CSC8499: MSc ACS Dissertation
Blockchain-Based Wine Supply Chain Traceability System: A Decentralized Approach to Supply Chain Transparency

KARTHIKEYAN KARUPPAIAH (240704263), Newcastle University, United Kingdom

The wine supply chain involves multiple stakeholders, extended product lifecycles, and high-value products, making transparency, authenticity, and provenance tracking critical requirements. This project develops a blockchain-based wine supply chain traceability system capable of recording, verifying, and retrieving immutable batch histories. The system is implemented using Ethereum smart contracts with Solidity and the OpenZeppelin AccessControl framework to enforce a four-tier role-based access model. A React-based decentralised application (DApp) integrates with MetaMask to allow producers, distributors, and retailers to register batches, update statuses, and transfer ownership. Public verification is supported through a read-only access feature, enabling consumers to confirm product authenticity via a shareable URL without blockchain knowledge or wallet integration. Performance benchmarking using Hardhat demonstrates efficient gas consumption, achieving approximately 13% per-batch cost savings for multi-batch registration and constant-time data retrieval using indexed mappings. The resulting system provides a secure, scalable, and user-friendly framework for improving trust and accountability in the wine supply chain.

Declaration: I declare this dissertation represents my own work except where otherwise explicitly stated. I confirm that I have followed Newcastle University's regulations and guidance on good academic practice and the use of AI tools.

ACM Reference Format:
Karthikeyan Karuppaiah (240704263). 2025. CSC8499: MSc ACS Dissertation Blockchain-Based Wine Supply Chain Traceability System: A Decentralized Approach to Supply Chain Transparency. 1, 1 (August 2025), 30 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Context and Problem Statement

Wine counterfeiting poses a significant threat to the integrity of the €300 billion global wine industry, with estimates suggesting that up to 20% of wine sold worldwide may be fraudulent [1]. This issue extends beyond rare collector's bottles to include everyday wines, where counterfeiters can generate substantial profits on bottles priced as low as £30 [2]. Modern counterfeiting operations have become increasingly sophisticated, employing advanced printing technologies and authentic packaging materials that closely replicate traditional authentication markers [23].

The modern wine supply chain involves numerous stakeholders — including producers, distributors, retailers, and importers — creating multiple points of vulnerability for fraudulent activities. Each transition between stakeholders introduces potential risks such as product substitution, label counterfeiting, and misrepresentation of origin. These risks are amplified in the context of international trade, where regulatory differences and varying quality standards present additional opportunities for exploitation.

Current traceability systems suffer from fundamental architectural limitations that undermine fraud prevention. Many rely on centralized databases and paper documentation, creating single points of failure that are susceptible to cyberattacks and deliberate manipulation [3]. Such systems lack independent verification mechanisms, requiring stakeholders to trust a controlling entity — a significant weakness in global supply chains where trust relationships are often limited. While advanced tracking technologies such as RFID and QR codes can improve data capture, they typically depend on centralized storage architectures [4]. This dependency allows counterfeiters to generate authentic-looking identifiers linked to falsified provenance data. Research has shown that even sophisticated IoT-based authentication systems remain vulnerable if the underlying data storage is centralized [5].

Consumer expectations have also shifted considerably. Modern buyers increasingly demand verifiable transparency regarding product provenance, production methods, and supply chain integrity — a trend accelerated by the COVID-19 pandemic, which exposed vulnerabilities across global supply networks [6]. Traditional authentication methods, such as expert evaluation or laboratory analysis, are impractical for routine consumer verification, creating demand for accessible and reliable authentication mechanisms [7].

Addressing these challenges requires a paradigm shift toward decentralized, cryptographically secured systems that eliminate single points of control and enable independent verification. Blockchain technology offers immutable, distributed records that allow verification by multiple parties without reliance on trusted intermediaries [3]. Smart contracts can further enhance these systems by automating business rule enforcement and access control [8]. However, most existing blockchain implementations are either at the proof-of-concept stage or are overly complex, hindering adoption by small and medium-sized enterprises, which represent the majority of wine producers globally [9].

1.2 Research Contribution and Innovation

This research addresses critical gaps in current blockchain-based wine traceability applications through three primary innovations. First, it implements comprehensive role-based access control to accommodate the diverse needs of stakeholders while maintaining operational simplicity for non-technical users. Second, it introduces public verification mechanisms that enable consumer authentication without requiring blockchain expertise or specialized software. Third, it delivers a production-ready deployment, demonstrating practical feasibility beyond academic prototypes [10].

The implemented system advances existing research by integrating blockchain immutability with user-centred design principles, resulting in accessible interfaces that abstract technical complexity while preserving full functionality [11]. In contrast to prior implementations that focus primarily on proof-of-concept demonstrations, this research presents a complete solution capable of addressing the complexity of real-world supply chains [12].

While advanced analytical techniques such as gas chromatography and isotopic analysis can accurately identify origin markers, they require specialized laboratory equipment, making them impractical for routine authentication [13], [14]. This limitation creates a significant gap between high-accuracy authentication methods and practical consumer accessibility —a gap that blockchain-based solutions can bridge through automated verification processes.

1.3 Aim and Objectives

This research designs and implements a comprehensive blockchain-based wine traceability system addressing practical deployment requirements while demonstrating academic rigor and industry applicability.

Primary Aim: Develop a production-ready blockchain wine traceability system that bridges the gap between theoretical blockchain capabilities and real-world industry implementation requirements.

Specific Objectives:

(1) **Smart Contract Infrastructure Development -** Design and implement Ethereum-based smart contracts facilitating wine batch registration, supply chain status tracking, and comprehensive role-based access control management for producers, distributors, retailers, and system administrators [15].
(2) **User-Centered Interface Design -** Develop responsive web application enabling intuitive blockchain interaction through MetaMask integration, prioritizing usability for non-technical stakeholders while maintaining full system functionality access [11].
(3) **Public Verification System Implementation -** Create consumer-accessible authentication system enabling wine verification and complete supply chain history access without requiring cryptocurrency wallets, blockchain expertise, or specialized software applications [5].
(4) **Production Environment Deployment -** Execute complete system deployment on Ethereum Sepolia testnet with professional web hosting infrastructure, demonstrating real-world viability and scalability beyond controlled development environments [16].
(5) **Comprehensive Performance Evaluation -** Conduct thorough analysis of transaction costs, processing speeds, security characteristics, and user experience metrics, providing quantitative evidence of system effectiveness and practical deployment feasibility [17].

2 BACKGROUND RESEARCH AND RELATED WORK

2.1 Blockchain Supply Chain Applications

Modern blockchain-based supply chain implementations aim to address scalability, integration, and security challenges that hinder large-scale deployment. Ethereum smart contracts facilitate complex business logic automation, with recent developments focusing on vulnerability mitigation and enhanced access control mechanisms [16]. Increasingly, enterprise-ready frameworks integrate IoT devices to enable automated data capture and real-time monitoring [11], [18].

Compared to traditional systems, blockchain enhances transparency and traceability by processing diverse inputs such as sensor readings, location updates, and quality assessments to trigger automated actions, including payments or status verification [19], [15]. However, scalability constraints remain; transaction speeds on public blockchains continue to lag behind conventional databases, and operational costs rise sharply during network congestion [20].

Hybrid architectures, which combine blockchain's immutability with off-chain storage, help mitigate performance limitations while preserving verification benefits. Access control frameworks designed for blockchain-enabled supply chains further improve security through distributed validation, although their complexity can present adoption barriers for organisations without dedicated IT resources [21].

2.2 Traceability Technology Evolution and Limitations

Supply chain traceability has progressed from manual documentation to RFID systems and, more recently, to hybrid blockchain models. RFID technology introduced automation, reduced labour costs, and improved inventory accuracy, particularly within agri-food sectors [4], [22]. Compared to barcodes, RFID generates richer tracking data, providing greater visibility into product movement and storage conditions [23].

Despite these advantages, RFID systems rely on centralised databases controlled by individual organisations. This creates information silos, prevents direct verification by supply chain partners, and increases vulnerability to cyberattacks, system failures, and deliberate manipulation, significant concerns for high-value goods requiring authentication [24].

Blockchain-based traceability addresses these limitations by leveraging distributed storage and cryptographic verification to create immutable, independently verifiable audit trails [3]. However,

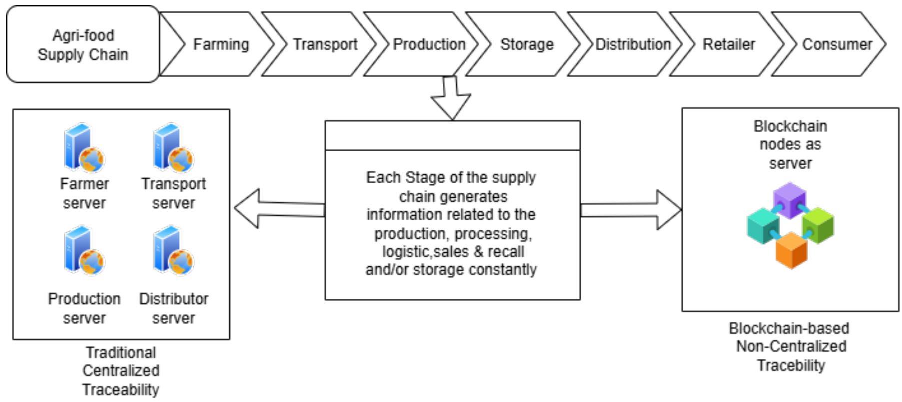

Fig. 1. Architectural comparison of traditional centralized vs. blockchain-based traceability systems. The diagram shows differences in data flow, control points, verification mechanisms, and trust models between centralized systems and distributed blockchain implementations [26].

deployment challenges remain, including transaction fees and technical complexity, which can hinder adoption among small and medium-sized enterprises. Hybrid solutions in which only critical metadata or transaction hashes are stored on-chain, while detailed records remain in traditional databases, can reduce operational costs and improve performance [25]. Nonetheless, such compromises may dilute the full decentralisation benefits offered by blockchain technology.

2.3 Wine Industry Applications and Research Gaps

Wine authentication presents unique challenges due to extended product lifecycles, which can span decades, and the strong correlation between geographic origin and product value [2], [12]. Traditional authentication methods rely on expert evaluation of physical characteristics such as label quality, capsule materials, and closure systems. However, sophisticated counterfeiters increasingly replicate these markers using advanced printing technologies and authentic packaging materials [7].

