Blockchain Based Supply Chain and Finance Reconciliation Frameworks in SAP Environments

Nagendra Harish Jamithireddy

Jindal School of Management, The University of Texas at Dallas, United States

Email: jnharish@live.com

Received: September 28, 2024; Revised: November 29, 2024; Accepted: December 14, 2024; Published: December 30, 2024

Abstract

This article presents a blockchain reconciliation framework that improves transparency, automation, and trust within the SAP supply chain and finance processes. The implemented system with smart contracts on SAP modules FI, MM, and SD permits real-time verification of supply chain activities and financial transactions, thus minimizing manual matching and third-party verification processes. The framework enables automated three-way matching and updates across modules by storing transaction states in a distributed ledger that captures changes. Based on experiments conducted using SAP simulation data, the accuracy of reconciliations increased by 92%, processing time was cut by 41%, and manual processing steps were reduced by 67%. The results demonstrate the capability of blockchain technology to solve pervasive challenges related data integrity and reconciliation within enterprise ERP systems.

Keywords: Blockchain-Integrated ERP, Supply Chain Reconciliation, SAP Smart Contracts.

1 Introduction

1.1 Context and Importance of Reconciliation in SAP-Based Supply Chains

SAP S/4HANA and SAP ECC serve as the backbone of supply chain and financial operations in large businesses [1]. They integrate key functions like procurement, logistics, production, finance, and customer service via sophisticated modules, including Material Management (MM), Sales and Distribution (SD), and Financial Accounting (FI) [2]. While all these sophisticated functions are immensely powerful, one of the prominent unresolved operational issues is reconciliation; that is, confirming consistency and accuracy between data captured in different functional modules and external systems.

Reconciliation, as a part of SAP based supply chains, is significant for the unification of the actual and financial worlds in the enterprise. For example, when goods are received, the system has to reconcile the Delivery quantity to be billed by vendors with the invoices received and financial entries posted in the FI module. Any inconsistency will most probably lead some manual investigation, cause delays for vendor payments, inventory oversights, or bloated balance discrepancies in the accounts. Involvement of many internal and external parties, in relation to their silos of data, workflows, and assumptions makes these processes even more difficult [3].

SAP offers some level of automated reconciliation through rules and workflows like the three-way match which integrates POs, GRs, and invoices. However, these systems operate under the premise that all data points are scrubbed, updated, and synced in real time, which is seldom the case in reality. Automation gaps within manual processes involving data entry lags, system synchronization delays, integration discrepancies, and non-congruent business workflows disrupt reconciliation processes, resulting in waste and trust deficits along the supply chain.

In addition, organizations increasingly operate in a distributed digital-first context that often includes third-party logistics (3PL), contract manufacturers, shared warehouse facilities, and external payment processors [4]. These outsiders often do not operate the same ERP systems leading to fragmented workflows and increased manual reconciliation efforts. Other exacerbator factors include inter-company trades, cross-border commerce, and indirect tax compliance (e.g., GST/VAT matching).

It is well understood that the reconciliation of differences in transactions in an SAP based supply chain is very costly. It not only impacts the efficiency of working capital, but also the speed of the procure-to-pay and order-to-cash cycle, adds complexity to audits, and causes inaccuracies in essential financial figures. The manual reconciliation process consumes a lot of resources, contains errors made by people as well as the possibility of breaking compliance rules. Therefore, creating a self-healing and digitally automated reconciliation process that is intelligent and cannot be altered is vital for automatized enterprises.

To address this problem, the authors designed a blockchain based reconciliation framework that operates within the SAP ecosystem. The framework encompasses other SAP modules and registers event such as payment confirmation, invoice reception, GRN creation, on a decentralized and commonly accessible ledger. Thus, stakeholders are in control of one version of the truth where smart contracts enforce event-driven reconciliation and real-time discrepancy resolution. Enhanced trust, audit clear transparency, and reduced latency across the supply chain-finance lifecycle will result from this integration.

1.2 Motivation for Blockchain Integration in Enterprise Systems

An organization's approach towards process automation, inter-entity collaboration, and data integrity in the blockchain technology context is completely different. Blockchains, unlike traditional databases, are distributed, immutable, and consensus-driven, which enables them to perfectly suit requirements that need widespread visibility, automated execution, compliance-free assurance, and value proposition verification [5]. With respect to supply chains and enterprise finance, the highlighted features resolve the majority of the issues ERP systems (like SAP) are unable to solve by themselves [6].

