and that can be independently verified by any authorized party, including the end consumer.

Conventional centralized databases can process transactions quickly, but they place strong trust in whichever organization controls the database and in every interface through which data are entered. Because the pharmaceutical supply chain is inherently multi-stakeholder, no single organization is a natural custodian of the shared record. Blockchain-based traceability has therefore been investigated as a way to preserve shared provenance, authorization, and transaction history across organizational boundaries, and smart contracts extend this model by encoding registration, ownership-transfer, and verification rules directly in executable logic rather than in application code that a database administrator could bypass.

This paper develops and evaluates a proof-of-concept drug traceability system built around an Ethereum-compatible smart contract. The original university project scope referenced broader future-facing capabilities involving artificial intelligence, IoT devices, RFID, and predictive analytics. The implementation evidence available in the completed system, however, is strongest

for blockchain-based stakeholder registration, drug ownership transfer, application-level tracking, consumer verification, and map-based location visualization. To avoid overstating the prototype, this paper deliberately limits its claims to these demonstrated functions and treats the remaining capabilities as future work.

The contribution of this paper is fourfold. First, it implements a compact smart-contract model with explicit drug and stakeholder state. Second, it enforces administrator-only stakeholder registration and current-holder validation during transfer. Third, it integrates this blockchain-backed state with a web application that records and displays supply-chain stages and shipment locations for administrators, stakeholders, and consumers. Fourth, it evaluates the implemented workflow through functional test cases and a code-level review of the security properties and limitations of the smart contract, so that the gap between the system's original conceptual scope and what has actually been built and validated is stated explicitly rather than left implicit.

II. RELATED WORK AND RESEARCH GAP

Prior work has established blockchain as a promising foundation for pharmaceutical provenance. Uddin et al. [3] proposed and compared two blockchain architectures for drug traceability, Hyperledger Fabric and Besu, and identified privacy, trust, transparency, security, authorization, and scalability as the key requirements a traceability system must satisfy. Zakari et al. [4] conducted a systematic review of 38 studies on blockchain adoption in the pharmaceutical industry and found that counterfeit-drug prevention, distribution tracking, and safety/security were the dominant application areas, with data governance and prescription monitoring emerging as newer directions. These reviews make clear that the underlying problem is not simply storing a product identifier on a ledger; a practical system must coordinate several independent stakeholders and preserve meaningful, authorized transitions across the full product lifecycle.

Several implemented systems illustrate more elaborate architectures than the one presented here. Jamil et al. [5] developed a Hyperledger Fabric-based medical blockchain for a smart-hospital setting and evaluated its throughput, latency, and resource utilization, giving a quantitative baseline this prototype does not attempt to reproduce. Gomasta et al. [6] presented PharmaChain, a provenance-verification system that, like the present work, models the supply chain as a sequence of custody transfers between defined stakeholder roles, but additionally incorporates hybrid on-chain/off-chain storage. Banerjee [7] and Alzahrani and Bulusu [8] survey the broader landscape of blockchain-based pharmaceutical supply chains and catalogue adoption barriers including scalability, stakeholder onboarding, and integration with legacy enterprise systems, while Kamble and Gunasekaran [9] examine digital-twin architectures as a complementary mechanism for simulating supply-chain state, and Khan and Reddy [10] discuss blockchain-enabled healthcare supply chain management more generally.

More recent work extends this design space along three directions relevant to the present study. Closer in architecture to the present prototype, Musamih et al. [11] implemented an Ethereum smart-contract drug-traceability system with stakeholder roles comparable to those used here, and Valencia-Payan et al. [18] proposed a self-updating smart contract with built-in data validation for general supply-chain traceability. A second direction pairs blockchain with a physical- or cryptographic-binding mechanism this prototype does not implement: Zoughalian et al. [12] apply a zero-knowledge-proof protocol to pharmaceutical distribution monitoring, Chiacchio et al. [13] bind each product to a non-fungible token for track-and-trace, and Mezquita et al. [15] address cross-platform interoperability between pharmaceutical blockchain deployments. A third direction targets consortium or permissioned platforms rather than public Ethereum: Dash et al. [16] use Hyperledger Composer to reduce logistics losses, Abdallah and Nizamuddin [14] survey blockchain adoption across the pharmaceutical industry more broadly, and Konapure and Nawale [17] combine IoT sensing with smart contracts for end-to-end pharma traceability. Table I situates the present system's implemented scope against a representative subset of this literature.

