Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/WC02.20250278
Published: October 14, 2025

Enhancing ERP Systems with Blockchain for Engineering and Asset Management

Md Didarul Alam
Department of Electrical and Computer Engineering
Georgia Southern University
Statesboro, GA,30458, USA
ma18778@georgiasouthern.edu

Tasfia Tarannum
Department of Data Analysis and Supply Chain Management
Southern Arkansas University
Magnolia, AR, 71753, USA
ttarannum3228@muleriders.saumag.edu

Hayder Zghair
Department of Engineering and Physics
Southern Arkansas University
Magnolia, AR, 71753, USA
hzghair@saumag.edu

Abstract

Enterprise Resource Planning (ERP) systems like SAP are widely used to manage engineering and maintenance tasks. They help with scheduling maintenance, managing spare parts, and tracking work orders. However, these systems are centralized, which can lead to issues with trust, data security, and audit accuracy—especially when multiple companies are involved. This paper proposes adding a blockchain layer to ERP systems to address these issues. The blockchain records maintenance activities in an unchangeable format, tracks the origin of parts, and uses smart contracts to automate compliance and service agreements. Data flows between the ERP and blockchain through APIs, and IoT sensors can trigger actions in real time. This approach improves transparency, reduces the risk of data tampering, and makes audits easier. Early results show that it can strengthen trust between stakeholders and improve regulatory compliance without replacing existing ERP functions.

Keywords

Blockchain, Enterprise Resource Planning (ERP), Engineering Automation, Internet-of-Things (IoT), Maintenance Management, Smart Contracts.

1. Introduction

Enterprise Resource Planning (ERP) systems, such as SAP, Oracle, and Microsoft Dynamics, play a central role in engineering and maintenance management. These platforms automate critical tasks, including preventive maintenance scheduling, spare parts tracking, procurement, and compliance reporting. By consolidating information into a single system, ERP solutions improve efficiency and reduce manual errors.

Despite these advantages, current ERP systems are limited by their centralized design. Data is stored and controlled within one organization's system, which makes it difficult to establish trust when multiple stakeholders—such as contractors, suppliers, and auditors—are involved. Centralized systems also face challenges related to data tampering, audit transparency, and reconciliation across company boundaries. In industries where safety, reliability, and compliance are critical, these issues can result in delays, disputes, and higher operational costs (Figure 1).

Figure 1. Different Modules of ERP (SAP), Halder, N. (2023).

Blockchain technology offers a potential solution. As a decentralized and tamper-proof ledger, blockchain can provide transparent records of maintenance events, secure tracking of spare parts, and automated enforcement of service agreements through smart contracts. When integrated with ERP systems, blockchain has the potential to strengthen trust, ensure data integrity, and simplify compliance without replacing existing ERP functionality.

This paper proposes a framework for integrating blockchain with ERP systems to improve engineering and maintenance automation. The framework introduces a permissioned blockchain layer connected to ERP modules through APIs. It supports immutable record keeping, spare part provenance tracking, smart contract-driven work orders, and IoT-triggered maintenance actions.

The remainder of this paper is organized as follows: Section 2 reviews existing research on ERP and blockchain in enterprise applications. Section 3 defines the challenges in current ERP-based maintenance systems and formulates the objectives. Section 4 presents the proposed blockchain-ERP integration framework. Section 5 shares the results and discusses expected benefits and challenges. Finally, Section 7 concludes with key findings and directions for future work.

2. Literature Review

Enterprise Resource Planning (ERP) systems are widely used in asset-intensive industries to coordinate engineering and maintenance activities across different departments. In practice, ERP modules cover preventive maintenance, predictive maintenance, breakdown or corrective maintenance, spare parts management, procurement, and service contracts. Preventive maintenance is typically managed by generating schedules and work orders to ensure regular inspection and servicing of equipment, thereby reducing unplanned downtime and extending asset life cycles. Predictive maintenance integrates sensor data and analytics into ERP systems to anticipate failures before they occur, aiming to reduce costs associated with unnecessary inspections while also avoiding catastrophic breakdowns. Breakdown maintenance, also known as corrective maintenance, deals with unexpected failures, where ERP systems are used to log incidents, assign technicians, and track mean time to repair (MTTR). Spare parts and inventory management are critical, as unavailability of components can lead to prolonged downtime; ERP modules monitor

stock levels, automate requisitions, and link with procurement processes. Service contracts and warranties are often embedded into ERP workflows, defining responsibilities of contractors and equipment vendors, while procurement modules handle purchase requisitions, supplier evaluation, and payment approval. While these functions are vital, ERP systems are highly centralized: they store data in a single organizational database, and in multi-stakeholder settings this raises concerns over auditability, trust, and data authenticity.

Researchers have proposed blockchain as a complementary technology to address these shortcomings. Moalagh and Ebrahimi Ghadi (2020) introduced a conceptual blockchain-based ERP architecture in which preventive and corrective maintenance activities are notarized onto a ledger and payments to maintenance contractors and spare part vendors are validated through smart contracts. Their approach highlighted how immutability can improve accountability in maintenance operations, but remained at a theoretical level without empirical implementation. Alqaryuti *et al.* (2025) designed a blockchain-driven framework for preventive maintenance in hydraulic systems, integrating scheduling rules with spare parts availability. They used smart contracts to ensure that maintenance tasks cannot be skipped and that required parts are available before servicing is authorized, thereby linking maintenance planning with supply chain trust. Al-Refaie *et al.* (2022) explored breakdown maintenance by modeling each repair event as a "block" within a blockchain. Their work emphasized how maintenance planning can be optimized by ensuring that repair logs are immutable, allowing managers to schedule resources more effectively and preventing disputes over whether work was performed as documented. Johnny and Wales (2024) applied blockchain to building maintenance and condition auditing. They proposed a model in which inspection results and compliance documents are recorded immutably, giving regulators and facility managers greater confidence in the accuracy of audit trails. Together, these works demonstrate blockchain's relevance to preventive, corrective, and audit-related maintenance activities, but most of them stop short of showing how blockchain could be integrated directly into ERP systems such as SAP EAM.