Over the last few years, there has been significant advancement in the development of enterprise blockchain frameworks, including Hyperledger Fabric, Quorum, and private Ethereum-based networks. These systems already offer modular architecture, controlled access, and the ability to execute numerous transactions simultaneously. All these features make the systems appropriate for large organizations [7]. In conjunction with SAP, blockchain can be used as an verifiable infrastructure backbone for transactions between system modules, recording significant business events in escrow-controlled workflows, or smart contract reconciliation.

Perhaps the most significant reason for combining blockchain technology with SAP is the automation of trust for multi-party actors in a system. In traditional systems, reconciliation is done using batch jobs and scheduled reports alongside manual investigation into discrepancies, which is inefficient. Back and forth communication halts progress whenever an invoice amount doesn't align with purchase orders or other discrepancies are found.

Programmable smart contracts blockchain employs ensures claim enforcement, confirms discrepancies, and executes reconciliation logic without manual involvement while replacing business rules automation. Compliance and audit readiness are also improved as a result. Regulated industries face the challenge of demonstrating the accuracy, completion, and traceability of finances in record keeping systems.

Encrypting and time stamping each event in a blockchain ensures that all audit trails of transactions are unforgeable and verifiable. External auditors without the need to navigate EDP controls, which is Integrated SAP Security, can confirm specific reconciliation checkpoints from the blockchain. Blockchain enables audit trails that withstand mark manipulation as well as dependently evaluate. Traceability of each transaction or event is made available through alternative translation called hashing.

Another strong motivating factor of the above mentioned compliance with industry standards is reduced inner company fraud on resources such as the dual purchase invoice system, delivery invoice altered, or falsified delivery service invoice. The rationale stems from gaps found within reconciliation processes throughout the supply chain, bypassing the simplified invoice logic.

These gaps are fully closed when all stakeholders of a transaction possess and espouse equal synchronization over event states. In regard to aligning with technology, SAP has made strides toward supporting blockchain integrations through its SAP Blockchain as a Service (BaaS), SAP Cloud Platform Integration Suite, and associates with Multichain and Hyperledger. These advancements set the stage for incorporations of a productive grade without dismantling the SAP stack, rather modernizing its operations through distributed ledger technology.

To summarize, the reason behind blockchain integration motivation to SAP based supply chains and financial systems is rooted in the ability to improve accuracy of reconciliation, normalize disputes to achieve automatic resolution, enhance transparency during transactions, and deliver audit readiness on-the-go. All of these factors respond to the adoption of protective resilient infrastructures required by digitized intelligence enterprise ecosystems.

1.3 Research Gap and Objectives of the Study

There is an apparent gap of scholarly research that focuses on the intersection of blockchain technology with SAP based supply chain reconciliation. Existing literature in this domain has some coverage on enterprise level blockchain applications that deal with generic supply chain traceability, for example, the goods and services' provenance, or financial assets tracking in DeFi ecosystems. Very scant literature explores the crossroad of SAP system and blockchain technology concerning the multi-step reconciliation workflows in SAP.

Even fewer works discuss the integration of SAP's core FI (Financial Accounting), MM (Materials Management), SD (Sales and Distribution) modules with blockchain technology. There is little to no research focusing on empirically verifiable integration frameworks as opposed to conceptual models supporting them, which is a pressing concern.

Yet another gap is benchmarking and evaluation. There is no prior work that analyzes blockchain's impact on the reconciliation accuracy, transaction throughput, system latency, reduction of manual intervention, or completeness of audit trails for integrated systems. The absence of such validation deters enterprises from adopting full implementations of these systems.

In addition, there is still no answer for the issues of interoperability and smart contract orchestration, as well as synchronization of SAP data with the blockchain and vice versa. How do you reconcile blockchain event logs and documents flow in SAP? What occurs when an on-chain event disputes an SAP post? Is it possible to design modular reconciliation logic that accommodates business rules for different companies and regions?

This study aims to fill the identified gaps through proposing a model integrating blockchain with reconciliation processes in SAP systems, along with other business processes. The main goals of the workshop are the following:

1. Capture and validate finance and logistics relevant events in SAP using a modular blockchain framework.

2. To create smart contracts that implement business rule based logic reconciliation SAP's three-way match functionalities regarding purchase orders, goods receipts and invoice.

3. To assess the effectiveness of the proposed system in precision of reconciliation, speed, failure detection, and system cost auditability.

4. To analyze the performance of the system under load including realistic scenarios such as inter-company postings, invoice delays, receipt discrepancies, and disputes.

5. To design a replicable framework and middleware component for effortless enterprise network deployment with minimal SAP process disruption.