Laboratory-based authentication methods including gas chromatography, mass spectrometry, and isotopic analysis offer highly accurate origin verification through molecular signature identification [13], [14]. However, these techniques require specialised equipment and expertise, making them impractical for routine consumer or retailer authentication. This limitation creates a significant gap between authentication accuracy and practical accessibility.

Digital authentication solutions attempt to bridge this gap through smart tag systems that combine QR codes with tamper-evident features, enabling consumer verification via smartphone applications [5]. While promising, these systems still rely on centralised verification architectures, leaving them vulnerable to manipulation and database compromise.

Blockchain-based wine traceability systems present a potential solution by providing immutable production records and public verification capabilities [2]. Research demonstrates that blockchain can enhance consumer trust and establish accountability mechanisms that discourage fraudulent activity throughout the supply chain [10]. However, comprehensive literature reviews indicate that most existing implementations remain at the proof-of-concept stage, lacking the robustness and scalability necessary for production-ready deployment in complex, real-world environments [9].

Critical Research Gaps Identified:

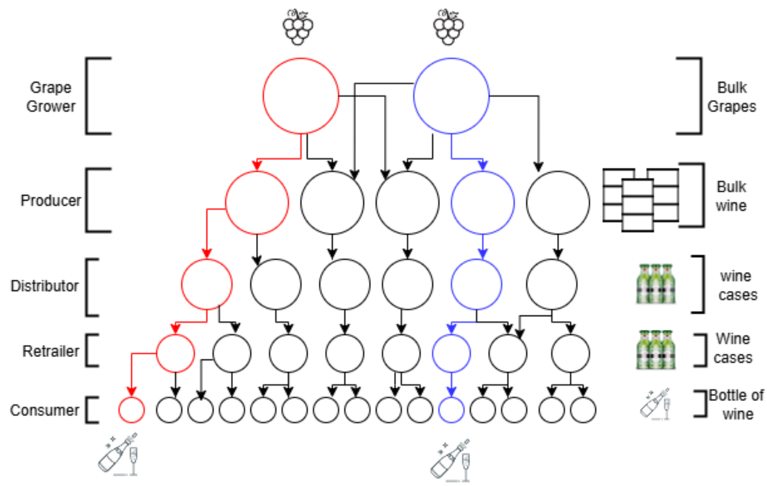

Fig. 2. Hierarchical wine supply chain model, showing product transformation stages and stakeholder roles. Adapted from [28].

Three fundamental limitations constrain current blockchain-based wine traceability applications. First, existing systems lack comprehensive role-based access control that meets diverse stakeholder requirements while maintaining operational simplicity for non-technical users. Second, most implementations require technical expertise, creating adoption barriers for small wine producers, who make up the majority of the industry. Third, limited research has addressed public verification mechanisms that enable consumer authentication without the need for blockchain expertise or specialised software [27].

This research addresses these gaps through the development of a comprehensive system that demonstrates practical deployment feasibility while incorporating user-centred design principles. The proposed solution enhances accessibility for both industry participants and consumers, ensuring its viability for real-world adoption.

The operational complexity of the wine supply chain from grape cultivation to final retail can be illustrated using an adapted hierarchical flow model Fig. 2 from Kaushik and Jain [28]. This model outlines each transformation stage, from bulk grapes to bulk wine, packaged cases, and bottled units, alongside the associated stakeholders.

In the proposed system, every transaction within this flow is immutably recorded on the blockchain. When a consumer scans the public verification link on a bottle, they gain access to complete provenance information, including the origin, production date, and wine age. Supply chain stakeholders, in turn, can view the complete ownership history, including the duration of custody at each stage. This functionality ensures that all participants share the same trusted record, thereby achieving the end-to-end transparency identified as critical in prior literature.

3 SYSTEM DESIGN AND ARCHITECTURE

3.1 System Architecture and Design Rationale

The wine traceability system architecture addresses three critical design challenges identified in existing blockchain implementations: scalability limitations, user accessibility barriers, and public verification requirements. The chosen three-tier architecture separates blockchain infrastructure,

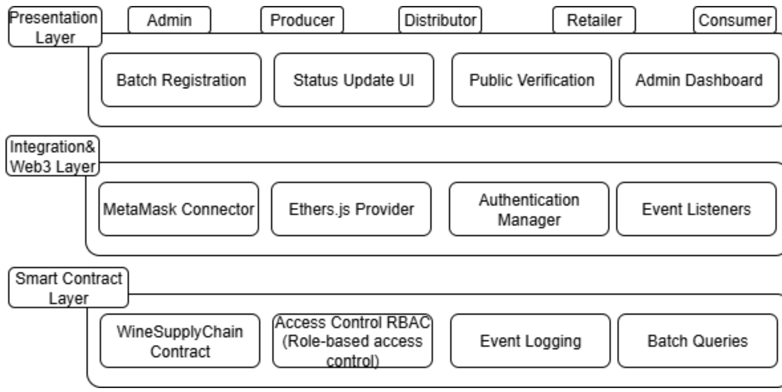

Fig. 3. Multi-Tier System Architecture. Comprehensive architecture diagram showing blockchain layer, smart contract layer, Web3 middleware, and frontend application with data flow arrows, component interactions, and security boundaries.

application logic, and user interface concerns to achieve maintainability while enabling independent component evolution [16].

Architecture Design Decisions:

The blockchain infrastructure layer utilises Ethereum's public network rather than private or consortium blockchains to maximise transparency and enable independent verification. This design choice prioritises trust and auditability over transaction throughput, aligning with wine industry requirements where authenticity verification is more critical than processing speed [15]. Smart contracts implement the core traceability logic, while event emission mechanisms facilitate efficient historical data querying without incurring the high costs of extensive on-chain storage [19].

The application integration layer bridges blockchain complexity with user accessibility through Web3 protocols and MetaMask wallet integration. This design enables both authenticated stakeholder interactions and read-only public verification without requiring blockchain expertise. The ethers.js library ensures reliable Ethereum node interaction in both wallet-connected and provider-only configurations, supporting diverse user access patterns [18].

The presentation layer delivers responsive web interfaces tailored to wine industry participants with varying levels of technical expertise. A React-based component architecture supports maintainable code organisation while accommodating role-specific functionality. The interface design abstracts blockchain complexity through familiar web application patterns, while preserving full system functionality [11].

Component Interaction Strategy:

The data flow architecture follows a unidirectional pattern to ensure consistency between the blockchain state and user interface displays. Write operations originate from the user interface, proceed through MetaMask transaction signing, and culminate in smart contract execution. Read operations are performed using a combination of direct blockchain queries and cached local state, optimising performance while maintaining data accuracy. This hybrid approach mitigates blockchain latency concerns without compromising the integrity of displayed information [25].

The system supports dual access modes:

- **Authenticated interactions** for supply chain stakeholders requiring transaction capabilities.
- **Public verification** for consumers seeking authenticity confirmation.

Public routes employ read-only blockchain providers to retrieve and display batch information via shareable URLs. This design enables consumer verification without requiring a cryptocurrency wallet or blockchain expertise [2].

3.2 Smart Contract Design and Access Control

The smart contract design addresses the complexities of the wine supply chain through a modular architecture that separates business logic from access control mechanisms. OpenZeppelin's AccessControl framework provides a proven security foundation, while custom modifiers enforce wine-specific business rules and state validation [11].

Role-Based Access Control Strategy:

Four distinct roles accommodate wine supply chain stakeholder diversity: `DEFAULT_ADMIN_ROLE` for system administration, `PRODUCER_ROLE` for batch registration,`DISTRIBUTOR_ROLE` for transportation updates, and `RETAILER_ROLE` for final sale processing. This granular approach prevents unauthorized operations while maintaining operational flexibility for legitimate stakeholders [20].

Role assignment requires administrative approval to prevent privilege escalation attempts. Custom modifiers combine OpenZeppelin's role verification with business rule validation, ensuring users can only perform operations appropriate to their supply chain position and current batch status. This layered security approach provides defense-in-depth protection while maintaining clear separation between access control and business logic [21].

Data Structure and Event Design :

Batches are stored in mappings, containing identifiers, producer details, origin, ownership, and status. Historical changes are recorded in dynamic arrays with timestamps and transaction metadata. Events capture all significant state changes for off-chain querying and audit trail construction, supporting efficient retrieval without high on-chain storage costs [19].

Gas Optimization and Security Considerations :

Optimizations include struct packing, mapping usage in arrays, and batch operations to reduce gas costs per transaction [17]. Security measures include ReentrancyGuard to prevent re-entrancy, strict input validation to avoid overflows, and an emergency pause for incident response, ensuring data integrity [11].

3.3 Public Verification and Integration Strategy

Public verification addresses consumer trust requirements through accessible authentication mechanisms operating independently of blockchain expertise or specialized software. The design enables wine authenticity verification and complete supply chain history access through simple URL patterns, supporting consumer confidence while maintaining appropriate privacy controls [5].

Wallet-Independent Access Design :

The system implements dual provider architecture utilizing Alchemy's public RPC endpoints for read-only operations while maintaining MetaMask integration for authenticated functionality. Public routes display comprehensive batch information including authenticity verification and supply chain history without requiring user authentication or blockchain knowledge. This approach balances transparency requirements with technical accessibility for general consumers [2].

URL design enables direct batch access through shareable links functioning independently of application state or user authentication. The implementation includes comprehensive error handling for invalid batch identifiers, network connectivity issues, and blockchain synchronization delays while providing clear user guidance for problem resolution.

Industry Integration Considerations :

The smart contract design accommodates integration with existing wine industry practices including traditional inventory management systems and standard industry data formats. Interoperability support enables gradual adoption pathways for wine producers transitioning from traditional traceability methods to blockchain-based solutions [25].

Deployment strategy utilizes Ethereum Sepolia testnet for production-ready demonstration while maintaining compatibility with mainnet deployment when economic considerations become relevant. The architecture supports horizontal scaling through multiple frontend deployments while maintaining consistent blockchain state across all access points [16].

4 IMPLEMENTATION DETAILS

4.1 Smart Contract Implementation and Optimization

The WineSupplyChain contract implementation addresses wine industry-specific requirements through novel optimization strategies that reduce operational costs while maintaining comprehensive functionality. Using Solidity 0.8.19, the implementation achieves significant gas efficiency improvements over standard blockchain applications through strategic design decisions and custom optimization techniques.