System	Platform	Smart Contract	Consumer Verification	Location Visualization	Access-Control Testing
Uddin et al. [3]	Fabric / Besu	Yes	—	—	—
Jamil et al. [5]	Hyperledger Fabric	Yes	—	—	Throughput/latency
PharmaChain [6]	Hyperledger Fabric	Yes	Yes	—	—
Musamih et al. [11]	Ethereum	Yes	Yes	—	—
Zoughalian et al. [12]	ZKP-based	Partial	—	—	—
Proposed system	Ethereum	Yes	Yes	Yes	14 automated scenarios

Table I. Comparison With Representative Blockchain-Based Pharmaceutical Traceability Systems

As Table I indicates, the present system's distinguishing combination is not any single feature in isolation — consumer verification and multi-stakeholder custody both appear in prior work — but the pairing of map-based location visualization with an openly reported, automated access-control test suite, neither of which the compared systems document at this level of detail.

The present prototype is intentionally narrower than these research platforms. It does not implement decentralized off-

chain storage, cryptographic certificate validation, sensor telemetry, or measured blockchain throughput. Its contribution instead lies in a transparent, fully working implementation of a complete multi-stage ownership workflow, exposed through role-specific interfaces to administrators, stakeholders, and consumers alike, together with an explicit and honest accounting of which parts of the original concept were actually built. This supports a research gap that is easy to overlook in more elaborate systems: how to implement a minimal, auditable, blockchain-backed chain-of-custody model without presenting unimplemented components as completed functionality. Accordingly, this paper treats location information as recorded shipment metadata visualized through the Google Maps API rather than as autonomous GPS tracking, and it treats invalid-product detection as registry-based verification of whether a product identifier maps to an authentic on-chain record rather than as physical or chemical counterfeit detection — blockchain can protect the integrity of a digital record after entry, but it cannot by itself prove that the physical item associated with that record is genuine.

III. SYSTEM DESIGN AND METHODOLOGY

A. Design Objectives

The system was designed around five objectives: (1) restrict the creation of trusted participants to an administrator; (2) create a persistent, uniquely identified record for each drug batch; (3) allow only registered stakeholders to create or transfer a drug record; (4) prevent an account from transferring a drug unless it is the currently recorded holder; and (5) provide an interface through which the complete status history can be inspected by authorized users and queried by consumers. These objectives reflect the minimum controls needed for a credible chain-of-custody prototype, without assuming capabilities — such as sensor-verified authenticity — that were not part of the implemented system.

B. System Architecture

The implemented architecture separates on-chain state from application-level presentation. Users interact with a browser-based application built on Python/Django server logic, which coordinates user authentication, stakeholder management, order creation, tracking updates, consumer queries, database records, blockchain calls, and map rendering. The Ethereum smart contract maintains the authoritative stakeholder and drug state — who is registered, who currently holds a given batch, and whether that batch is marked authentic — while descriptive tracking information such as status text, update dates, and location names is handled by the Django/SQLite application layer. This hybrid design keeps the smart-contract surface small and inspectable while avoiding the cost of writing every piece of user-interface data directly to the chain.