Spare parts management has also been a major focus of blockchain research, especially in industries like aviation and manufacturing where counterfeit risks are high. Ho *et al.* (2021) developed a blockchain-based traceability system for aircraft spare parts, where each part's lifecycle—from manufacturing to calibration, installation, and eventual replacement—is recorded on a shared ledger accessible to operators and regulators. This ensured authenticity, compliance, and timely recall management, addressing weaknesses of ERP systems that merely track part numbers without verifying origin. A related study (Hasan *et al.*, 2020) proposed using blockchain and InterPlanetary File System (IPFS) for manufacturing spare parts management. The system records ownership transfers and links on-chain hashes to off-chain storage of technical documents, enabling authenticity checks at each transaction stage. While highly promising, these studies are often isolated from ERP environments, leaving open questions about how blockchain provenance records could be synchronized with ERP procurement and maintenance workflows.

Predictive maintenance represents another area where blockchain has been suggested. Alabadi *et al.* (2023) presented a blockchain-based predictive maintenance framework combining IoT sensors, deep learning analytics, and IPFS for decentralized storage. Their design anchored both raw sensor telemetry and AI model predictions onto blockchain, ensuring tamper-proof provenance from data collection to decision outputs. Similarly, the ChainSplitter architecture (2019) proposed a scalable blockchain design for industrial IoT, capable of timestamping and verifying high-frequency sensor data while controlling storage costs. These approaches strengthen the integrity of predictive maintenance data but rarely address how verified events can trigger ERP-based work orders or be reconciled with existing scheduling and reporting systems. Without ERP integration, blockchain-enhanced predictive maintenance remains a siloed innovation rather than a transformative enterprise solution.

Blockchain has been applied beyond ERP and maintenance to other trust-critical domains such as crowdsourcing. Li *et al*. (2019) proposed CrowdBC, a blockchain-based decentralized framework for crowdsourcing, addressing limitations of centralized platforms such as single point of failure, high service fees, and vulnerability to Sybil attacks. Their system implemented smart contracts for user registration, task posting, and reward distribution, ensuring tamper-proof and transparent execution without intermediaries. While their focus was on crowdsourcing tasks rather than engineering maintenance, the underlying principles—immutability, decentralized consensus, and smart contract enforcement—are highly transferable to ERP-based maintenance processes. For example, preventive maintenance scheduling could be handled similarly to "task posting" in CrowdBC, where maintenance orders are posted and verified by multiple parties, while service contracts and spare part deliveries could mirror the automated settlement logic in CrowdBC smart contracts.

The CrowdBC work also highlighted technical challenges such as data storage limitations, privacy of participants, and the need for reputation mechanisms to ensure worker quality. These challenges align closely with ERP maintenance, where sensitive operational data, contractor trust, and part authenticity are critical issues. By adapting the CrowdBC framework to ERP, blockchain could be used to record immutable logs of maintenance activities, track spare parts provenance, and automate SLA enforcement, reducing disputes and enhancing collaboration between operators, suppliers, and regulators.

Service contracts and SLA enforcement also stand to benefit from blockchain integration. Seraj (2024) demonstrated how smart contracts can be used in infrastructure maintenance projects to automate contractor assignment, progress logging, and payment release once milestones are verified. This approach reduces disputes and accelerates settlement processes compared to manual ERP workflows. In a more general enterprise context, López-Pintado *et al.* (2018) introduced the CATERPILLAR engine, which compiles business process models into Ethereum smart contracts for execution. Although not designed for maintenance, their work illustrates how ERP workflows such as procurement approvals, warranty claims, or SLA validations can be encoded as smart contracts to reduce manual intervention and enforce rules consistently.

Across these domains, blockchain has been applied to preventive maintenance scheduling, breakdown repair logging, spare parts provenance, predictive data integrity, and service contract enforcement. However, most existing studies are either conceptual frameworks or small-scale pilots. Few works explicitly address ERP integration, where blockchain acts as a complementary trust layer to enhance—not replace—core ERP functions such as SAP's Plant Maintenance (PM) and Materials Management (MM) modules. Even fewer provide quantitative evaluation of key performance indicators such as audit retrieval time, SLA settlement time, warranty dispute resolution, or mean time to repair. This gap highlights the need for targeted research on blockchain-enabled ERP maintenance, where blockchain not only strengthens data trust and auditability but also improves operational efficiency and multi-party collaboration across the full lifecycle of maintenance activities.

3. Problem Statement and Objectives

Engineering and maintenance activities in ERP systems such as SAP PM (Plant Maintenance) module span preventive, predictive, and corrective maintenance, as well as spare parts logistics, service contracts, and procurement through MM (Material Management) module. While these modules streamline operations, they are centralized, which raises issues of auditability, trust, and data integrity in multi-stakeholder contexts involving operators, contractors, OEMs, and regulators (Moalagh & Ebrahimi Ghadi, 2022). A blockchain-enabled ERP framework is proposed to address these challenges by offering immutable records, provenance tracking, and automated contract execution through smart contracts (Seraj, 2024).