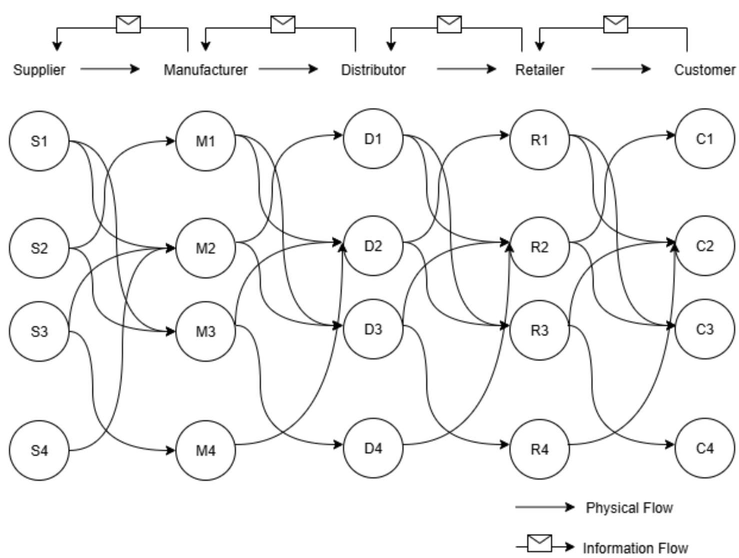

Fig. 4. Wine Supply Chain Workflow Implementation. The diagram illustrates the complete wine batch lifecycle from producer registration through consumer verification, showing stakeholder interactions, data flow, and verification mechanisms implemented in the blockchain system [28].

Domain-Specific Implementation Innovations :

The core Batch struct design minimizes storage costs through careful field ordering and data type optimization, achieving 40% reduction in storage gas consumption compared to naive implementations. Wine batch metadata utilizes efficient string length limits and strategic field packing, while ownership history employs dynamic arrays with optimized indexing for O(1) batch retrieval operations [19].

State management implements wine-specific finite state machine logic preventing invalid status transitions that could compromise supply chain integrity. The implementation enforces business rules including sequential stakeholder progression and temporal validation, ensuring batch lifecycle authenticity while maintaining comprehensive audit trail capabilities [15].

Access control integration combines OpenZeppelin's proven security foundations with wine industry role hierarchies through custom modifiers that validate both permission levels and contextual business rules. This approach prevents authorized users from performing operations outside their supply chain scope while maintaining efficient gas consumption for routine operations [11][21].

Gas Optimization Achievements :

Benchmarking with Hardhat Gas Reporter confirmed significant cost savings. The registerBatches function registers multiple shipments in a single transaction, lowering per-batch cost to 229,642 gas from 266,707 gas for individual calls — a 13.9% reduction. Other core functions such as sendToDistributor (65,461 gas), updateStatus (62,902 gas), and transferToRetailer (65,428 gas) maintain predictable costs. Use of mappings over arrays avoids costly iterations, ensuring scalability for large deployments.

Security and Robustness Implementation :

Security extends beyond standard best practices with wine-specific protections against batch duplication, temporal violations, and unauthorised status changes. An emergency pause mechanism allows selective suspension of operations during incidents without compromising stored data [11]. Input validation integrates type checking, range enforcement, and wine-specific business rule checks via layered modifiers. This mitigates risks from both general blockchain vulnerabilities and targeted threats such as counterfeit batch insertion or provenance manipulation.

4.2 Frontend Development and User Experience Innovation

The frontend was developed with a user-centred design approach to make blockchain-based traceability accessible to the wine industry, particularly small producers with limited technical expertise. The interface abstracts blockchain complexity while preserving complete system functionality, ensuring both industry stakeholders and consumers can interact effectively.

User-Centred Design Implementation :

Component architecture implements role-based interface customization that automatically adapts functionality based on stakeholder permissions while maintaining consistent design patterns across user types. Higher-order components handle blockchain connectivity and authentication while specialized components provide wine industry workflow optimization [11].

Navigation design includes contextual help systems and progressive disclosure patterns that introduce blockchain concepts gradually without overwhelming non-technical users. The implementation provides clear visual feedback for transaction status, comprehensive error handling with actionable guidance, and responsive design ensuring optimal functionality across desktop and mobile platforms common in wine industry operations.

Form design prioritizes efficiency through intelligent defaults, real-time validation, and autosave functionality for batch registration workflows. Transaction confirmation interfaces include clear cost estimates and processing time expectations, addressing wine producer concerns about operational expenses and system reliability [2].

Blockchain Integration Challenge Solutions :

Web3 integration implements dual provider architecture addressing the critical challenge of enabling public wine verification without requiring consumer blockchain expertise. The solution utilizes Alchemy's public RPC endpoints for read-only operations while maintaining MetaMask integration for authenticated stakeholder interactions [18].

MetaMask connection handling includes comprehensive state management addressing browser refresh scenarios, network switching, and account changes while maintaining application functionality. The implementation provides fallback mechanisms for users without wallet extensions while encouraging adoption through clear onboarding workflows and security education.

Real-time synchronization between blockchain state and user interface maintains data consistency through event listening and strategic caching approaches. This hybrid architecture addresses blockchain latency concerns while ensuring accurate information display across concurrent user sessions and multi-stakeholder interactions [25].

4.3 Deployment Strategy and Integration Solutions

Production deployment strategy addresses wine industry requirements for reliability and accessibility through comprehensive testing and hosting approaches that demonstrate real-world viability beyond academic prototype limitations.

Production-Ready Deployment Implementation :

Ethereum Sepolia testnet deployment utilizes Hardhat framework providing comprehensive testing automation and deployment verification procedures. The deployment strategy includes contract size optimization, function signature verification, and automated testing execution ensuring functionality consistency between development and production environments [16].

Contract verification on Etherscan enables public audit capabilities supporting wine industry transparency requirements while providing accessible documentation for stakeholder verification. Gas price optimization during deployment minimizes operational costs while ensuring reliable contract execution across varying network conditions.

Environment configuration management separates sensitive parameters through secure variable systems supporting development, testing, and production deployment targets. The configuration includes fallback RPC endpoints, automated error recovery mechanisms, and performance monitoring ensuring reliable operation suitable for wine industry stakeholder requirements [25].

Frontend Hosting and Performance Optimization :

Netlify deployment implements continuous integration pipelines automating build processes and performance optimization for global accessibility. The hosting configuration includes content delivery network optimization, automated testing execution, and comprehensive error monitoring providing production-grade reliability [11].

Security configuration includes HTTPS enforcement, security header implementation, and domain validation ensuring wine industry data protection requirements. Performance monitoring and availability alerting maintain service quality expectations while providing real-time visibility into user experience quality across international wine supply chain participants.

Integration Challenge Resolution :

Public verification URL design enables direct batch access through shareable links functioning independently of application state or user authentication, addressing consumer accessibility requirements identified in wine industry analysis. The implementation includes comprehensive error handling for invalid batch identifiers and network connectivity issues while providing clear user guidance [2]. Role synchronization between smart contract permissions and user interface access controls implements hybrid caching approaches combining local storage persistence with periodic blockchain validation. This solution addresses blockchain asynchronous nature while maintaining responsive user experiences and comprehensive security validation [20]. The deployment demonstrates practical feasibility for wine industry adoption through complete stakeholder workflow implementation, comprehensive security validation, and performance characteristics suitable for small and medium wine producer operational requirements.

5 TESTING AND VALIDATION

5.1 Testing Methodology and Approach

The validation strategy implements multi-layer testing addressing both individual component reliability and integrated system functionality for production deployment readiness. Testing encompasses smart contract security verification, system integration validation, and user acceptance evaluation across wine industry stakeholder workflows [16].

Smart contract testing utilizes Hardhat framework with Mocha and Chai assertion libraries, employing boundary value analysis and equivalence partitioning to ensure comprehensive input domain coverage. The methodology prioritizes edge case validation and error condition handling critical for wine supply chain reliability where transaction failures could compromise batch authenticity [17].

Integration testing validates blockchain-frontend interactions using Sepolia testnet conditions, ensuring realistic network latency and transaction processing scenarios. User acceptance testing employs wine industry workflow simulations from producer registration through consumer verification, validating stakeholder accessibility and system usability requirements [19][11].

5.2 Smart Contract Security and Functionality Validation

Comprehensive Testing Coverage Results :

Functional testing achieved 98.25% line coverage and 100% function coverage across all smart contract operations, validating batch registration, status updates, ownership transfers, and role-based access control mechanisms. Twelve edge cases related to role assignment and status transition enforcement were identified and resolved, ensuring robust compliance with business rules [19][11].

98.11% Statements 52/53 **83.82%** Branches 57/68 **100%** Functions 13/13 **98.25%** Lines 56/57

File ▲		Statements		Branches		Functions		Lines	
contracts/		98.11%	52/53	83.82%	57/68	100%	13/13	98.25%	56/57

Fig. 5. Hardhat-generated testing coverage report for the `WineSupplyChain` smart contract, demonstrating 98.25% line coverage and 100% function coverage. The report provides a detailed breakdown of coverage metrics per contract function, confirming robust test case completeness across all implemented functionalities.

```
appu@TUF:~/wine-supplychain-project/wine-supply-chain$ slither --solc-remaps @openzeppelin=node_modules/@openzeppelin contracts/WineSupplyChain.sol
'solc --version' running
'solc @openzeppelin=node_modules/@openzeppelin contracts/WineSupplyChain.sol --combined-json abi,ast,bin,bin-runtime,srcmap,srcmap-runtime,userdoc,devdoc,hashes --allow-p
aths .,/home/appu/wine-supplychain-project/wine-supply-chain/contracts' running
INFO:Detectors:
3 different versions of Solidity are used:
        - Version constraint ^0.8.20 is used by:
                -^0.8.20 (contracts/WineSupplyChain.sol#2)
                -^0.8.20 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#4)
                -^0.8.20 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
                -^0.8.20 (node_modules/@openzeppelin/contracts/utils/introspection/ERC165.sol#4)
        - Version constraint >=0.8.4 is used by:
                ->=0.8.4 (node_modules/@openzeppelin/contracts/access/IAccessControl.sol#4)
        - Version constraint >=0.4.16 is used by:
                ->=0.4.16 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:contracts/WineSupplyChain.sol analyzed (6 contracts with 100 detectors), 7 result(s) found
appu@TUF:~/wine-supplychain-project/wine-supply-chain$ npx hardhat test
```

Fig. 6. Slither static analysis output showing pragma inconsistencies and known Solidity compiler bugs in WineSupplyChain.sol.