C. Smart-Contract Data Model and Access Control

The Solidity contract defines two primary structures. Drug records a drug's name, batch number, manufacturer, current holder address, and an authenticity flag. Stakeholder records a participant's name, role, and blockchain account. Drugs are indexed by batch number and stakeholders by account address, and the administrator account is fixed at contract deployment. Two modifiers implement access control: `onlyAdmin` restricts stakeholder registration to the administrator, and `onlyRegistered` requires the caller to correspond to a registered stakeholder before any protected supply-chain operation. The `transferDrug` function adds three further checks — the batch must be marked authentic, the caller must equal the batch's current holder, and the recipient address must itself correspond to a registered stakeholder — and the transaction reverts if any check fails, which prevents one registered stakeholder from reassigning custody of a batch it does not currently hold and prevents custody from being handed to an address the administrator has not registered. Table II summarizes the contract's core operations.

Operation	Authorization	Purpose
Register Stake holder	Administrator only (only Admin)	Registers a stakeholder's name, role, and blockchain account address.
Add Drug	Registered stakeholder (only Registered)	Creates a new drug batch record (name, batch number, manufacturer, current holder) and sets its authenticity flag.
Transfer Drug	Current holder, and must be a registered stakeholder	Reassigns the current holder after verifying the drug is authentic, the caller is the recorded holder, and the recipient is a registered stakeholder.
Verify Drug	Public, read-only	Returns the stored authenticity flag for a batch; used by the consumer-facing verification portal.

Table II. Core Smart-Contract Operations

The contract emits an event on every stakeholder registration, drug addition, and drug transfer, which gives transaction-level evidence that the application (or any external blockchain-analysis tool) can consume independently of the application database. The design is intentionally compact, which keeps the authorization logic easy to inspect manually, but it also leaves several controls — role-sequence enforcement, batch-uniqueness checks, and stakeholder revocation — for future work, discussed in Section VI.

D. End-to-End Traceability Workflow

The prototype represents the pharmaceutical supply chain as a sequence of status transitions. An administrator first registers each stakeholder — Raw Material Supplier, Manufacturer, Distributor, and Retailer. A drug order is then created with identifying details, a description, an uploaded product image, and a starting location. As the product moves, each stakeholder updates the tracing status and location before handing off to the next stage, and the sequence concludes with a Sold status recorded once the Retailer completes the sale to the consumer. At the consumer stage, a Product ID can be queried to retrieve the full recorded history; an invalid or unregistered identifier is treated as non-authentic by the verification workflow. This registry-based check is useful for demonstrating that a given identifier corresponds to a genuine on-chain record, but it should not be interpreted as physical authentication of the package itself — a production deployment would need a secure, tamper-evident binding between the digital record and the physical product, for example a serialized and cryptographically protected QR code, NFC tag, or RFID tag.

IV. IMPLEMENTATION

A. Application Layer

The application provides separate workflows for administrators, supply-chain stakeholders, and consumers. The administrative interface supports login, stakeholder registration, viewing registered stakeholders, generating product orders, and inspecting the tracing table for any batch. Stakeholder interfaces allow a participant to update the product's status and location as it is handed to the next stage. The consumer interface accepts a Product ID and returns the recorded movement history together with a per-stage "View on Map" link. The backend is implemented in Python with a Django-style web framework, SQLite for application-level storage, and the Google Maps API for location rendering; the project documentation additionally references React/JavaScript for interactive front-end components. Screenshots captured during system testing show the admin login and dashboard, stakeholder registration, order generation, stage-wise tracing updates from Raw Material Supplier through Manufacturer, Distributor, and Retailer, and the consumer-facing traceability report with location links — confirming that the described workflow executes end-to-end against the deployed contract rather than existing only as a design description.

The application database stores information that is convenient to query and display but that does not need to live on-chain, such as product descriptions, update dates, status text, uploaded images, and location names. This separation is common in blockchain applications, since writing all descriptive data directly to the chain increases both storage requirements and transaction cost. The present prototype does not implement a cryptographic hash linking every off-chain database row to its corresponding on-chain transaction; this is identified in Section VI as a future integrity enhancement rather than a claimed feature.