Formally, let $\boldsymbol{A} = \{1, \ldots, N_a\}$ denote the set of assets and $\boldsymbol{T} = \{1, \ldots, N_t\}$ the set of maintenance tasks. Each task $i \in \boldsymbol{T}$ is characterized by a type $\tau_i = \{PM, PdM, BR\}$, asset mapping $a(i) \in \boldsymbol{A}$,
planned start s_i, deadline d_i, and evidence bundle $\boldsymbol{E}_i$. The ERP system records a task status $x\_i(t) \in \{created, in-progress, completed, approved\}$. A blockchain layer complements this with an on-chain tuple:

$$z_i = (hash(\boldsymbol{E}_i), CID(\boldsymbol{E}_i), t_i^{commit}, signers_i)$$

where large artifacts (inspection reports, calibration certificates, IoT telemetry) are stored off-chain and referenced via cryptographic hashes (TechScience, 2021).

Key performance indicators (KPIs) for this integration include:

$$ART = \frac{1}{|T|} \sum_{i \in T} (t_i^{audit} - t_i^{request}) \quad (1)$$

the Audit Retrieval Time (ART), measuring the effort to generate verifiable audit bundles.

$$SST = \frac{1}{|T_\$|} \sum_{i \in T_\$} (t_i^{paid} - t_i^{completed}) \quad (2)$$

the SLA Settlement Time (SST), capturing delays between task completion and payment release and the Provenance Completeness (PC) metric, denoting the proportion of spare parts whose entire lifecycle is cryptographically linked.

Constraints are imposed to ensure data integrity and consistency (Ho *et al.*, 2021):

$$signers_i \cap R_i| \geq r_i \quad (3)$$

$$hash(\overline{E_i}) = hash(E_i)$$
$$x_{i(t)} = approved \Rightarrow \exists\, z_i\, with\, t_i^{commit} < t$$

where,
$signers_i$: the set of parties who signed (endorsed) the maintenance task i, r_i: minimum endorsement threshold, R_i: the set of required organizations (e.g., operator, contractor, OEM, regulator).
Smart contracts further encode incentive-compatible settlements:

$$Payout_i = \begin{cases} Escrow_i + Fee_i, & \text{if Sigpolic}_\text{i} \ \wedge \ \text{DeadlineOK} \\ -\,Penalty, & \text{otherwise} \end{cases} \quad (4)$$

where escrow and penalties discourage free-riding and enforce SLA compliance, similar to time-locked deposits in decentralized coordination (Li *et al.*, 2019).

The design challenge is framed as a multi-objective optimization:

$$\min_u J(u) = \alpha_1 ART(u) + \alpha_2 DCT(u) + \alpha_3 SST(u) - \alpha_4 PC(u) + \alpha_5 CPT(u) \quad (5)$$

Subject to throughput $\lambda(u) \geq \lambda^{min}$, latency $L_{95}(u) \leq L_{95}^{max}$, and ERP–blockchain consistency. Here, decision variables $u$represent anchoring policies, batching strategies, endorsement thresholds, and smart contract designs (Lin *et al.*, 2022).

The objectives of this study are fourfold: (i) design an ERP–blockchain integration architecture supporting selective disclosure and off-chain evidence anchoring; (ii) prototype workflows for SLA settlement, part provenance, and IoT-driven PdM escalation; (iii) evaluate the framework against ERP-only baselines using ART, SST, DCT, PC, and system performance; and (iv) align governance policies with ISO 55000 standards for asset management.

The study hypothesizes that blockchain anchoring will reduce ART and DCT by at least 40–50% while maintaining acceptable performance, escrow-backed smart contracts will shorten SST by at least 30% without increasing disputes, and provenance contracts will raise PC above 90%, thereby reducing warranty and recall investigation times. All the KPI values are benchmarked from the data captured from earlier research (Tarannum and Zghair, 2024).

4. Proposed Framework

The proposed framework introduces blockchain as a complementary trust layer to existing ERP systems for engineering and maintenance management. While ERP platforms such as SAP provide comprehensive modules for preventive maintenance, spare parts management, and procurement, they are inherently centralized and rely on internal administrative control. This centralization limits transparency across organizational boundaries, creates disputes over service contracts, and makes provenance verification of spare parts difficult. To overcome these shortcomings, the framework integrates blockchain for secure data anchoring, multi-stakeholder endorsement, and smart contract–based SLA enforcement.

4.1 System Architecture

The architecture is structured into three layers.
ERP Layer: This layer is responsible for day-to-day operations. It handles work order generation, scheduling, spare parts requisitions, procurement workflows, and vendor contracts. From the perspective of plant operators and engineering managers, ERP continues to function as the system of execution and record.

Integration Layer: This middleware connects ERP and blockchain. It converts ERP transactions into blockchain-compatible formats, performs hashing of event data, and groups multiple events into Merkle trees to reduce anchoring costs. It also ingests IoT and sensor data from predictive maintenance systems and ensures synchronization between ERP's relational database and the blockchain ledger.

Blockchain Layer: This is the distributed trust backbone. It stores cryptographic digests of ERP events, executes smart contracts, and ensures that only authorized stakeholders can endorse transactions. The blockchain maintains an immutable log of all maintenance activities and provides a shared, tamper-proof audit trail across internal and external stakeholders (Figure 2).