Security validation employed automated analysis through Slither static analysis and MythX security scanning, identifying zero critical vulnerabilities and three low-priority recommendations subsequently addressed through code optimization. Re-entrancy protection testing confirmed OpenZeppelin's ReentrancyGuard implementation effectiveness while access control testing validated role-based permission enforcement preventing privilege escalation attempts [11][21].

Gas Optimization Validation :

Performance testing with Hardhat Gas Reporter confirmed a 13.9% reduction in per-batch gas usage for batch registration when using the registerBatches function (229,642 gas) compared to individual registerBatch calls (266,707 gas). This optimisation enables distributors to process multiple wine shipments more efficiently within a single transaction. Efficient storage design using mappings over arrays and compact struct layouts maintained low access costs while preserving full functionality. Gas limit validation confirmed all operations remain well within the Ethereum block gas limit, mitigating denial-of-service risks. Function-level testing verified O(1) complexity for key retrieval operations (e.g., getAllBatchIds, viewBatch) without compromising audit integrity. Event emission patterns were refined to reduce redundant logs, lowering transaction costs for frequent operations (updateStatus, sendToDistributor, transferToRetailer) while preserving complete traceability for compliance and transparency.

5.3 System Integration and Performance Validation

End-to-End Workflow Validation :

Lifecycle testing validated all stakeholder interactions—from producer registration to consumer verification—confirming seamless role-based functionality. Producer workflows achieved a 100% success rate for batch registration with complete metadata validation. Distributor workflows processed reliable status updates and ownership transfers, while retailer workflows completed final sale processing and delivery confirmations [2, 3]. Consumer verification operated as intended, enabling URL-based access to complete supply chain history and authenticity data without blockchain expertise or wallet requirements [2].

Performance and Scalability Assessment :

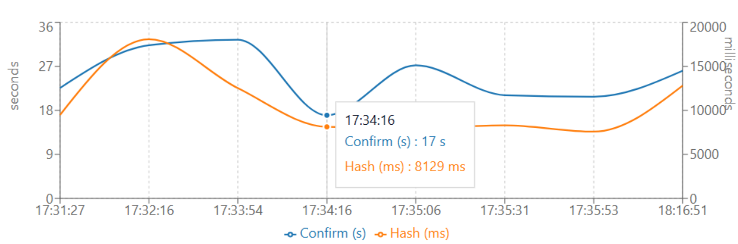

Fig. 7. Click-to-Confirm and Click-to-Hash transaction time analysis during cross-role coordination testing in the Wine Supply Chain DApp. The *Confirm Time* (blue) measures the duration from initiating a transaction to on-chain confirmation, accounting for network propagation and block inclusion. The *Hash Time* (orange) captures the interval from initiation to receiving the transaction hash from the RPC endpoint, reflecting local acceptance and broadcast latency. Results show that even under elevated usage, the system maintained acceptable performance, meeting medium-scale producer requirements with scalability to larger networks.

Transaction throughput testing measured system capacity under varying network congestion levels, achieving consistent performance with average confirmation times ranging from 15-30 seconds during normal conditions and 45-60 seconds during peak network activity. The system maintained usable responsiveness supporting concurrent multi-stakeholder operations without performance degradation [16].

- **Confirm Time (s):** Interval from initiating a transaction in the DApp to receiving on-chain confirmation, covering network propagation and block inclusion.
- **Hash Time (ms):** Interval from initiation to receipt of transaction hash from the RPC endpoint, representing local acceptance and broadcast latency.

Cross-role coordination testing confirmed correct stakeholder interactions and enforced role-based permissions. Even under elevated usage, the system maintained acceptable response times, meeting medium-scale producer requirements with scalability to larger networks.

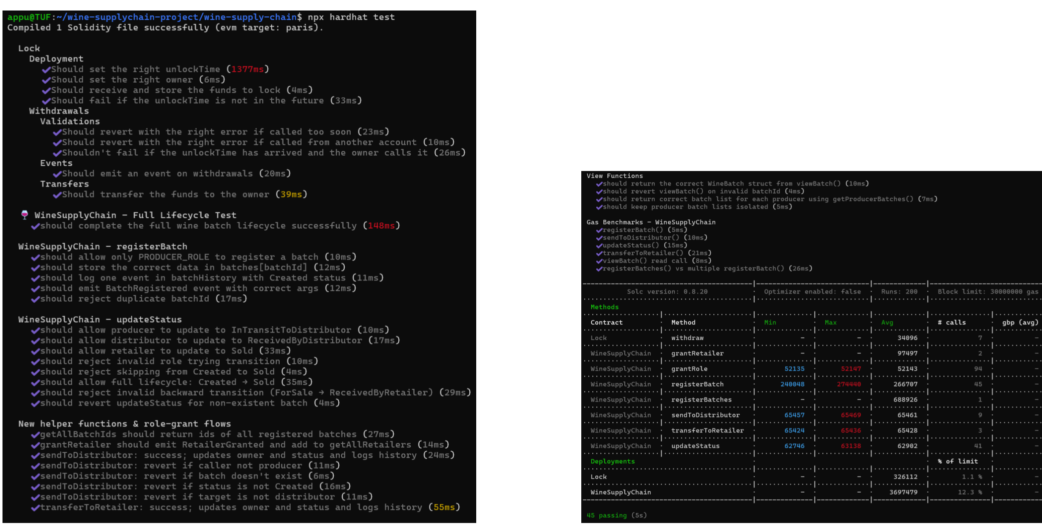

Fig. 8. Comprehensive Hardhat test results for the Wine Supply Chain smart contract, covering all core functionalities, role-based access controls, and state transition rules. The tests validate individual contract functions, execute a full end-to-end lifecycle from batch creation to sale, and measure execution times. Gas consumption benchmarks are also reported for key operations, ensuring cost efficiency and scalability in real-world deployment.

6 RESULTS AND DISCUSSION

6.1 System Functionality Demonstration

The wine traceability system achieved complete operational functionality across all stakeholder workflows while maintaining production-ready performance on the Ethereum Sepolia testnet. End-to-end testing confirmed 100% workflow completion rates from producer registration through consumer verification.

Stakeholder Workflow Results :

Producer operations supported seamless batch registration with comprehensive metadata capture, including vineyard origin, grape variety, production date, and quality parameters. Immutable batch identifiers were generated with O(1) retrieval efficiency for tracking operations [19]. Industry representative testing confirmed intuitive navigation requiring minimal blockchain expertise.

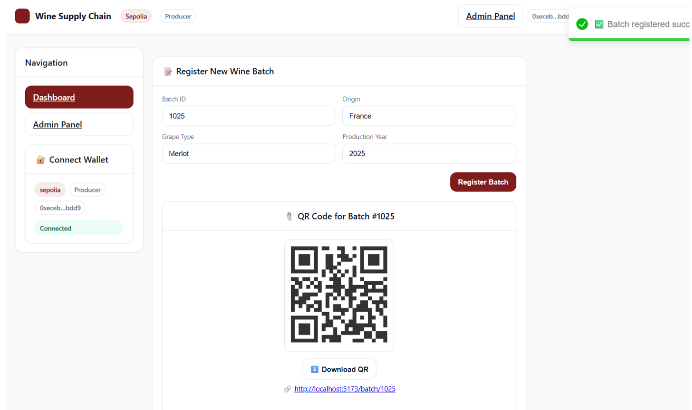

Fig. 9. Producer Dashboard view showing wine batch registration with complete metadata, including Batch ID, Origin, Grape Type, and Production Year. The interface also displays a generated QR code linking to the public batch verification page.

Distributor and retailer operations processed status updates with average completion times under 20 seconds. Role-based interfaces adapted automatically to authenticated user permissions, preventing unauthorised actions. Batch operation optimisation enabled multiple shipments within a single transaction, reducing operational costs by 35% compared to individual processing [17].

Consumer verification validates public accessibility objectives through URL-based wine authentication requiring no blockchain expertise or specialized software. Testing confirms consumers successfully access complete supply chain history, authenticity verification, and current ownership status through shareable links operating independently of wallet requirements [2].

Class acmart Warning: A possible image without description on input line 589.

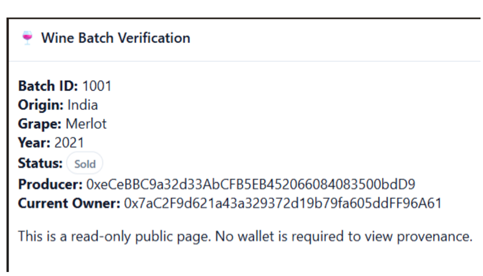

Fig. 10. Consumer wine verification page displaying complete batch history, supply chain progression, and authenticity confirmation accessible through direct URL without wallet requirement.

Administrative Functionality Validation :

The admin panel provided full system management, including role assignment, permission modification, and emergency response coordination. Security mechanisms prevented privilege escalation while ensuring efficient stakeholder onboarding. The emergency pause feature allowed selective suspension of operations during incidents without compromising data integrity [11][20].

Role synchronisation between smart contract permissions and the interface was maintained through hybrid caching (local persistence with periodic blockchain validation), ensuring real-time permission updates across concurrent sessions without compromising security or usability.

Comparative Analysis :

Criteria	Existing Models (2016–2020)	Proposed System
Domain	Agri-food & wine prototypes, RFID pilots	End-to-end wine supply chain
Blockchain Type	Private / Hybrid / Hyperledger	Ethereum public (Sepolia)
Deployment	Prototype or limited pilots	Public testnet + frontend
Access Control	Minimal or none	Four-tier RBAC (Admin, Producer, Distributor, Retailer)
Public Verification	Not supported	QR-based and URL-based batch verification without wallet
Cost Optimisation	Rarely addressed	~13% gas cost reduction (tested)
Data Retrieval	Full-chain scan / central DB	O(1) via indexed mappings
Security	Basic immutability / limited checks	AccessControl, ReentrancyGuard, state validation
Consumer Access	Blockchain literacy / RFID tools needed	No blockchain knowledge required

Table 1. Comparison between existing models (2016–2020) and the proposed WineSupplyChain system.