B. Blockchain Layer

The on-chain component is written in Solidity (pragma ^0.8.0) and deployed to an Ethereum development network. Drug and stakeholder mappings serve as persistent state, and Ethereum account addresses represent participants. The constructor assigns the deploying account as administrator; register Stakeholder writes a Stakeholder record to the account-address mapping; addDrug writes a Drug record to the batch-number mapping; transferDrug updates the currentHolder field only after the authenticity and ownership checks described in Section III-C succeed; and verifyDrug exposes a public, read-only authenticity check consumed by the application's consumer-facing portal. Every write operation returns a transaction receipt containing the transaction hash, block number, cumulative gas used, and contract address, which the application surfaces to the operator as evidence that the operation was committed to the blockchain rather than to an ordinary mutable database table.

This design uses blockchain primarily as a state machine for provenance rather than as a payments or tokenization platform: the contract does not implement cryptocurrency transfers, non-fungible tokens, or decentralized file storage. Gas-consumption and confirmation-latency measurements were not part of the original project's test plan and are not reported here from the original deployment; Section V instead reports gas and latency figures collected separately from a local Hardhat sandbox benchmark of the same contract logic, presented as sandbox-characterized execution cost rather than as figures observed on the original deployment or on a public network.

C. Location-Aware Tracking

The application integrates the Google Maps API to visualize the location associated with each status update. During order generation and each subsequent stage update, a location name is recorded and later rendered on a map; in the demonstrated test run, locations including Mehdipatnam, Ameerpet, Karwan, and Begumpet were associated with successive supply-chain stages for a single tracked order. Because these locations are entered as application data by the stakeholder performing the update rather than acquired automatically from a trusted GPS or IoT sensor, the accurate description of the current implementation is location-aware traceability — a recorded and displayed location history — rather than autonomous real-time location sensing.

V. VALIDATION METHOD

Validation combined application-level functional testing with automated, sandbox-based testing of the smart contract itself. Four application-level functional test cases were documented and executed: administrator authentication using valid credentials; order creation using a drug batch number, supplier identifier, and destination; retrieval of the movement history for a valid Product ID; and the system's handling of an invalid or fake Product ID. For each case, the expected output was compared against the observed output and assigned a pass/fail result, summarized in Table III. To evaluate the contract's access-control and state-transition logic directly, an automated Mocha/Chai test suite (via Hardhat) was additionally written against the Solidity contract compiled with solc 0.8.20 and deployed to a local, in-process Ethereum development network. This suite exercised fourteen

Blockchain-Enabled Drug Traceability in Healthcare Supply Chains Using Ethereum Smart Contracts and Location-Aware Tracking

scenarios spanning authorized and unauthorized calls to every state-changing function, the full supplier-to-retailer transfer chain, consumer-facing verification of valid and invalid batch identifiers, and independent tracking of multiple concurrently held batches; results are summarized in Table IV. The same sandbox was then used to run a 40-transaction benchmark measuring gas consumption and confirmation latency for each core operation, summarized in Table V.

Test	Expected Behaviour	Observed Result	Status
Admin authentication	Valid credentials grant access to the admin panel	Credentials verified; access granted	Pass
Order creation	Order is stored and a tracking ID is assigned	Order stored; tracking ID generated	Pass
Drug tracking	Valid Product ID returns full stage-wise history	Location/status history correctly displayed	Pass
Counterfeit / invalid Product ID	Unregistered ID is rejected and flagged	System flagged the ID as counterfeit and alerted the user	Pass