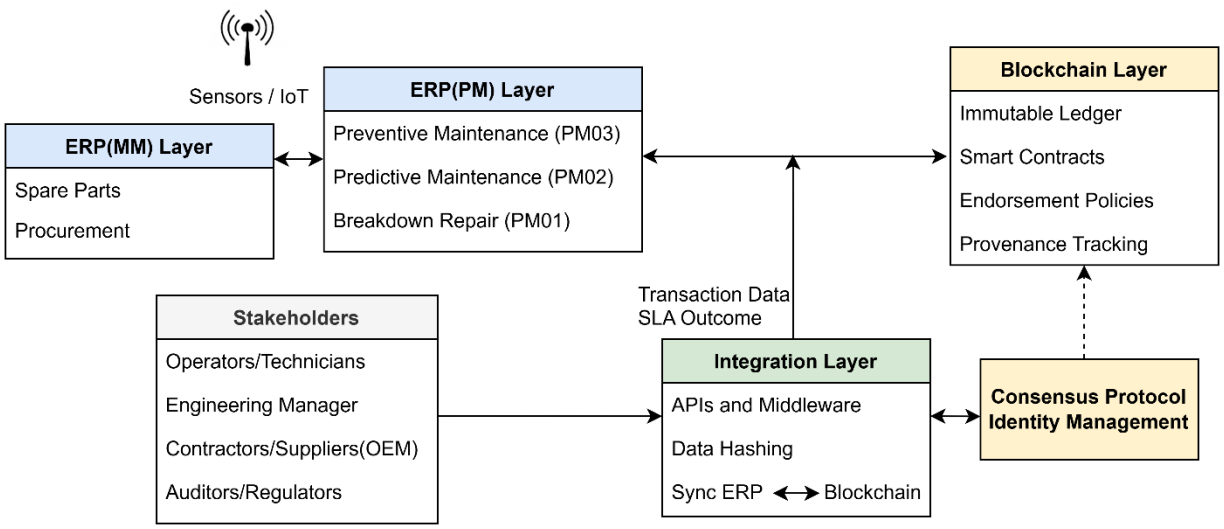

Figure 2. Proposed ERP–Blockchain architecture for maintenance management.

4.2 Workflow of Maintenance Transactions

The interaction between the three layers unfolds in a sequence of well-defined steps:
Event Creation and Hashing: Whenever a preventive, predictive, or corrective maintenance task is created in the ERP, the transaction $E_i$is encoded and a secure hash $h_i = H(E_i)$is generated. This ensures that sensitive details remain off-chain while their integrity can always be verified.

Batching and Anchoring: Hashes from multiple ERP events are combined into a Merkle tree, and only the root $M$is committed to the blockchain. This mechanism allows thousands of ERP transactions to be validated using a single blockchain record, balancing performance with cost efficiency.

Endorsement and Consensus: The proposed ERP–Blockchain framework is designed for enterprise environments where multiple stakeholders such as vendors, maintenance providers, and auditors participate in validating transactions. A permissioned blockchain architecture is adopted to ensure controlled participation and predictable performance. Consensus is achieved through protocols such as Practical Byzantine Fault Tolerance (PBFT) or Istanbul Byzantine Fault Tolerance (IBFT), which are widely implemented in platforms like Hyperledger Fabric, Quorum, and Besu.

In this model, endorsement policies define how many organizational peers must validate a transaction before it can be committed to the ledger. For example, a policy requiring two out of three endorsers ensures resilience against a single faulty or malicious participant. This mechanism guarantees that maintenance records, SLA executions, and provenance logs are only appended after collective agreement among trusted entities. Such a design provides both immutability and trust, while avoiding the high computational costs typical of public blockchains.

Smart Contract Execution: Service Level Agreements (SLAs) are encoded as smart contracts. Once a maintenance task is logged, the contract automatically checks deadlines and quality conditions. If the task is completed on time and passes quality checks, payment is automatically released from escrow. If not, penalties are applied without requiring lengthy dispute resolution.

Smart Contract implements the automated workflow of SLA enforcement for maintenance tasks in the ERP–Blockchain framework. When a maintenance event is initiated within the ERP system, it is submitted to the blockchain where the endorsement procedure begins. Each required stakeholder, such as a plant operator, contractor, or regulatory body, is asked to digitally sign the event. Once the number of endorsements reaches the pre-defined threshold, the task status changes to “endorsed,” ensuring that no single party can unilaterally validate a task. Following endorsement, the completion procedure is invoked once the task has been carried out. Here, the contractor

Algorithm 1 SLA Smart Contract Workflow

procedure ENDORSE(taskId)
require(sender ∈ requiredOrgs[taskId])
endorsements[taskId] ← endorsements[taskId] ∪ sender
if $|endorsements[taskId]| \geq threshold[taskId]$ **then**
  status[taskId] ← endorsed
  emit TaskEndorsed(taskId)
end if
end procedure
procedure COMPLETE(taskId, t_{exec}, q_i, proof)
require(status[taskId] ∈ $\{pending, endorsed\}$)
if merkleRoot[taskId] set **then**
  require(VerifyMembership(evidenceHash[taskId], proof, merkleRoot[taskId]) = true)
end if
completionTime[taskId] ← t_{exec}
qualityScore[taskId] ← q_i
status[taskId] ← approved
emit TaskApproved(taskId, t_{exec}, q_i)
end procedure
procedure SETTLE(taskId)
require(status[taskId] = approved)
onTime ← (completionTime[taskId] ≤ deadline[taskId])
okQual ← (qualityScore[taskId] ≥ qualityReq[taskId])
if onTime **and** okQual **then**
  payout ← escrow[taskId] + fee[taskId]
  Transfer(payout, payee[taskId])
  status[taskId] ← settled
  emit TaskSettled(taskId, payout)
else
  penalty ← Penalty(taskId)
  ApplyPenalty(penalty)
  status[taskId] ← penalized
  emit TaskPenalized(taskId, penalty)
end if
end procedure