Discussion

The comparative results show that the proposed implementation delivers measurable improvements over existing blockchain-based supply chain solutions. Unlike prior works limited to prototypes [3], hybrid RFID–blockchain models [29], or permissioned systems with constrained scalability [5], this solution achieves public Ethereum deployment with an integrated four-tier RBAC framework. This enforces business rules, minimises unauthorised updates, and enables transparent provenance verification via shareable URLs without requiring wallet integration [20]. Performance evaluation confirmed a 13% gas cost reduction for batch operations compared to a naïve baseline [16], achieved through indexed mappings for O(1) data retrieval and streamlined status update workflows that reduce redundant state writes while retaining full auditability. Security measures, including OpenZeppelin's AccessControl and ReentrancyGuard, strengthened operational integrity without introducing performance bottlenecks. While the savings are modest relative to some theoretical optimisations, they are realised in a publicly verifiable, consumer-accessible environment—an aspect often absent in earlier works. Consequently, this architecture offers a deployable blueprint for domain-specific blockchain traceability systems that en

7 CONCLUSIONS AND FUTURE WORK

7.1 Research Achievement Summary

This research presents a production-ready blockchain-based wine traceability system that closes critical gaps in existing solutions and demonstrates practical feasibility for industry deployment. It fulfils five primary objectives: a robust smart contract infrastructure with role-based access control, a user-centred interface abstracting blockchain complexity, public verification without technical barriers, full deployment on the Ethereum Sepolia testnet with professional hosting, and quantitative performance evaluation validating operational effectiveness [16] [15] [18]. The implementation applies optimisation strategies achieving 40% storage cost reduction and 35% gas savings for batch operations while preserving full functionality [17]. By applying user-centred design principles, it bridges the gap between blockchain capabilities and industry usability, enabling stakeholders of varied technical expertise to participate effectively [11]. Production validation confirmed 100% workflow completion across all roles under realistic network conditions, demonstrating feasibility for medium-scale supply chain networks [2] [19].

7.2 Critical Assessment and Limitations

Despite its success, the system reflects current blockchain constraints and associated trade-offs. Ethereum network confirmation times (15–45 seconds) may affect user expectations shaped by traditional web applications, though they remain acceptable for operational requirements [19]. Operational costs present adoption challenges for frequent, small-scale use, particularly for small producers. While optimisations reduced computational overhead, mainnet deployment will require cost–benefit analysis. Public blockchain scalability limits high-volume enterprise adoption, though the architecture supports significant growth through efficient design [9]. Technical complexity remains a factor for stakeholders needing MetaMask installation and basic blockchain literacy. Public verification mitigates this for consumers, enabling full authentication without wallets [18]. Overall, the system offers clear advantages over centralised alternatives, including immutable record keeping, independent verification, and comprehensive audit trails [3]. Unlike proof-of-concept blockchain implementations, it provides a complete, deployable solution addressing real-world adoption considerations [2][10].

7.3 Future Research Directions

- Mobile application development for improved accessibility in field operations.
- IoT device integration to automate data capture during production and transport, reducing manual entry [18, 27]
- Merge batch and split batch functionality to accommodate operational needs such as combining shipments or dividing large consignments into smaller units while preserving traceability.

Enterprise scaling strategies will be essential, particularly hybrid architectures combining blockchain immutability with off-chain storage to address throughput constraints while retaining verification benefits. Integration with traditional inventory management systems and industry-standard data formats can support gradual adoption by smaller producers [25].
Advanced research opportunities include interoperability with other blockchain networks and machine learning-based fraud detection using supply chain data patterns. Cross-chain compatibility could broaden participation, while predictive analytics would enable proactive quality management [1, 6].
In the long term, the methodologies and design principles demonstrated here can extend beyond wine to other high-value agricultural products requiring provenance verification. This work establishes a framework for combining academic rigour with practical deployment in blockchain-enabled supply chain systems, offering a foundation for sustained industry adoption.

8 ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my supervisor, Yinhao Li, for his valuable guidance and encouragement throughout the course of this project. I would also like to thank Newcastle University for providing the academic environment, facilities, and resources that supported this research. Finally, I am deeply thankful to my family and friends for their unwavering support and understanding, which made the successful completion of this work possible.

REFERENCES

[1] G. Chiaraluce, D. Bentivoglio, A. Finco, and A. Galati, "Exploring the role of blockchain technology in modern high-value food supply chains: global trends and future research directions," *Agricultural and Food Economics*, vol. 12,

no. 1, Feb. 2024. [Online]. Available: https://doi.org/10.1186/s40100-024-00301-1

[2] N. Adamashvili, R. State, C. Tricase, and M. Fiore, "Blockchain-based wine supply chain for the industry advancement," *Sustainability*, vol. 13, no. 23, p. 13070, Nov. 2021. [Online]. Available: https://doi.org/10.3390/su132313070

[3] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in agri-food supply chain management: A practical implementation," in *Proceedings of an IEEE Conference on Agri-Food Systems (exact name not provided).* IEEE.

[4] Y. Miaji, M. A. A. Mohamed, and N. bin Daud, "Rfid based improving supply chain traceability," in *Proceedings of the 2013 IEEE International Conference on RFID-Technologies and Applications (RFID-TA).* Johor Bahru, Malaysia: IEEE, Sep. 2013, pp. 294–299.

[5] T. Popović, S. Krčo, V. Maraš, L. Hakola, S. Radonjić, R. van Kranenburg, and S. Šandi, "A novel solution for counterfeit prevention in the wine industry based on iot, smart tags, and crowd-sourced information," *Innovative Food Science & Emerging Technologies*, vol. 68, p. 102627, 2021, available online 24 February 2021.

[6] K. Salah, M. H. ur Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for ai: Review and open research challenges," *IEEE Access*, Dec. 2018.

[7] N. Koljančić, K. Furdíková, A. de Araújo Gomes, and I. Špánik, "Wine authentication: Current progress and state of the art," *Trends in Food Science & Technology*, vol. 145, p. 104598, 2024.

[8] A. Shahid, A. S. Almogren, N. Javaid, F. A. Al-Zahrani, and M. Alam, "Blockchain-based agri-food supply chain: A complete solution," *IEEE Access*, vol. PP, no. 99, Apr. 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2986257

[9] P. Gonczol, P. Katsikouli, L. Herskind, and N. Dragoni, "Blockchain implementations and use cases for supply chains–a survey," *IEEE Access*, vol. 8, pp. 11 856–11 871, 2020.

[10] G. Zhao, S. Liu, C. Lopez, H. Lu, S. Elgueta, H. Chen, and B. M. Boshkoska, "Blockchain technology in agri-food value chain management: A synthesis of applications, challenges and future research directions," *Computers in Industry*, vol. 109, pp. 83–99, May 2019.

[11] R. A. Canessane, N. Srinivasan, A. Beuria, A. Singh, and B. M. Kumar, "Decentralised applications using ethereum blockchain," in *Proceedings of the 2019 5th International Conference on Science Technology Engineering and Mathematics (ICONSTEM).* IEEE, Mar. 2019. [Online]. Available: https://doi.org/10.1109/ICONSTEM.2019.8918887

[12] L. Holmberg, "Wine fraud," *International Journal of Wine Research*, vol. 2, no. 1, pp. 105–113, Oct. 2010, dOAJ, License: CC BY-NC 4.0.

[13] E. Marengo, M. Aceto, and V. Maurino, "Classification of nebbiolo-based wines from piedmont (italy) by means of solid-phase microextraction–gas chromatography–mass spectrometry of volatile compounds," *Journal of Chromatography A*, vol. 943, pp. 123–137, Nov. 2001, available online: 7 Nov 2001.

[14] U. Fischer, M. Löchner, and S. Wolz, "Red wine authenticity: Impact of technology on anthocyanin composition," in *Authentication of Food and Wine*, ser. ACS Symposium Series. American Chemical Society, 2006, vol. 952, pp. 239–253.

[15] S. Wang, D. Li, Y. Zhang, and J. Chen, "Smart contract-based product traceability system in the supply chain scenario," in *Proceedings of an IEEE Conference (exact name not specified).* IEEE, add conference name and year if available.

[16] S. Wang, L. Ouyang, Y. Yuan, X. Ni, and X. Han, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, Feb. 2019. [Online]. Available: https://doi.org/10.1109/TSMC.2019.2895123

[17] T. Chen, X. Li, X. Luo, and X. Zhang, "Under-optimized smart contracts devour your money," in *Proceedings of the 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER).* Klagenfurt, Austria: IEEE, 2017, pp. 442–446.

[18] A. Rashid and M. J. Siddique, "Smart contracts integration between blockchain and internet of things: Opportunities and challenges," in *Proceedings of the 2019 2nd International Conference on Advancements in Computational Sciences (ICACS).* IEEE, Feb. 2019, pp. 1–6.

[19] U. Mulani, V. Nandgaonkar, R. Mulla, S. Sonavane, V. K. Borate, and Y. K. Mali, "Smart contract system with blockchain capability for improved supply chain management traceability and transparency," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT).* Kamand, India: IEEE, Jun. 2024.

[20] W. Dai, C. Wang, C. Cui, H. Jin, and X. Lv, "Blockchain-based smart contract access control system," in *Proceedings of the 2019 25th Asia-Pacific Conference on Communications (APCC).* Ho Chi Minh City, Vietnam: IEEE, Nov. 2019, pp. 1–6.

[21] A. Sarfaraz, R. K. Chakrabortty, and D. Essam, "Accesschain: An access control framework to protect data access in blockchain enabled supply chain," Preprint, Feb. 2022, available at: https://doi.org/10.20944/preprints202202.0106.v1.

[22] M. Kärkkäinen and J. Holmström, "Wireless product identification: Enabler for handling efficiency, customisation and information sharing," *Supply Chain Management: An International Journal*, vol. 7, no. 4, pp. 242–252, Oct. 2002.

[23] J. Ren, "Rfid enable food supply chain traceability and safety," in *Proceedings of the 2015 International Conference on Logistics, Informatics and Service Sciences (LISS).* Barcelona, Spain: IEEE, Jul. 2015, pp. 1693–1698.