Table III. Documented Functional Validation

Test	Expected Behaviour	Observed Result	Status
T1 – Admin registers stakeholder	Registration succeeds for an authorized administrator call	Stakeholder registered; isRegistered() returns true	Pass
T2 – Unauthorized registerStakeholder	Reverts under the onlyAdmin modifier	Reverted: “Caller is not the administrator”	Pass
T3 – Registered stakeholder calls addDrug	Drug record is created with the caller as current holder	Drug record created; currentHolder = Supplier	Pass
T4 – Unauthorized addDrug	Reverts under the onlyRegistered modifier	Reverted: “Caller is not a registered stakeholder”	Pass
T5 – Supplier → Manufacturer transfer	currentHolder updates to the Manufacturer account	currentHolder = Manufacturer	Pass
T6 – Manufacturer → Distributor transfer	currentHolder updates to the Distributor account	currentHolder = Distributor	Pass
T7 – Distributor → Retailer transfer	currentHolder updates to the Retailer account	currentHolder = Retailer	Pass
T8 – Unauthorized transferDrug	Reverts under the onlyRegistered modifier	Reverted: “Caller is not a registered stakeholder”	Pass
T9 – Registered non-owner attempts transfer	Reverts under the current-holder check	Reverted: “Caller is not the current holder”	Pass
T10 – Duplicate batch number	Should be rejected if uniqueness were enforced	Record silently overwritten — no uniqueness check exists	Fail – confirms the limitation documented in Section VI-C
T11 – Invalid batch ID verification	verifyDrug() returns false for a nonexistent batch	verifyDrug() returned false	Pass
T12 – Transfer to unregistered recipient	Reverts because the recipient is not a registered stakeholder	Reverted: “Recipient is not a registered stakeholder”	Pass
T13 – Consumer verification of valid Product ID	verifyDrug() returns true for an authentic, existing batch	verifyDrug() returned true	Pass
T14 – Multiple products traced independently	Each batch retains independent, uncorrupted state	BATCH-011 holder = Manufacturer; BATCH-012 holder = Supplier (unaffected)	Pass

Table IV. Automated Smart-Contract Test Suite (Hardhat Sandbox, 14 Scenarios)

To complement the functional pass/fail results, the same contract was exercised through a 40-transaction benchmark on a local Hardhat Ethereum development network (in-process EVM, single node, no network contention), covering 10 repetitions each

of registerStakeholder, addDrug, transferDrug, and verifyDrug. Gas usage was read from each transaction receipt and confirmation time was measured as wall-clock time between transaction submission and receipt. These figures characterize the deployed contract’s execution cost under a local, non-adversarial sandbox and should not be read as mainnet gas-price or public-network latency figures, which depend on prevailing network conditions.

Operation	Avg. Gas Used	Min–Max Gas	Avg. Confirmation Time (ms)	n
Register Stakeholder	96,505	96,495 – 96,507	2.92	10
Add Drug	118,071	118,071 (fixed)	2.46	10
Transfer Drug	34,778	34,778 (fixed)	1.75	10
Verify Drug (read-only)	0 (view call)	—	1.28	10

Table V. Gas Consumption and Confirmation Latency (40-Transaction Sandbox Benchmark)

Beyond these four application-level scenarios, the end-to-end workflow was exercised manually across all five stakeholder roles. Stakeholders were registered by the Admin, a sample order was created and tracked through each stage with the location updated at every transfer, and the consumer-facing verification portal was used to query the resulting Product ID; it correctly returned the full stage-wise history together with the location recorded at each stage. This confirms that the map-linked traceability report reflects the actual sequence of on-chain transfers rather than a static or hard-coded view. The automated sandbox suite (Table IV) closes part of the gap left by manual testing: thirteen of fourteen scenarios, including every unauthorized-caller and non-owner-transfer attempt listed in Table IV, produced the expected reverted-transaction result, directly exercising the onlyAdmin, onlyRegistered, and current-holder checks described in Section III-C rather than only their happy-path outcomes. The fourteenth scenario (T10) is not a passing functional test in the ordinary sense; it is a deliberately adversarial input — a duplicate batch number submitted by a second registered stakeholder — retained specifically to give empirical, reproducible evidence for the batch-uniqueness limitation already stated in Section VI-C, rather than to claim a defect-free contract. Taken together, these tests remain functional rather than load-based: they do not establish reliability under high transaction volume, network failure, or concurrent-access conditions on a public network, and no synthetic accuracy percentage is reported for counterfeit detection, since the prototype performs a registry lookup rather than a statistical or physical classification of the product.