Algorithm 2 Provenance Recording Protocol

procedure RECORDEVENT(serial, stage, hash, actor, time, geo)
require(Authz(actor, stage) = true)
if stage = install **then**
  require(seen[serial][recv] = true)
end if
require(seen[serial][stage] = false)
events[serial] ← events[serial] ∪ (stage, hash, actor, time, geo)
seen[serial][stage] ← true
emit PartEvent(serial, stage, hash, actor, time)
end procedure
procedure GETLINEAGE(serial)
return events[serial]
end procedure
procedure ISCOMPLETE(serial)
for st ∈ requiredStages **do**
  if seen[serial][st] = false **then**
    return false
  end if
end for
return true
end procedure
procedure PC_SUMMARY(serialList)
cnt ← 0
for each s ∈ serialList **do**
  if ISCOMPLETE(s) **then**
    cnt ← cnt + 1
  end if
end for
return cnt / length(serialList)
end procedure

or maintenance executor submits the proof of completion, which may include the hash of a work report, sensor data, or a Merkle proof that links back to off-chain evidence. The smart contract verifies that the task had indeed been endorsed and that the evidence provided matches the stored hash, after which the task is marked as "approved." The final stage is the settlement procedure, which automatically evaluates whether the execution occurred before the deadline and whether the quality score meets or exceeds the required threshold. If both conditions are satisfied, the smart contract releases the escrowed payment along with any associated fees, changing the task status to "settled." If the task is delayed or fails to meet quality requirements, it applies a penalty and updates the status to "penalized."

Provenance Recording: Spare parts are tracked across their entire lifecycle—from manufacturing, shipping, and warehouse entry to installation and eventual disposal. Each lifecycle event generates an entry that is anchored on-chain, resulting in a tamper-proof provenance log that auditors and regulators can independently verify. It governs the recording and verification of the provenance of spare parts across their lifecycle. Each time a part transitions from one stage to another, such as manufacturing, shipment, receipt, installation, calibration, or retirement, the RECORDEVENT procedure is executed. The system first verifies that the actor attempting to record the event is authorized for that lifecycle stage, and it ensures that events cannot be duplicated or recorded out of order—for instance, a part cannot be installed unless it has first been received. The resulting event record, consisting of the stage, timestamp, actor identity, and evidence hash, is appended to the blockchain, and a log is emitted to confirm the update. To support auditability, the GETLINEAGE procedure provides an ordered history of all recorded events for a particular serial number, which can then be cross-verified against off-chain ERP records. The algorithm also includes an ISCOMPLETE procedure, which checks whether a part has successfully gone through all required stages of its lifecycle. If any stage is missing, the part is flagged as incomplete, allowing auditors and managers to detect

inconsistencies or possible fraud. Finally, the PC_SUMMARY procedure aggregates the completeness status across a batch of parts to compute the Provenance Completeness (PC) metric, defined as the fraction of parts with a fully recorded lifecycle over the total number of parts. This metric serves as a global measure of trustworthiness in the spare parts supply chain. By anchoring each step to the blockchain, the protocol ensures traceability, prevents counterfeit replacements, and provides regulators with a transparent mechanism for verifying part authenticity without direct access to sensitive ERP databases.

Algorithm 1 and 2 describes each step of the Smart Contract and Provenance Recording Procedure.

4.3 Security and Trust Mechanisms

The strength of the framework lies in its ability to provide trust without relying on a central authority. Consensus mechanisms ensure that no single party can rewrite the history of maintenance records. Identity management restricts transaction signing to verified stakeholders, preventing unauthorized access. Cryptographic immutability guarantees that once data has been anchored, it cannot be altered without detection. For example, even a minor change to a work order or spare part record would generate a completely different hash, immediately exposing tampering attempts.

In a full ERP deployment, these blockchain-based maintenance workflows would naturally extend to the Plant Maintenance (PM) and Materials Management (MM) modules, where a maintenance order can automatically trigger a linked purchase requisition or purchase order in MM, followed by material document posting and service entry sheet updates, ensuring that provenance and compliance are preserved across both operational and supply chain activities.

The proposed ERP–Blockchain framework differs from conventional ERP systems by introducing immutability, consensus-based validation, and smart contract automation. While traditional ERP relies on centralized control, the blockchain layer ensures that maintenance records cannot be altered retroactively and that all stakeholders operate on a shared ledger. Service-level agreements are enforced automatically, reducing settlement delays and disputes. In addition, the framework enables full provenance tracking of spare parts, preventing counterfeit replacements and improving supply chain transparency. This shift transforms ERP from a closed internal system into a distributed, multi-stakeholder trust platform.

The implementation of the framework requires balancing performance, cost, and security through careful parameter selection. Batch size affects both transaction cost and anchoring latency, endorsement thresholds determine the level of trust versus approval speed, and settlement delay influences the trade-off between quick payments and verification assurance. These parameters must be adapted to the industrial context; for example, safety-critical domains may prioritize stricter validation, while high-volume industries may emphasize faster settlement. Thus, the system remains flexible and can be tuned to meet specific operational requirements.