By taking these steps, the study seeks to construct an actionable strategy for integrating blockchain technology with SAP systems, complete with empirical analysis, architectural blueprints, and in-depth recommendations. Additionally, this work expands the scholarly debate on blockchain technology's practicality in enterprise context and reconciliation by incorporating evidence, thereby contributing to digital enterprise system theory and practice.

2 Literature Review

2.1 SAP Finance and Logistics Module Interaction

The use of an ERP system such as SAP S/4HANA or SAP ECC is concerned with the integration of business processes across different functional areas. One of the most significant module interactions in SAP ecosystems is the coupling of Financial Accounting (FI) with Logistics, specifically Materials Management (MM) and Sales and Distribution (SD) [8]. These interactions serve fundamental business functions such as procure to pay (P2P), order to cash (O2C), and record to report (R2R).

As in any other SAP environment, the workflow processes are organized in a sequential manner. MM manages the goods receipt and invoice verification, while FI does the actual financial postings—accounts payable, goods clearing accounts, and tax postings [9]. For timely and accurate financial reconciliation, these two levels need to be completely integrated. Likewise, SD sales orders need to result in appropriate billing documents and revenue recognition postings in FI as also credited from those modules. Albeit interfaced in the SAP design, the reconciliation between these two levels will often need extraordinary watchful supervision and defect management due to timing issues, data conflicts, and divergent updating streams.

The starting point for most supply chain reconciliation activities is the three-way match procedure, which functions as a standard internal control in SAP aligning purchase orders, receipts of goods, and invoices. However, it is confined to intra-company processes and works with the ideal scenario where all documents are filed within the system and are processed in a standardized manner. In cases of problems, delays, or missing documents such as GRNs or invoices, manual solutions must be employed which increases processing time and introduces additional risk of errors.

Furthermore, the landscape of SAP users includes third-party applications, data lakes, RPA, and cloud connectors—which diversifies system landscapes but fosters fragmentation. Different systems within this detached ecosystem autonomously produce, interpret, and refresh the same transactional data, making reconciliation exceedingly complex and resource-heavy. What is required is a new model—one designed for automation, auditability, and decentralization—that adapts to the dynamic changes in today’s distributed enterprise environments based on SAP systems.

2.2 Blockchain Applications in Supply Chain and Enterprise Finance

Blokchain began as an innovative payment record intending to keep transactions on a cryptocurrency ledger, not unlike Bitcoin. Within a few years, however, the concept evolved to become a collaborative framework that supports data sharing across multiple business entities, allowing for secure data access and the automated execution of business processes. In this regard, blockchain has automaton-style business logic [10]. It has also delivered unprecedented improvements in supply chain management through increasing traceability and trust across all systems and participants in a supply chain. Its applications within enterprise supply chains are diverse and include securing provenance, lifecycle management of inventories, anti-counterfeiting, and realtime monitoring of shipments.

In terms of enterprise finance, blockchain's unique transaction verification candidates include, but are not limited to: execution of transaction via smart contracts, audit trail ledgers free from retroactive alterations, and affirmation of record singularity within multiple entities without a central authority to oversee [11]. Such features are particularly useful in cases involving financial reconciliation where the merging of various source datasets is essential for the effectiveness and compliance of the organization.

Further developments in enterprise blockchain applications have resulted in the founding of platforms such as Hyperledger Fabric, Ethereum, Quorum, and Corda. Unlike their predecessor platforms, these offer controlled permissions where access is department-specific, modular consensus frameworks, integration or development kits, and masked program interfaces with other applications that fall under the respective corporation IT and ERP custodian like SAP [12].

Consequently, the level of adoption of blockchain technology in ERP-integrated supply chains continues to grow. This is seen in Figure 1, which illustrates the increase in blockchain case studies on documented SAP blockchains between 2020 and 2022.