VI. RESULTS AND DISCUSSION

A. Functional Results

All four documented application-level test scenarios produced the expected behaviour. Administrator credentials were accepted, an order was created and assigned a tracking identifier, a valid Product ID returned its recorded movement history, and an invalid Product ID was rejected by the authenticity workflow. At the contract level, the automated 14-scenario sandbox suite (Table IV) produced thirteen expected-pass results across every registration, transfer, and verification path, including all authorized-caller and unauthorized-caller variants, and one deliberately adversarial scenario that reproduces the documented absence of batch-uniqueness enforcement rather than an unexpected failure. The 40-transaction gas/latency benchmark (Table V) shows that a full custody transfer (transferDrug) is the least expensive state-changing operation at an average of 34,778 gas and 1.75 ms confirmation time on the local sandbox, while stakeholder registration and drug creation, which write more state, average 96,505 and 118,071 gas respectively; the read-only verifyDrug call used by the consumer-facing portal incurs no gas cost and averaged 1.28 ms. These results should be read as successful functional and access-control validation of the tested prototype paths on a local, non-adversarial sandbox, not as a general reliability, mainnet-cost, or counterfeit-detection-accuracy claim — the benchmark characterizes execution cost under controlled conditions rather than behaviour under public-network congestion, concurrent access, or adversarial load. The consolidated, stage-by-stage view available to both administrators and consumers addresses a real limitation of fragmented record-keeping: instead of relying on separate records maintained by each participant, a single party can inspect one consistent product history, with the Google Maps view making the recorded locations easier to interpret.

B. Security and Traceability Properties

The smart contract provides four meaningful controls, summarized alongside their residual limitations in Table VI: administrator-only stakeholder registration prevents arbitrary accounts from self-registering; the onlyRegistered modifier requires a known account for any protected operation; the current-holder check on transfer prevents a registered participant from moving a batch it does not hold; and the authenticity check on transfer prevents custody changes to a batch already marked non-authentic.

This analysis also exposes an important distinction between data immutability and data truth. Blockchain can make a committed transaction difficult to alter retrospectively, but the contract has no way to determine whether the original input was accurate. If an authorized stakeholder enters an incorrect location or registers a counterfeit batch as authentic, the blockchain will faithfully preserve that incorrect information — the classic oracle problem in supply-chain applications of blockchain. Future deployments must therefore pair the on-chain integrity guarantees demonstrated here with independent verification mechanisms

— secure serialization, verified device identity, or regulator-issued records — rather than treating blockchain alone as proof of authenticity.

Property	Implemented Control	Residual Limitation
Stakeholder trust	Administrator-only registration via onlyAdmin	No revocation mechanism or multi-admin governance
Transfer authority	Caller must equal the drug's current holder	Role sequence (supplier → manufacturer → ...) is not enforced on-chain
Drug validity	Transfer requires the stored authenticity flag to be true	Authenticity is asserted at registration, not derived from independent physical evidence
Audit evidence	Events emitted on registration, drug creation, and transfer	No formal event-log retention or analysis pipeline
Location visibility	Recorded locations rendered through the Google Maps API	Location is application-supplied text, not authenticated GPS/IoT telemetry