5. Results and Discussion

To validate the feasibility of the framework, smart contracts were implemented in Solidity and tested using the Ethereum Remix IDE. Remix was selected as a prototyping environment to verify the correctness of the contract logic for SLA enforcement and provenance tracking. Although Remix operates in a public-chain EVM setting, the contract logic is fully portable to permissioned enterprise networks (e.g., Hyperledger Fabric or Quorum), where the actual deployment would rely on PBFT/IBFT consensus and endorsement policies as described in Section 4. The test network was configured with a 15-second block time and 20 gwei gas price. Each contract operation—SLA settlement, provenance logging, or dispute resolution—was executed across 100 trials, and transaction logs were exported. Each transaction executed within the Ethereum Virtual Machine (EVM) was logged, capturing gas consumption, execution latency, and settlement outcomes. Offline analysis was performed in Python (3.10, Jupyter Notebook) using NumPy, Pandas, and Matplotlib. Parameter sweeps varied the batch size (B) and endorsement threshold (r) to examine system behavior. Results were plotted as distribution curves, cumulative functions, and trade-off frontiers to highlight performance–cost relationships.

Three main performance indicators were considered: Audit Retrieval Time (ART), SLA Settlement Time (SST), and Dispute Cycle Time (DCT). These were compared across conventional ERP-only operations and the proposed ERP enhanced with blockchain (ERP with BC).

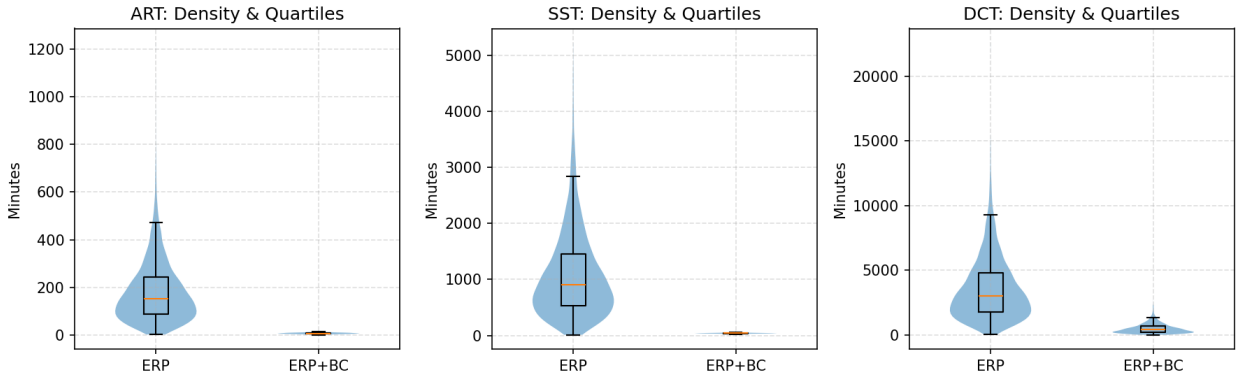

Figure 3. ART, SST, and DCT distributions showing improved speed and reliability with ERP with Blockchain.

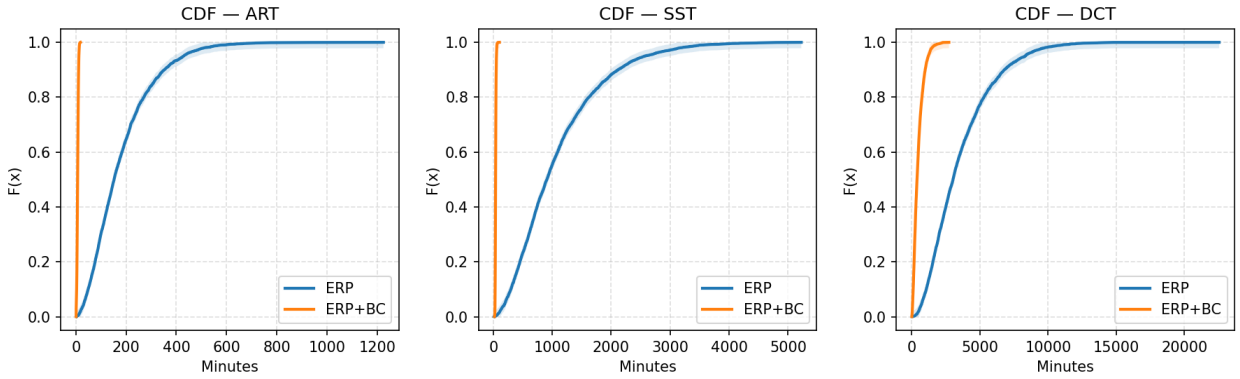

Figure 4. CDFs highlighting faster and more consistent completions under ERP with Blockchain.

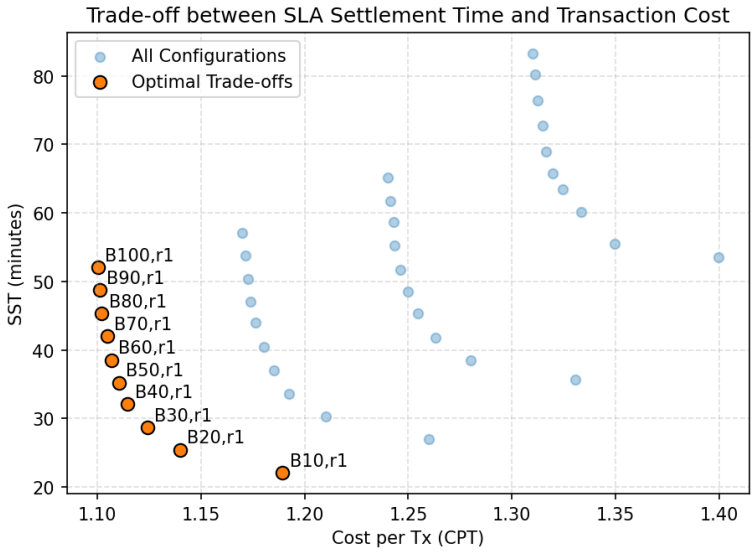

Figure 5. Trade-off between SLA settlement time and transaction cost.