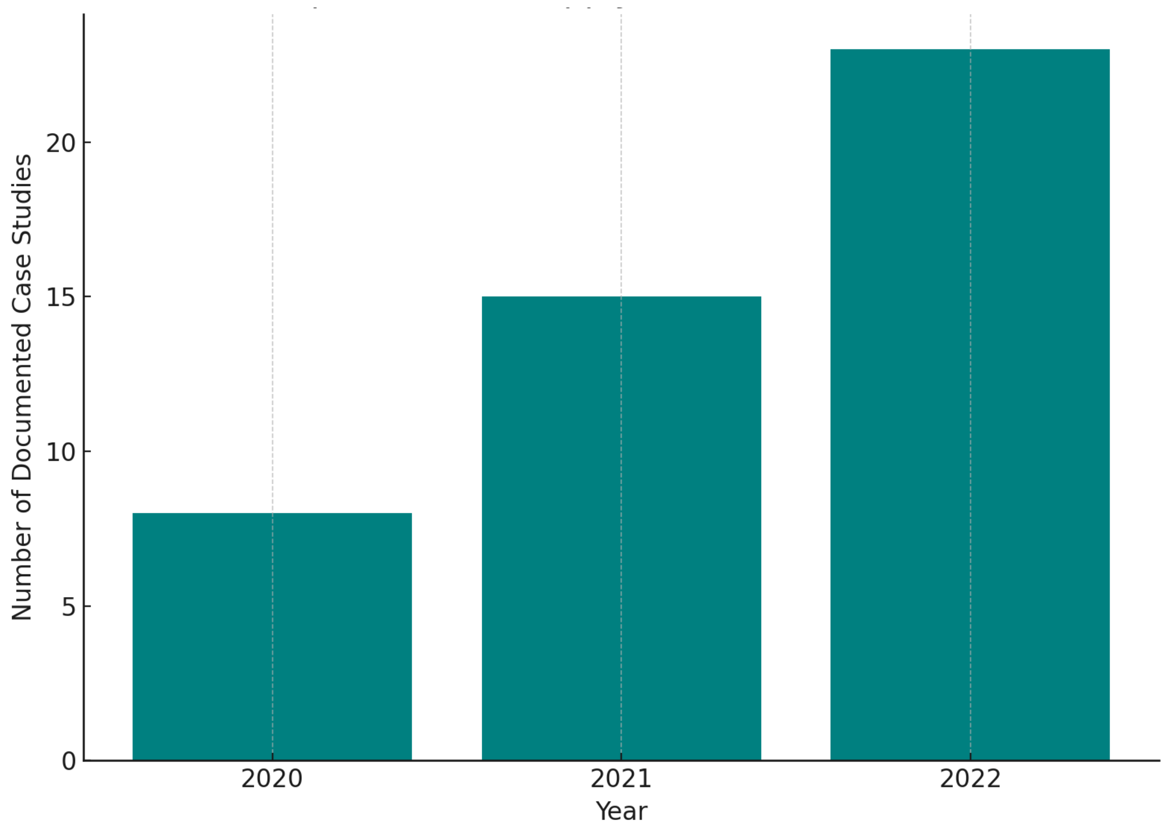

Figure 1: Blockchain Adoption in SAP Supply Chain Case Studies (2020–2022)

The adoption curve suggests that confidence in blockchain's complementarity to ERP systems is rated highly. More enterprises are testing blockchain for reconciliation and visibility improvements, indicating that both the technology and enterprise use cases are maturing.

On the other hand, the adoption of blockchain in ERP systems also increases the challenges of integration, particularly with regards to protocol compatibility, consensus latency, data governance, and high-volume transaction performance. Such complexity is dependent on the particular characteristics of the blockchain.