Table VI. Security Properties and Implementation Status

C. Limitations

The prototype has several limitations that would need to be resolved before pharmaceutical deployment. The contract does not currently prevent overwriting an existing batch identifier during addDrug, so explicit uniqueness checking is required; this was confirmed empirically by sandbox test T10 (Table IV), where a second registered stakeholder submitting an existing batch number silently overwrote the original record rather than being rejected. Stakeholder roles are stored as strings, but the contract does not enforce a permitted transition sequence (supplier → manufacturer → distributor → retailer), so a registered stakeholder could in principle receive a batch out of the intended order. There is no stakeholder-suspension or key-recovery procedure, and the authenticity flag is set by the application workflow at registration rather than derived from independent physical evidence. The off-chain database and location metadata are not cryptographically anchored to the on-chain record, and the system has not undergone smart-contract vulnerability scanning, formal verification, penetration testing, or public-testnet deployment; gas and latency were measured only on a local, non-adversarial sandbox (Section V, Table V), not on a public network under realistic load or gas-price conditions. The location component requires similar caution: map visualization confirms that the interface can display a recorded place, but it does not establish continuous or authenticated shipment geolocation. Functional validation now comprises four application-level scenarios plus the fourteen-scenario contract-level sandbox suite described in Section V, which covers invalid credentials at the application layer, duplicate batch creation, unauthorized transfer attempts, and unregistered callers; concurrent-access behaviour, adversarial network conditions, formal verification, and high-volume transaction workloads on a public network remain outside the scope of this validation and are identified here as future work.

VII. CONCLUSION AND FUTURE SCOPE

This paper presented a blockchain-enabled prototype for drug traceability across a healthcare supply chain, combining a Solidity smart contract with a Django web application, stakeholder-specific workflows, consumer product verification, and Google-Maps-based location visualization. The implemented contract stores drug and stakeholder state, restricts participant registration to an administrator, permits protected operations only for registered users, and prevents transfer by an account that is not the recorded current holder. The application demonstrates product movement across raw-material supplier, manufacturer, distributor, retailer, and consumer stages; four documented application-level functional tests passed, an end-to-end walkthrough across all five stakeholder roles confirmed correct, sequential product history and map-based location display, and an automated 14-scenario sandbox suite exercising the contract's access-control logic directly returned thirteen passing results plus one adversarial scenario that confirms a documented limitation. A 40-transaction sandbox benchmark further measured average gas consumption and confirmation latency for every core contract operation, giving a concrete, if preliminary, quantitative baseline where the original evaluation had none.

These results support the feasibility of a compact smart-contract architecture for prototype-level chain-of-custody tracking, but they do not establish production readiness. Future work should strengthen the contract with unique batch enforcement, explicit role-transition rules, stakeholder revocation, secure QR/NFC/RFID linkage between the digital record and the physical product, and formal security testing; off-chain records should be cryptographically anchored to their on-chain counterparts; location collection should move from manually entered text to authenticated IoT or GPS telemetry; and quantitative evaluation should extend beyond the local sandbox gas/latency baseline reported here to measure throughput and behaviour under realistic public-network conditions, concurrent access, and adversarial workloads. Carrying out this future work would move the system from a functional, well-scoped educational prototype toward a more rigorous platform suitable for regulated pharmaceutical traceability.

Acknowledgment

The authors thank Dr. A. Jyothi, Associate Professor, Department of Computer Science and Engineering, Anurag University, for guidance and review during the original project implementation.