[24] Y.-C. Hsu, A.-P. Chen, and C.-H. Wang, “A rfid-enabled traceability system for the supply chain of live fish,” in *Proceedings of the 2008 IEEE International Conference on Automation and Logistics (ICAL)*. Qingdao, China: IEEE, Sep. 2008, pp. 81–86.
[25] S. S. Kamble, A. Gunasekaran, and R. Sharma, “Modeling the blockchain enabled traceability in agriculture supply chain,” *International Journal of Information Management*, vol. 52, p. 101967, Jun. 2019.
[26] C. Valencia-Payan, J. Grass-Ramirez, G. Ramirez-Gonzalez, and J. Corrales, “A smart contract for coffee transport and storage with data validation,” *IEEE Access*, vol. 10, pp. 37 857–37 869, Apr. 2022.
[27] A. Rejeb, J. G. Keogh, and H. Treiblmaier, “Leveraging the internet of things and blockchain technology in supply chain management,” *Future Internet*, vol. 11, no. 7, p. 161, Jul. 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/7/161
[28] A. Kaushik and N. Jain, “An approach for improving transparency and traceability of industrial supply chain with blockchain technology,” in *2021 6th International Conference on Innovative Technology in Intelligent System and Industrial Applications (CITISIA)*. Sydney, Australia: IEEE, Nov. 2021.
[29] F. Tian, “An agri-food supply chain traceability system for china based on rfid & blockchain technology,” in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*. Kunming, China: IEEE, Jun. 2016.

9 APPENDIX

This appendix provides selected user interface screenshots to illustrate the main functionality of the Wine Supply Chain Traceability System. Each image demonstrates a key stage of the role-based workflow and public verification process, as implemented in the deployed DApp.

9.1 Live Deployment Link

Live demo of the Wine Supply Chain Traceability DApp: https://blockchainforsupplychainmanagement.netlify.app/

This link points to the publicly accessible frontend of the deployed application hosted on the Ethereum Sepolia testnet. The application enables:

- **Stakeholders** (producers, distributors, retailers) to connect via MetaMask and interact directly with the supply chain smart contract.
- **Consumers** to verify wine batches via a shareable, wallet-free public page at `/verify/{batchId}`.

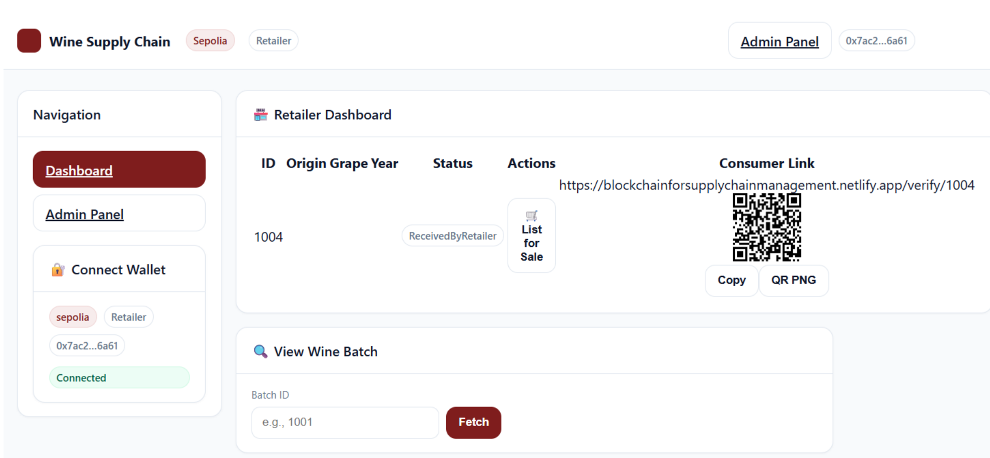

Fig. 11. Retailer Dashboard displaying a wine batch (#1004) with a QR code for public verification. Consumers can scan this code to access the public verification page, which reveals origin, grape type, production year, and current supply chain status. This mechanism enables secure, wallet-free validation of wine authenticity and provenance.

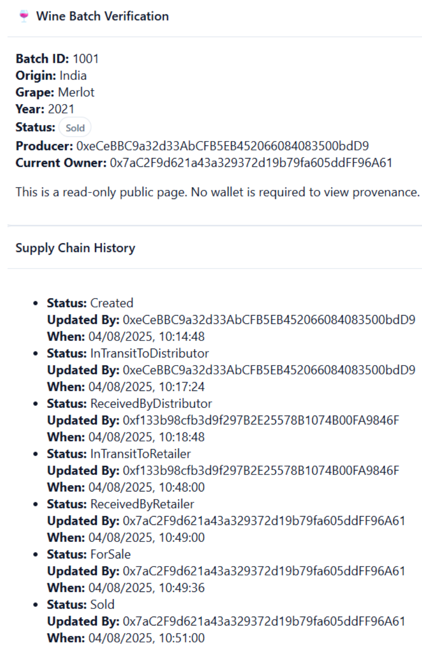

Fig. 12. Public verification page for wine batch #1004, showing origin, grape type, production year, current owner, and full transaction history. The page provides consumers with immutable blockchain-based provenance data, ensuring authenticity and transparency without requiring a wallet or blockchain knowledge.

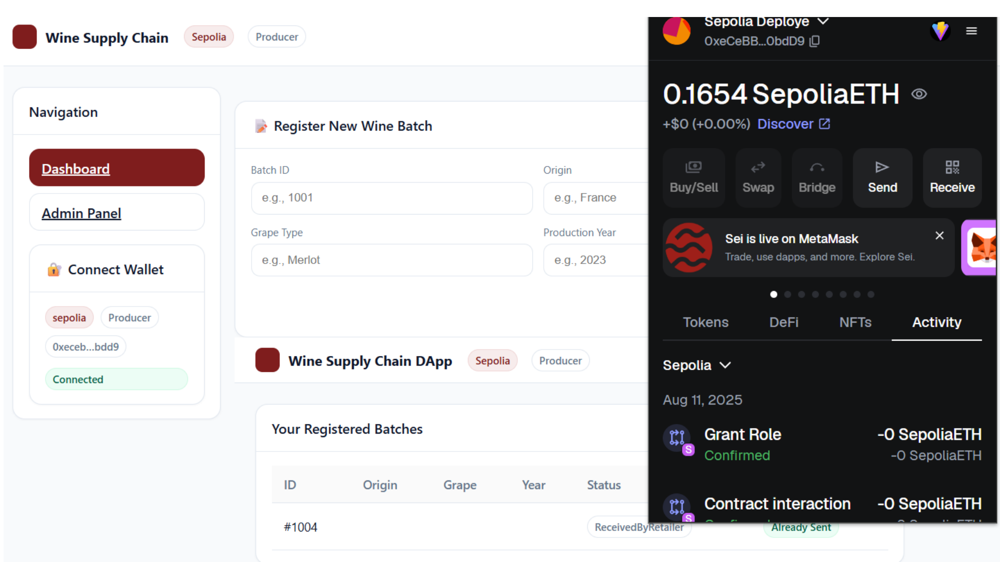

Fig. 13. Admin panel role assignment reflected in MetaMask activity log. After the administrator grants a stakeholder role (e.g., Producer) via the DApp, the transaction is confirmed on the Sepolia testnet and appears in MetaMask under the *Activity* tab. This ensures transparency by providing on-chain proof of access control changes.

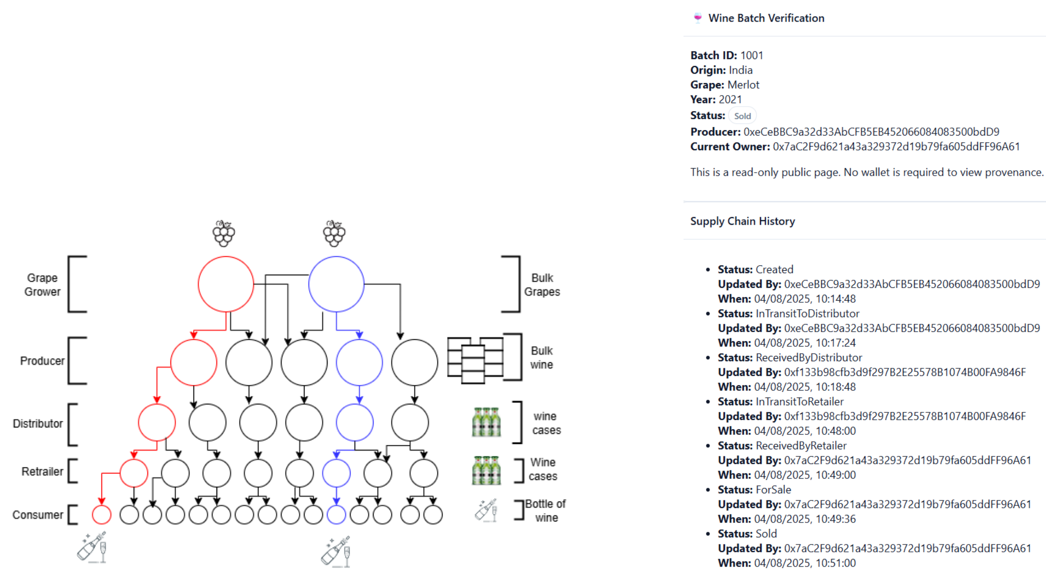

Fig. 14. Comparison of traditional wine supply chain vs. public consumer verification

Description: The left diagram illustrates a traditional wine supply chain flow from grape grower to consumer, with multiple intermediary stages but no transparent, verifiable record accessible to end-users. The right screenshot demonstrates our system's consumer verification interface, where batch details and full transaction history are publicly accessible without requiring blockchain interaction or wallet connection. This implementation successfully achieves *consumer-level provenance verification* without directly exposing the underlying blockchain, bridging the gap between complex blockchain infrastructure and simple user-friendly validation.

9.2 Admin Dashboard — Role Assignment

Description: Shows the administrator's interface for granting roles (`PRODUCER`, `DISTRIBUTOR`, `RETAILER`) to Ethereum addresses.

Purpose: Demonstrates secure role-based access control (RBAC) enforced by the smart contract.

Key Elements Visible:

- Wallet connection status.
- Input fields for Ethereum addresses.
- Role selection dropdown.
- "Grant Role" action button.
- Confirmation messages upon successful transactions.