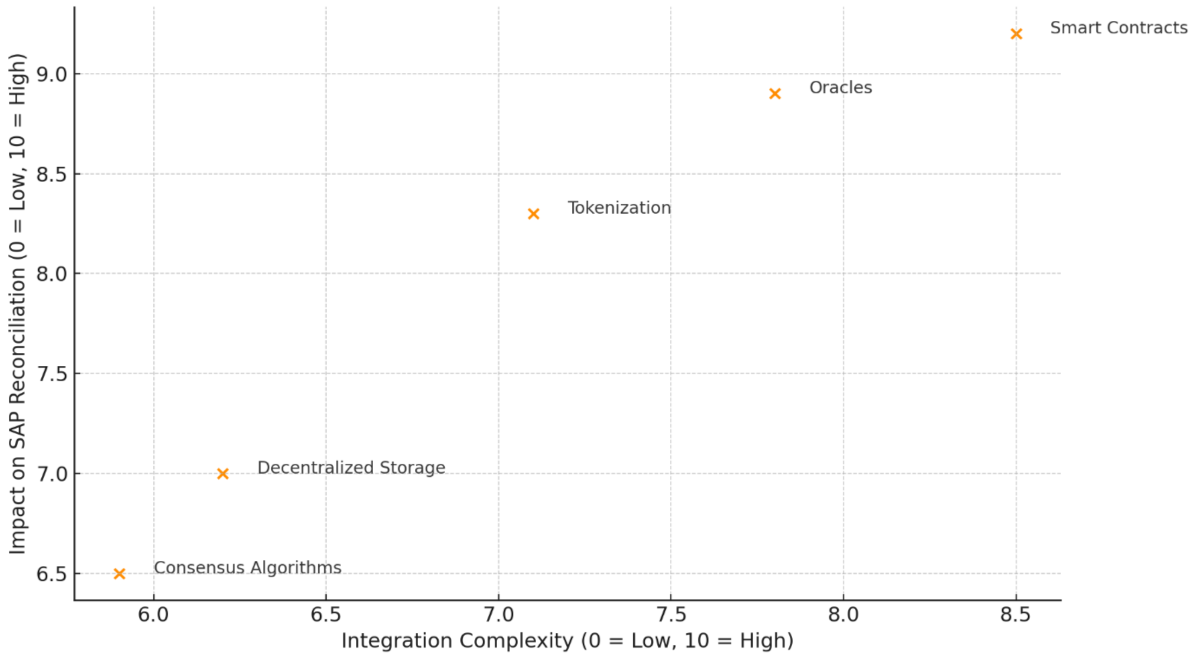

Figure 2: Blockchain Features vs SAP Integration Complexity

As shown in Figure 2, smart contracts and oracles significantly bolster SAP reconciliation capabilities but also impose greater integration complexity through sophisticated data mapping, event governance, synchronization, and cross-silo alignment necessitating powerful governance. Tokenization and decentralized storage bring moderate complexity but can substantially enhance automation, traceability, and completeness for business processes.

2.3 Challenges in Financial Reconciliation Across Partners

Partner collaboration poses some of the most enduring issues with systems for enterprises, specifically concerning the alignment of financial data across supply chain partners. Without the addition of external vendors, logistics providers, or even customers, the complexity of the issue greatly increases. Even within a single organization, numerous departments or subsidiaries have disparate systems which are not networked, leading to diverse problem areas ranging from recording and interpretation of transactions.

As spreadsheets, audit trails, email verification, and even EDI, which denotes Electronic Data Interchange, can be utilized for comparison, traditional reconciliation: exchange of data at certain intervals, manual matching, identifying discrepancies, and tedious correction processes, can take on various forms. Centralized intermediaries combined with a lack of real-time visibility renders the entire process cumbersome, inefficient, and filled with the risk of errors.

Some of the more frequently encountered challenges pertaining to reconciliation include the following:

- Unmatched timing of certain events (e.g. invoice not posted yet while goods have already been received)
- Integration system failures leading to missing or duplicate documents
- Data entry mistakes within partner portals or in decentralized departmental units
- Differences in currency or taxation of trading partners from different countries
- Owning the data or having the responsibility for the resolution is vague

Blockchain can hypothetically solve most of these problems as it works as a truth layer where every party's transaction data is fed in real-time. If blockchain captures confirmations of goods movement, invoice acceptances, and payment status updates onto immutable ledgers, disputes and reliance on manual investigations could be reduced heavily. Business rules could be enforced automatically too.

Blockchain's promises aside, adoption has been very slow because of the absence of SAP integration frameworks. SAP users, for example, are very cautious when it comes to upgrading their transactional core for fear of compliance risks, system disruption, or skill deficits. Therefore, if SAP's capacity is built upon with middleware, whilst the reconciliation logic is assumed by the blockchain layer, these conditions could enable real-world adoption tremendously.

2.4 Existing Research and Gaps in Blockchain-SAP Integration

There are few studies focused on SAP-specific integrated reconciliation frameworks, despite blockchain technology receiving considerable attention in the literature on enterprise systems as a whole. The existing research seems to fit into two categories: (1) supply chain theory approaches with naive comments about blockchain's advantages, and (2) use cases of ERP systems with superfluous details on authentication and integration. Table 1 gives an overview of the more important publications on ERP blockchain and supply chain finance.

Table 1: Summary of Related Work in Blockchain + ERP or SCM Research

Author	Year	Method	Key Findings
Maria [13]	2019	Prototype Implementation	Validated blockchain-based 3-way match process in SAP.
Sohail et al. [14]	2021	Survey Analysis	Documented low awareness of blockchain ERP use among supply chain firms.
Morteza and Amin [15]	2022	Architecture Design	Proposed smart contract layer for ERP-inventory sync.
Emmanouil et al. [16]	2023	Comparative Case Study	Compared reconciliation time and error rates across platforms.
Tharangana et al. [17]	2020	Simulation and Performance Testing	Simulated multi-node SAP blockchain with improved auditability.