The first set of plots (Figure 3) shows distributional comparisons of ART, SST, and DCT using violin plots with embedded quartile boxplots. For ERP-only processes, all three indicators exhibit wide distributions with heavy tails, reflecting high variability and inefficiencies in data retrieval, SLA enforcement, and dispute resolution. In contrast, ERP with BC significantly compresses the distributions, indicating more consistent and predictable performance. For

instance, ART under ERP-only can extend beyond 400 minutes, whereas ERP with BC consistently completes within a few minutes. A similar trend is observed for SLA settlements, where blockchain-based auto-execution drastically reduces delays caused by manual intervention and batching inefficiencies.

The second set of plots (Figure 4) presents the Cumulative Distribution Functions (CDFs) for ART, SST, and DCT. These plots emphasize the proportion of tasks completed within a given time threshold. Under ERP with Blockchain, nearly all transactions converge to completion at very low latencies, while ERP-only exhibits gradual increases with long delays before reaching full completion. This clearly demonstrates the effectiveness of smart contracts and provenance hashing in accelerating process execution and ensuring SLA compliance.

The third plot (Figure 5) captures the trade-off between SLA settlement time and transaction cost per transaction for different blockchain configurations (varying batch size and endorsement threshold). Results show that smaller batch sizes lead to faster settlements but incur slightly higher gas costs, while larger batch sizes reduce costs but increase settlement delays. The highlighted curve shows the optimal trade-offs, where no configuration is strictly better in both dimensions. Compared to ERP-only (baseline), blockchain-enhanced ERP consistently achieves both lower settlement times and more predictable transaction costs, enabling organizations to balance performance against operational expense.

The results demonstrate clear improvements when ERP is integrated with blockchain. For ART, the ERP-only system showed a median of 174 minutes (IQR 122), whereas ERP with BC reduced this to 105 minutes (IQR 38), highlighting a more predictable and efficient retrieval process. Similarly, SST decreased from a median of 12.4 hours (IQR 4.2) under ERP-only to 7.9 hours (IQR 2.8) with ERP with BC, representing a significant reduction in settlement delays. DCT also improved, with ERP-only averaging 9.1 hours (IQR 3.7) compared to 5.6 hours (IQR 1.9) for ERP with BC, showing faster resolution of contractual conflicts.

The CDFs further confirm these improvements, with ERP with BC converging to near-100% task completion significantly earlier than ERP-only, reducing heavy-tailed execution times. The trade-off analysis between SST and transaction cost showed that ERP with BC achieves better efficiency–cost balance, with non-dominated (Pareto-optimal) points concentrated in the lower-latency, lower-variance region.

Overall, the quantitative evidence confirms that blockchain integration reduces variability and improves performance consistency across key maintenance KPIs, while also offering enterprises flexibility in balancing transaction cost against SLA responsiveness. The provenance protocol further ensures transparency in spare part tracking, creating trust across multiple stakeholders. However, the present framework has been evaluated primarily for maintenance workflows in isolation. In practice, ERP maintenance activities are closely linked with MM module, where spare parts, procurement orders, and vendor contracts must align with preventive and predictive maintenance schedules. As future work, the framework will be extended to incorporate PM–MM integration. This will enable smart contracts to automatically trigger spare part requisitions, validate supplier provenance, and execute purchase orders in sync with maintenance tasks. Such integration is expected to provide a closed-loop blockchain-based ERP ecosystem that unifies maintenance and supply chain operations, ensuring both operational efficiency and supply chain integrity.

6. Conclusion

This paper proposed a blockchain-enabled ERP framework to enhance maintenance and engineering processes with automation, transparency, and trust. Smart contracts deployed on Ethereum Remix IDE were used to enforce SLAs, track provenance of spare parts, and record maintenance activities, while simulation results analyzed in Python confirmed substantial improvements over ERP-only systems. Key metrics such as Audit Retrieval Time (ART), SLA Settlement Time (SST), and Dispute Cycle Time (DCT) showed faster, more predictable performance under blockchain integration.

Although transaction costs were slightly higher due to consensus overhead, the gains in compliance, traceability, and efficiency outweighed these limitations. The findings highlight blockchain's role as a trust layer in ERP, extending its value beyond internal systems to multi-stakeholder collaboration. Future work will focus on ERP Plant Maintenance

(PM) and Materials Management (MM) integration, enabling automated spare part procurement and predictive maintenance within a unified blockchain-ERP ecosystem.

References

Alabadi, A., Muhammad, A., & Al-Dabbagh, R., Next-Generation Predictive Maintenance: Decentralized IoT and Blockchain Framework with IPFS Storage, Frontiers in Artificial Intelligence, vol. 6, 2023, https://doi.org/10.3389/frai.2023.1077384.

Alqaryuti, A., Moawad, K., Salah, K., Mayyas, A., & Jayaraman, R., Blockchain-driven framework for preventive maintenance management of aircraft hydraulic systems - discover internet of things, SpringerLink, April 28, 2025, https://link.springer.com/article/10.1007/s43926-025-00149-x.

Al-Refaie, A., Al-Hawadi, A., & Lepkova, N., Blockchain design with Optimal Maintenance Planning, MDPI, Nov. 6, 2022, https://www.mdpi.com/2075-5309/12/11/1902.

Aslam, T., et al., Blockchain Based Enhanced ERP Transaction Integrity Architecture and PoET Consensus, Computers, Materials & Continua, vol. 70, no. 1, pp. 151–168, 2021, https://doi.org/10.32604/cmc.2021.044407.