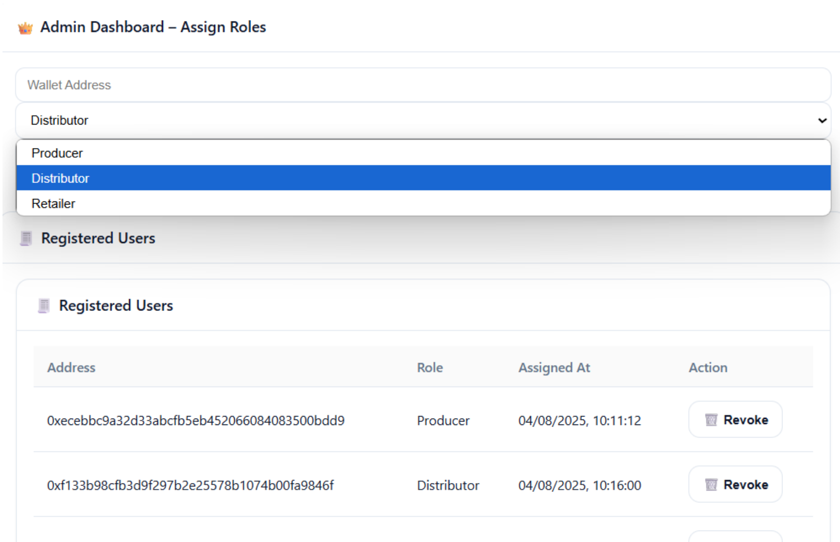

Fig. 15. Admin Dashboard showing role assignment interface in the Wine Supply Chain Traceability System.

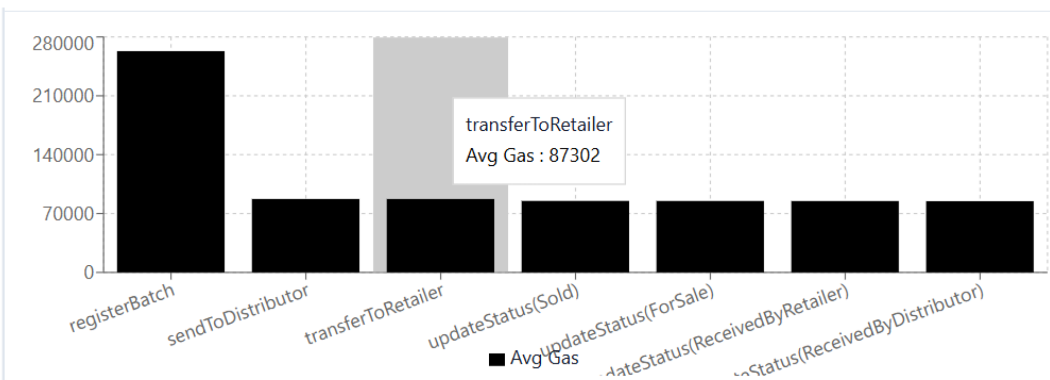

Fig. 16. Gas Consumption Chart from Admin Panel. This chart visualises the average gas consumption of different smart contract functions, enabling administrators to identify high-cost operations. As observed, the `registerBatch` function consumes the most gas (263,081 units), while transactional updates like `sendToDistributor` and `transferToRetailer` have significantly lower costs (87,000 units). This helps in optimising function design for cost-efficiency.

Metrics (Gas & Latency)

Download JSON | Clear Metrics | 8 records

Actor	Action	Batch	Gas	Click→Hash (ms)	Click→Confirm (s)	Block	When
Producer	registerBatch	1025	263081	12818	26.11	8948492	09/08/2025, 19:16:51
Retailer	updateStatus(Sold)	1100	85004	7582	20.78	8941123	08/08/2025, 18:35:53
Retailer	updateStatus(ForSale)	1100	84928	8302	21.11	8941121	08/08/2025, 18:35:31
Retailer	updateStatus(ReceivedByRetailer)	1100	84852	8023	27.2	8941119	08/08/2025, 18:35:06
Distributor	transferToRetailer	1100	87302	8129	17	8941115	08/08/2025, 18:34:16
Distributor	updateStatus(ReceivedByDistributor)	1100	84700	12519	32.45	8941113	08/08/2025, 18:33:54
Producer	sendToDistributor	1478	87325	18087	31.31	8941105	08/08/2025, 18:32:16
Producer	registerBatch	1456	263081	9478	22.57	8941101	08/08/2025, 18:31:27

Fig. 17. Function Execution Metrics and Latency. The admin panel displays real-time metrics for each transaction, including gas used, latency from click-to-hash and click-to-confirm, and the block number in which the transaction was recorded. This information is critical for monitoring system performance, identifying bottlenecks, and validating successful execution across the Ethereum Sepolia testnet.

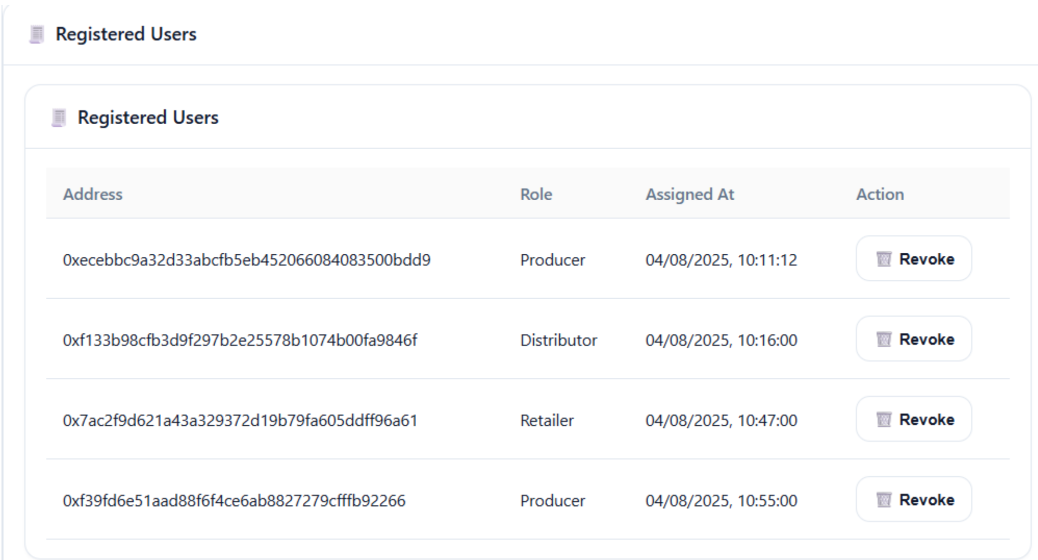

Fig. 18. Role Assignment Overview. This table lists all registered users along with their assigned roles (Producer, Distributor, Retailer) and associated Ethereum wallet addresses. Role-based access control ensures that only authorised stakeholders can perform specific actions in the supply chain, thereby maintaining the integrity of wine provenance records.

9.3 Producer, Distributor, and Retailer Dashboards — Transaction Stages

Description: Sequential views from each stakeholder's dashboard during a batch's lifecycle.

Purpose: Illustrates how stakeholders interact with the smart contract according to their assigned permissions.

Screenshots Included:

- **Producer:** Registering a new batch (with auto-generated QR code for public verification).
- **Distributor:** Receiving a batch from the producer and updating status.
- **Retailer:** Receiving the batch, marking it "For Sale," and recording it as "Sold."

Key Elements Visible:

- Transaction forms and status fields.
- QR code download option.
- Status change timestamps.
- Transaction hash links to Etherscan.

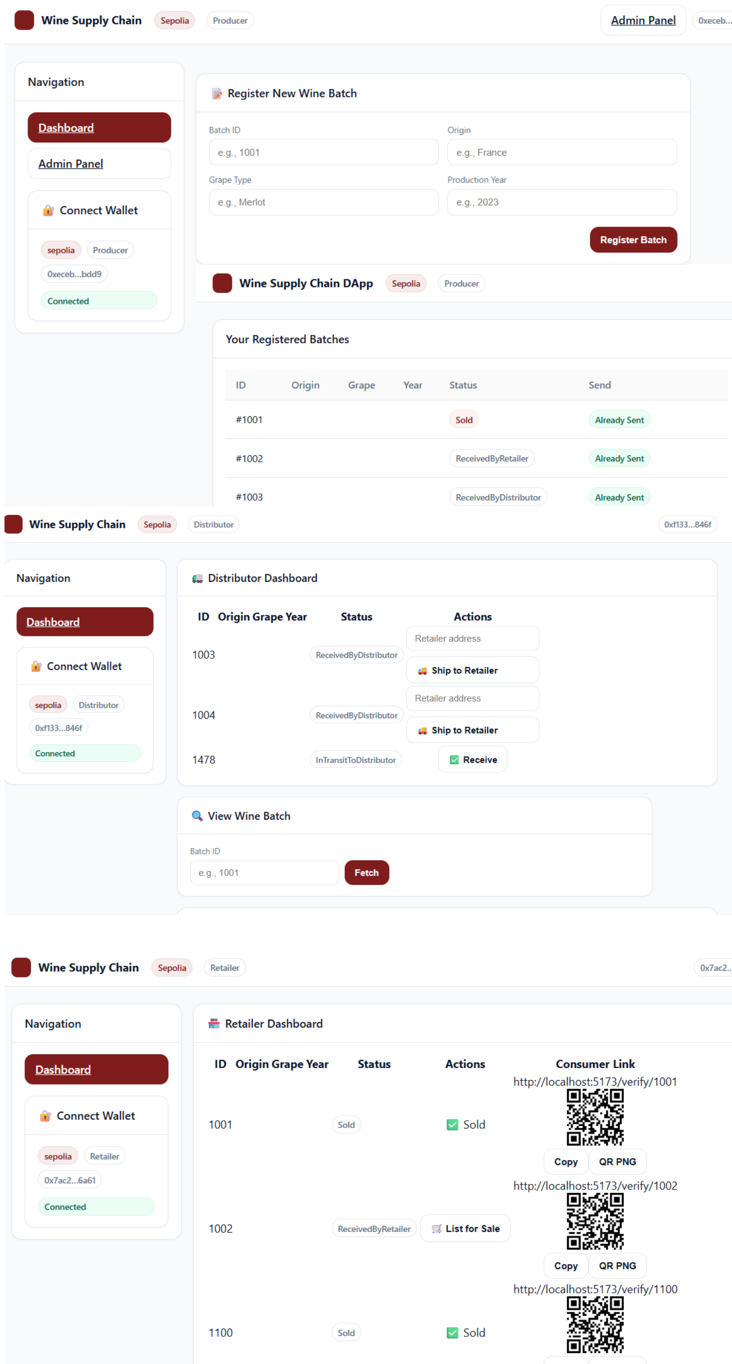

Fig. 19. Producer, Distributor, and Retailer dashboards showing key transaction stages in the Wine Supply Chain Traceability System.