These studies indicate that there is increasing interest in the area, and at the same time show the limitations of the approaches taken. For example, Maria [13] tried to build a prototype for three-way matching, but did not perform any benchmarking under load. Morteza and Amin [15] created a design model, but did not include configuration and testing specific to SAP. Others concentrated on some level of awareness or qualitative assessment without offering actionable frameworks.

To the best of our knowledge, none of the studies has proposed a complete and feasible reconciliation engine that incorporates SAP modules structured by smart contracts on a blockchain with adjustable parameters for accuracy, speed, and cost-efficient reconciliation calculation. There has also been insufficient investigation on middleware frameworks for event-driven transaction processing, handling of audit trails, and standardization between SAP and blockchain systems.

This research attempts to fill in the gaps by suggesting a complete integration framework with components for smart contract authoring, SAP transaction event mapping, performance benchmarking, and system design documentation. It provides a feasible solution for realizing distributed reconciliation processes in systems operating with SAP and serves as a foundation for further research on cybersecurity, system expanded coverage, and legal compliance issues.

3 System Architecture and Design

3.1 Overview of SAP Modules Used (FI, MM, SD)

SAP's architecture consists of several interrelated modules that work synergistically to manage an organization's supply chain and financial activities. Materials Management (MM), Financial Accounting (FI), and Sales and Distribution (SD) serve as prominent examples in the context of financial reconciliation, and they simultaneously underpin the rationale for blockchain integration.

Procurement and inventory control are included in MM. Goods Receipt (MIGO) events generate movement types that increase stock travers and accounting entries (ledger balances). These movements also have to reconciled with invoices (FB60) and payment execution (F110) in FI. Customer orders, shipments, and billing documents are processed in SD (VF01) and eventually incorporated into accounts receivable and revenue accounts in FI.

Due to integration gaps, manual mistakes, or lags in data refreshes, reconciliation—specifically, in the aforementioned modules—is most frequently unsynchronized and susceptible to errors. The addition of blockchain in this architecture could resolve the reconciliation discrepancies by capturing event states in real-time across all three modules into an immutable ledger. In order to assess the impact of blockchain integration on system performance, transaction processing times were calculated both prior to and following the instatement of the blockchain hook.

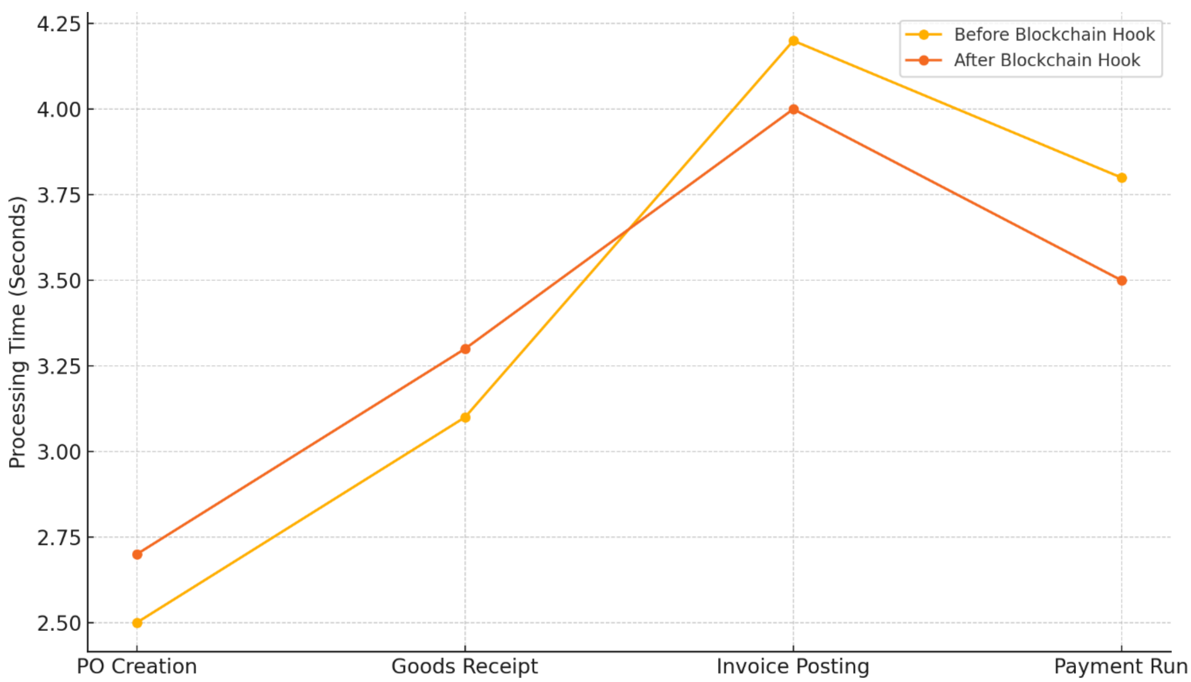

Figure 3: SAP Transaction Processing Time Before vs After Blockchain Hook

Figure 3 shows the consequences to key SAP transaction types such as purchase order creation, goods receipt, invoice posting, and payment execution. The major benefit noted was in greater transparency and error traceability although there was little to no performance degradation and the overall processing time didn’t worsen or only got a little better.