REFERENCES

1. World Health Organization, "Substandard and falsified medical products," WHO Fact Sheet, 2024. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/substandard-and-falsified-medical-products>
2. U.S. Food and Drug Administration, "Drug Supply Chain Security Act (DSCSA)," FDA, 2023. [Online]. Available: <https://www.fda.gov/drugs/drug-supply-chain-integrity/drug-supply-chain-security-act-dscsa>
3. M. Uddin, K. Salah, R. Jayaraman, S. Pesic, and S. Ellahham, "Blockchain for drug traceability: Architectures and open challenges," *Health Informatics Journal*, vol. 27, no. 2, 2021, doi: 10.1177/14604582211011228.
4. N. Zakari, M. Al-Razgan, A. Alsaadi, H. Alshareef, H. Al Saigh, L. Alashaikh, M. Alharbi, R. Alomar, and S. Alotaibi, "Blockchain technology in the pharmaceutical industry: A systematic review," *PeerJ Computer Science*, vol. 8, e840, 2022, doi: 10.7717/peerj-cs.840.
5. F. Jamil, L. Hang, K. Kim, and D. Kim, "A novel medical blockchain model for drug supply chain integrity management in a smart hospital," *Electronics*, vol. 8, no. 5, 505, 2019, doi: 10.3390/electronics8050505.
6. S. S. Gomasta, A. Dhali, T. Tahlil, M. M. Anwar, and A. B. M. S. Ali, "PharmaChain: Blockchain-based drug supply chain provenance verification system," *Heliyon*, vol. 9, no. 7, e17957, 2023, doi: 10.1016/j.heliyon.2023.e17957.
7. A. Banerjee, "Blockchain technology in the pharmaceutical supply chain: Implications and challenges," *Pharmaceutical Research*, vol. 36, no. 5, pp. 98–112, 2019.
8. N. Alzahrani and N. Bulusu, "Blockchain-based drug supply chain management: A review and future directions," *Journal of Healthcare Informatics Research*, vol. 5, no. 2, pp. 245–260, 2021.
9. S. S. Kamble and A. Gunasekaran, "A digital twin-based framework for smart pharmaceutical supply chains," *Computers in Industry*, vol. 131, 103498, 2021.
10. M. Khan and K. Reddy, "Blockchain-enabled healthcare supply chain management," *Telematics and Informatics*, vol. 51, 101405, 2020.
11. A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, "A Blockchain-Based Approach for Drug Traceability in Healthcare Supply Chain," *IEEE Access*, vol. 9, pp. 9728–9743, 2021, doi: 10.1109/ACCESS.2021.3049920.
12. K. Zoughalian, J. Marchang, and B. Ghita, "A Blockchain Secured Pharmaceutical Distribution System to Fight Counterfeiting," *International Journal of Environmental Research and Public Health*, vol. 19, no. 7, 4091, 2022, doi: 10.3390/ijerph19074091.
13. F. Chiacchio, D. D'Urso, L. M. Oliveri, A. Spitaleri, C. Spampinato, and D. Giordano, "A Non-Fungible Token Solution for the Track and Trace of Pharmaceutical Supply Chain," *Applied Sciences*, vol. 12, no. 8, 4019, 2022, doi: 10.3390/app12084019.
14. S. Abdallah and N. Nizamuddin, "Blockchain-based solution for pharma supply chain industry," *Computers & Industrial Engineering*, vol. 177, 108997, 2023, doi: 10.1016/j.cie.2023.108997.
15. Y. Mezquita, B. Podgorelec, A. B. Gil-González, and J. M. Corchado, "Blockchain-Based Supply Chain Systems, Interoperability Model in a Pharmaceutical Case Study," *Sensors*, vol. 23, no. 4, 1962, 2023, doi: 10.3390/s23041962.
16. S. Dash, U. Ghugar, D. Godavarthi, and S. N. Mohanty, "HCSRL: Hyperledger Composer system for reducing logistics losses in the pharmaceutical product supply chain using a blockchain-based approach," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-61654-7.
17. R. Konapure and S. Nawale, "Traceability and Verification to Prevent Counterfeit Drugs: A Secure, Efficient Pharma Supply Chain with IoT-Enabled Blockchain and Smart Contracts," *International Journal of Electronics and Communication Engineering*, vol. 12, no. 1, pp. 33–43, 2025, doi: 10.14445/23488549/IJECE-V12I1P103.
18. C. Valencia-Payan, D. Griol, and J. C. Corrales, "Blockchain self-update smart contract for supply chain traceability with data validation," *Logic Journal of the IGPL*, vol. 33, no. 5, jzae047, 2025, doi: 10.1093/jigpal/jzae047.