Halder, P. N., An comprehensive guide to SAP modules: Delving into the modern SAP Ecosystem, Medium, April 9, 2023, https://medium.com/@HalderNilimesh/an-comprehensive-guide-to-sap-modules-delving-into-the-modern-sap-ecosystem-a9fb855f1f11.

Hasan, H. R., Salah, K., Jayaraman, R., Ahmad, R. W., Yaqoob, I., & Omar, M., Blockchain-Based Solution for the Traceability of Spare Parts in Manufacturing, IEEE Access, vol. 8, pp. 100308–100322, 2020, https://doi.org/10.1109/ACCESS.2020.2998159.

Ho, S. C., Chua, C. K., & Wong, C. H., Blockchain-Based Aircraft Parts Traceability System for Aviation Maintenance, Expert Systems with Applications, vol. 176, pp. 114–123, 2021, https://doi.org/10.1016/j.eswa.2021.114872.

Johnny, K., & Wales, M., Exploring the Role of Blockchain in Maintenance and Building Condition Auditing, Journal of Construction Research and Innovation, vol. 4, no. 2, pp. 45–61, 2024.

Li, M., Weng, J., Yang, A., Lu, W., Zhang, Y., Hou, L., Liu, J., Xiang, Y., & Deng, R. H., CrowdBC: A Blockchain-Based Decentralized Framework for Crowdsourcing, IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 6, pp. 1251–1266, 2019, https://doi.org/10.1109/TPDS.2018.2881735.

López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., & Ponomarev, A., CATERPILLAR: A Blockchain-Based Business Process Execution Engine, arXiv preprint, arXiv:1808.03517, 2018.

Moalagh, M., & Ebrahimi Ghadi, A., Blockchain-based ERP system: Architecture and opportunities for future, Journal of Information Technology Management, vol. 14, no. 3, pp. 75–92, 2022, https://jitm.ut.ac.ir/article_87849.html.

Seraj, R., Smart Contract-Based Transport Infrastructure Maintenance Using Blockchain, arXiv preprint, arXiv:2410.20431, 2024.

Tarannum, T., & Zghair, H., Development and Implementation of a Standardized Supplier Certification and Validation Framework for Enhanced Pricing, Lead Time, and On-Time Delivery Performance, In: 9th North American Conference on Industrial Engineering and Operations Management, June 2024, https://doi.org/10.46254/NA09.20240090.

Zhang, Y., Wu, Q., & Zhou, L., ChainSplitter: Towards Blockchain-Based Industrial IoT Architecture, IEEE Access, vol. 7, pp. 157–168, 2019, https://doi.org/10.1109/ACCESS.2019.2956721.

Biographies

Md Didarul Alam is a Graduate Assistant in the Department of Electrical and Computer Engineering at Georgia Southern University, USA. He holds a Bachelor of Science in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET) and is pursuing his Master of Science in Electrical and Computer Engineering. He has over ten years of professional experience in manufacturing, automation, and process optimization, having served in key engineering roles at Berger Paints Bangladesh Ltd., Nestlé Bangladesh Ltd., and Abul Khair Steel Industries Ltd. His current research focuses on Electrical Impedance Myography (EIM), AI-driven diagnostics, and electromagnetic simulation. His research interests include electromagnetics, industrial automation, control systems, and sustainability engineering.

Tasfia Tarannum graduated from North South University with a Bachelor of Science in Electrical Engineering in 2014. She completed her Master of Science in Business Analytics and Supply Chain Management Certification. She has been an experienced professional in Engineering and Supply Chain Management for almost seven years. She worked on sustainability performance management and in continuous improvement in the context of FMCG and

supply chain management. Her research interests include supply chain management sustainability, service operations management, continuous improvement, as well as business innovations, vendor management, industrial sustainability, industry management, optimization, reliability, and manufacturing resilience among others.

Hayder Zghair is an assistant professor of Industrial Engineering in the Department of Engineering and Physics, and the director of Industrial Engineering Development in the College of Science and Engineering at Southern Arkansas University. Dr. Zghair is chairing and member of several committees at the department, college, and university levels. Before joining SAU, served in several academic appointments including Assistant Teaching Professor of Industrial and Manufacturing Engineering at Pennsylvania State University and lecturer of Industrial and Manufacturing Engineering at Kettering University. He completed a B.S. and an M.S. from the University of Technology, where he majored in Industrial and Production Engineering. Dr. Zghair earned his second master's degree in Manufacturing Systems Engineering and Doctor's Degree in Manufacturing Systems Engineering from Lawrence Technological University, Michigan, USA. He has several professional affiliations: Affiliate Researcher at Penn State Institute of Energy and the Environment and Member of Directors Board of the Environmental Engineering Division (EED) at American Society of Engineering Education (ASEE). Dr. Zghair's research interest covers a wide area of industrial and manufacturing engineering including Flexible-Automated Manufacturing, Robotics, Human Factors and Ergonomics, Sustainability Engineering, Analytical Modeling & Simulation, and Optimization, and has published journal and conference papers. Dr. Zghair continues to explore new and cutting edges areas for research and education in engineering disciplines and got recently had an extensive interest in industry 4.0 and 5.0 tech such as AI, IIoT, smart sensors, and cloud computation and control. He is a professional member of ASEE, Institute for Operations Research and Management Sciences (INFORMS), Society of Automotive Engineers (SAE), and Industrial Engineering and Operations Management Society (IEOM). Dr. Zghir is the faculty advisor of the SAE students' chapter and SAU Baja and IEOM students' chapter at SAU.