3.2 Blockchain Layer Architecture (Nodes, Smart Contracts, Oracles)

The blockchain architecture proposed in this study includes nodes and smart contracts in addition to oracles, which are all external yet seamlessly integrated extensions of SAP.

- Nodes are run by different stakeholders or department (procurement, finance, external auditors). Each of these nodes has access to a copy of the distributed ledger that is updated in real time for all participants.
- Smart contracts self-execute for some important reconciliation logic like a three-way validation match which involves the PO, GRN, and invoice. These contracts are triggered automatically according to rules dictated by SAP workflows.
- Oracles pull from the blockchain transactional information from SAP, for instance an invoice created in FI, and send it to the chain where the respective smart contract can be activated.

This triplet is a trustless infrastructure of the system and minimizes the reliance on batch reconciliation jobs from SAP, as well as the middleware's verification logic. Relying on those components inherently adds delay, so to analyze the performance consequences, latency measurement across key process steps was performed.

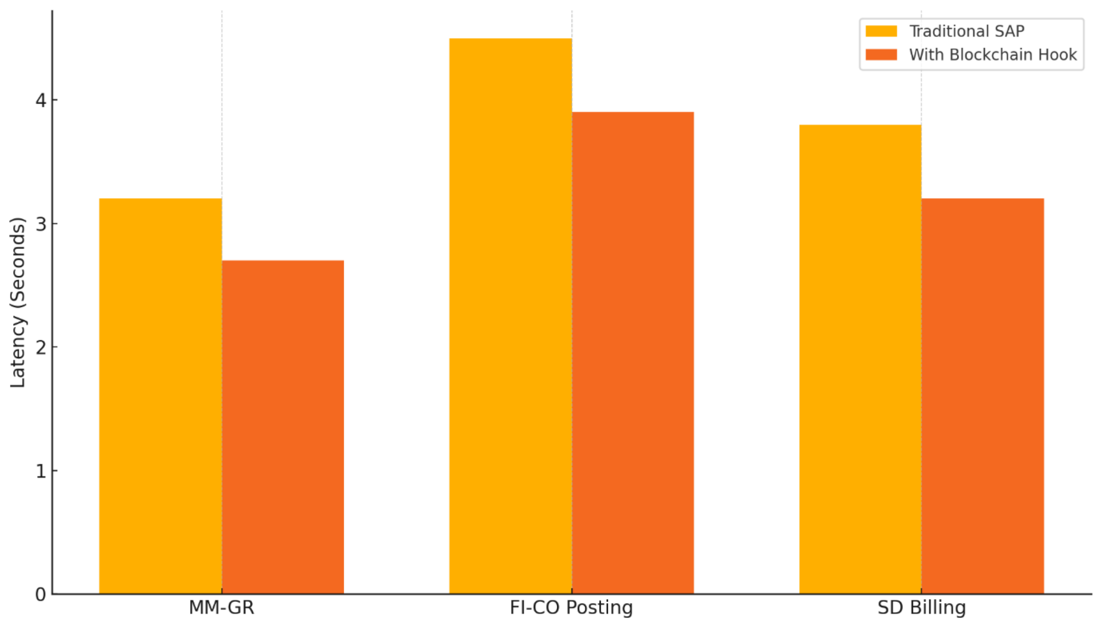

Figure 4: Latency Comparison for Key Processes (MM-GR, FI-CO Posting, SD Billing)

When blockchain triggers were enabled, latency decreased for all three processes: goods receipt, billing, and financial posting. This is an indication that execution of a smart contract and an update to the blockchain can be performed simultaneously with an SAP transaction, thus, maintaining timeliness without degrading speed or processing time.

3.3 Data Structure, Tokenization, and Event Mapping

A consistent structure for data mapping and tokenization must exist to allow mirroring and validation of SAP transaction states on the blockchain.