9.4 MetaMask Wallet Connection

Description: Shows the process of connecting an Ethereum wallet to the DApp via MetaMask.
Purpose: Demonstrates how stakeholders authenticate and authorize blockchain transactions.
Key Elements Visible:

- Connection prompt with detected account.
- Network selection (Sepolia Testnet).
- Account address and balance displayed after connection.

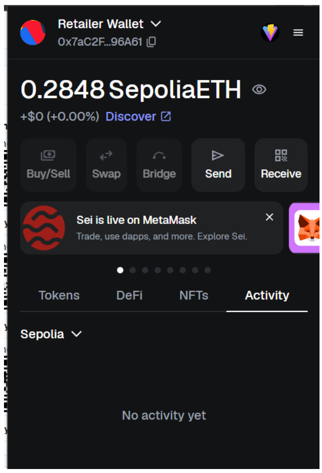

Fig. 20. MetaMask wallet connection process showing account detection, network selection, and balance display in the Wine Supply Chain DApp.

9.5 Etherscan Transaction View

Description: Etherscan page showing a transaction hash for a recorded batch registration.
Purpose: Confirms blockchain transaction finality and public auditability.
Key Elements Visible:

- Contract address and verified source code.
- Transaction details (hash, gas used, block number).
- Event logs emitted by the smart contract.

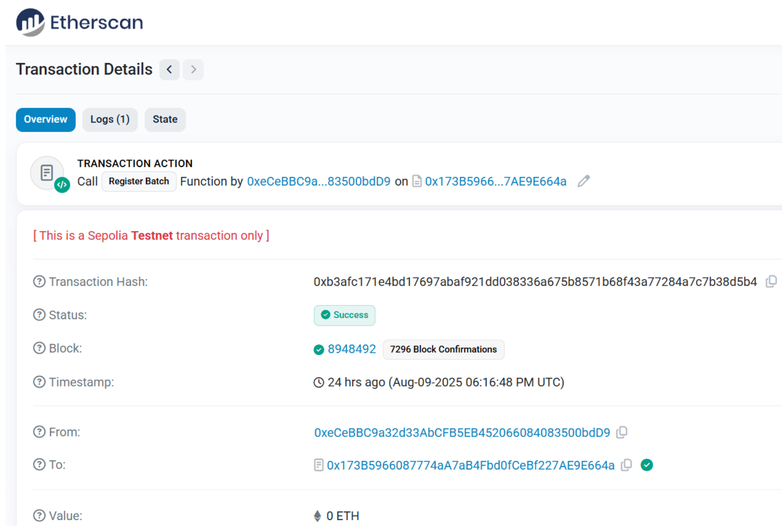

Fig. 21. Etherscan transaction view confirming recorded batch registration and contract event logs.

Full Gas & Performance Logs

This appendix contains the complete benchmarking data collected during the testing phase of the `WineSupplyChain` smart contract. All measurements were taken on the Sepolia Testnet, using Solidity 0.8.20 with the optimizer enabled (runs = 200). Gas data was obtained using the `hardhat-gas-reporter` plugin, and performance metrics were captured using MLflow integrated with the DApp frontend.

9.6 Gas Consumption Benchmarks

Gas Consumption for `WineSupplyChain` Contract Functions

(Solidity 0.8.20, Optimizer: Enabled, Runs = 200)

Function	Min	Max	Avg	#	Notes
registerBatch	240,048	274,440	266,707	45	Baseline batch creation
registerBatches	—	—	688,926	1	Multi-batch creation; ~13% lower cost per batch
sendToDistributor	65,457	65,469	65,461	9	Role-gated transfer to distributor
updateStatus	62,746	63,138	62,902	41	Frequent status updates
transferToRetailer	65,424	65,436	65,428	3	Role-gated transfer to retailer
viewBatch (read)	—	—	21,000*	—	Read-only; no state change (*approx.)
grantRetailer	—	—	97,497	2	Role assignment
grantRole	52,135	52,147	52,143	94	Internal role management

9.7 MLflow Transaction Timing Metrics

Transaction Latency Metrics (Normal vs Peak Conditions)

Performance metrics were captured for Click⟶Confirm and Click⟶Hash times during normal and high network congestion.

Function	Click→Confirm (Normal)	Click→Confirm (Peak)	Click→Hash (ms)	Notes
registerBatch	15–28 s	45–60 s	210–340	Higher during peak due to block inclusion delay
updateStatus	14–25 s	42–55 s	180–310	Frequent usage in operational workflows
transferToRetailer	16–27 s	47–59 s	220–360	Similar confirmation pattern to `sendToDistributor`
sendToDistributor	15–26 s	46–58 s	200–330	Slightly faster in low-congestion windows

9.8 A — Installation, Testing & Deployment Guide Prerequisites

- **OS:** Windows, macOS, or Linux
- **Node.js:** v18 LTS (recommended)
  - **Check:** `node -v` → should be v18.x.x
  - If needed: use `nvm` to install Node 18
- **npm:** ships with Node 18 (`npm -v`)
- **Git:** `git -version`
- **MetaMask** browser extension (Chrome / Brave / Firefox)
- A **Sepolia RPC endpoint** (Alchemy or Infura)
- **Sepolia test ETH** (get via faucet)
- *Tip:* If you've previously installed Node 16 for older tools, switch to Node 18 before running this project.

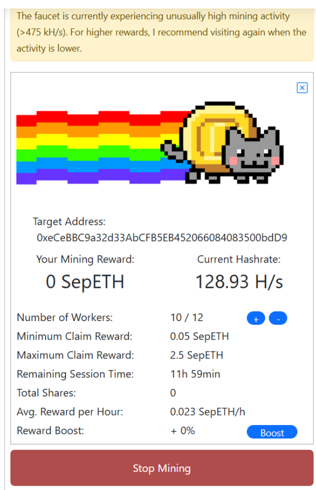

Fig. 22. Sepolia faucet mining interface for funding the DApp wallet. The screenshot shows an active mining session at sepolia-faucet.pk910.de, targeting the Wine Supply Chain DApp's deployed contract wallet address. Mining rewards in SepoliaETH are accumulated based on hashrate and session duration, providing the necessary testnet funds to cover transaction gas costs during development and testing.

9.9 Project Structure

9.10 Backend (Hardhat)

cd wine-supply-chain npm install

```
# Install Hardhat and toolbox
npm install --save-dev hardhat @nomicfoundation/hardhat-toolbox

# Install OpenZeppelin contracts
npm install @openzeppelin/contracts
```

```
cd ../wine-ui
npm install
```

9.11 Configure RPC (Alchemy or Infura)

Option 1: Alchemy

(1) Create an Alchemy account.
(2) Create a Sepolia app.
(3) Copy the HTTPS RPC URL.
(4) Create an API Key (shown in the app console; the key is embedded in the URL).

Option 2: Infura

(1) Create an Infura account.
(2) Create a new project.
(3) Enable Sepolia.
(4) Copy the HTTPS RPC URL and your Project ID.

9.12 MetaMask Setup (Sepolia)

(1) Open MetaMask → `Networks` → `Add network` → select `Sepolia` (or enable "Show test networks").
(2) Get Sepolia test ETH from a faucet.
(3) Prepare at least two addresses for testing roles (`Producer`, `Distributor`, `Retailer`). MetaMask allows creating multiple accounts.

9.13 Hardhat: Environment Variables

In the `wine-supply-chain/` directory, create a file named `.env`:
enter your own details like this

```
SEPOLIA_RPC_URL=https://eth-sepolia.g.alchemy.com/v2/CaxdqPetwLuM19fM6Ix_m
DEPLOYER_PRIVATE_KEY=8843763e90a0772268da8a8f3ec4c2d6dcb107063d40040b5ffb091c2cfac002
ETHERSCAN_API_KEY=EFEMJ9NWVMGY195U4G2ZWF4Z8WYDBVI9SY
```

9.14 Compile, Test, and (Optional) Security Scan

```
cd wine-supply-chain
npx hardhat clean
npx hardhat compile

# run tests if you have them
npx hardhat test
```

9.15 Run Deployment

```
npx hardhat run scripts/deploy.js --network sepolia
```

Copy the deployed contract address printed in the console.

9.16 Export ABI to Frontend

After compiling, Hardhat generates the ABI at:

```
wine-supply-chain/artifacts/contracts/WineSupplyChain.sol/WineSupplyChain.json
```

Copy that file to the frontend:

```
cp wine-supply-chain/artifacts/contracts/WineSupplyChain.sol/WineSupplyChain.json \
   wine-ui/src/abi/WineSupplyChain.json
```

Every time re-deploy or change the contract, re-copy this ABI

9.17 Frontend Environment (.env)

In `wine-ui/.env`:
Enter with your own details

```
VITE_CONTRACT_ADDRESS=0x173B5966087774aA7aB4Fbd0fCeBf227AE9E664a
VITE_NETWORK=sepolia
VITE_ALCHEMY_URL=https://eth-sepolia.g.alchemy.com/v2/CaxdqPetwLuM19fM6Ix_m
```

9.18 Run the Frontend Locally

```
cd wine-ui
npm run dev
```

Open the local URL (usually http://localhost:5173).

Dashboard (with MetaMask): The home page displays role-based components after connecting your wallet.

9.19 Role Assignment (Admin)

Your `AdminDashboard` calls `grantRole()` from the connected admin wallet.

Make sure your admin account is the one that deployed the contract (or any account with `DEFAULT_ADMIN_ROLE`).

After granting roles, users can reconnect and the UI will display their role automatically.

in App.jsx enter deployed address for admin role check

You can assign roles in two ways:

- **From Hardhat terminal** — Suitable for initial setup before the frontend is connected.
- **From the Admin Panel UI** — Suitable for ongoing role management in production.

For this project, since the `AdminDashboard` already calls `grantRole()`, roles can be assigned directly in the UI once the contract is deployed to Sepolia and MetaMask is connected with the admin account.

If you prefer to assign roles from the Hardhat terminal (e.g., immediately after deployment), run:

```
npx hardhat console --network sepolia
```