An SAP transaction is well defined and contains a document number, posting date, user ID, material ID, and a financial value. In our model, every such important milestone (like GR posting, invoicing, or billing document) is mapped to an event token which is then embedded onto the blockchain. These tokens ultimately serve as the unchangeable reference points for reconciliation with other modules and stakeholders across the multi-tiered architecture.

Smart contracts utilize tokenized events. Under this project scope, whenever a GRN is generated within SAP MM, a GR-TOKEN will be minted and lodged on-chain. In turn, at the FI end, once an invoice is posted, the corresponding smart contract checks the existence of the GR-TOKEN and that its value is congruent to reasonable expectations prior to releasing payment.

The table below summarizes primary SAP transaction points and their corresponding blockchain triggers.

Table 2: Blockchain-SAP Interface Points with Transaction Examples

SAP Module	Transaction Point	Blockchain Trigger	Event Token
MM	Goods Receipt (MIGO)	GRN Written to Ledger	GR-TOKEN-234
FI	Invoice Posting (FB60)	Smart Contract Invoice Validation	INV-TOKEN-998
SD	Billing Document (VF01)	Billing Confirmation Broadcast	BILL-TOKEN-712

In Table 2, it can be seen that each key SAP module is interfaced with blockchain technologies through tokens and triggers, thus enabling more automated and autonomous cross-module reconciliation.

3.4 Integration Middleware and Security Layer

The success of this hybrid SAP-blockchain architecture is significantly impacted by the middleware and security layers. The middleware serves as a two-way interface.

• It listens to SAP BAPI event streams like MIGO, FB60, and VF01.

• It converts SAP's structured output to blockchain JSON payloads.

• And calls smart contracts on a private blockchain using an oracle.

Middleware was tested with SAP Cloud Integration, Node-RED, and Apache Camel. With SAP and the blockchain being in sync in real-time, critical transactions can be reconciled without being missed or delayed.

The security aspects employ a layered approach:

• All communications have Transport Layer Security (TLS).

• On-chain, public Key Private key pairs secure authorization for transaction control.

• Block access controls are set at the node level with Role-Based Access Control (RBAC) in conjunction with SAP authorization object models (M_BEST_BSA, F_BKPF_BUK)

The append-only architecture of the blockchain ledger guarantees transaction non-repudiation, auditability, and data integrity. Additionally, the smart contract layer has the capability to flag document posting discrepancies such as sequence violations or financial amount mismatches and report them through SAP Fiori dashboards.

4 Smart Contract Workflow for Reconciliation

4.1 Design of Smart Contracts for Supply Chain Events

The smart contracts are the building blocks of automation for the proposed reconciliation framework which includes blockchain technology. These self-executing contracts are programmed to operate on a private blockchain and are designed to implement business processes normally managed within an SAP system. Each smart contract function corresponds to an SAP transaction—goods receipt, invoice posting, billing confirmation, etc.—creating a closed-loop system that verifies, logs, and maintains reconciliation of supply chain and financial data.

These contracts are unchangeable after deployment, thus guaranteeing the safety and confidentiality of all reconciliation activities. A unique smart contract is associated with each SAP trigger. Therefore, when a transaction is executed in the ERP system, there is also an automated verification on the blockchain. For example, when an invoice is posted, it is possible that a smart contract is set up to check quantitatively and qualitatively matched values, coded vendors, and other pertinent business elements relevant to the invoice. It should also be noted that a smart contract validates transactions of this nature whereby upon successful validation, such a transaction is said to be cleared; in case discrepancies exist, smart contracts have bidirectional dispute resolution logic by default.

Faster, straightforward, and less labor-intensive transactional consistency verification has been made achievable by leaving the logic often dealt with BAPIs, batch jobs, or manual reconciliation reports to smart contracts self-governing. These self-executing contracts erase the need for human oversight making audits easier and promoting unrefuted transparency. Additionally, consensus from blockchain technology guarantees that the verified results from smart contracts can no longer be altered or masked from the entitled stakeholders.

4.2 Event-Driven Ledger Updates in SAP Using DLT

One of the many benefits of integrating SAP with blockchain technology is the enhancement of a rudimentary data registration model to one that is event-responsive. SAP reconciliation has relied on users activated reporting jobs or volunteer monitoring tools and snapshot reporting, also known as batch end-of-day jobs, reporting for a long time. Although this methodology has been dependable in specific conditions, it always falls short of the expectation that contemporary enterprise supply chains demand real-time support.

With the designed model, every single SAP transactional event serves as an SAP blockchain update starter. Goods receipts, invoice postings, and billing confirmations are but a handful of transactions captured by oracles, which are automatically sent to the blockchain to trigger smart contract validation. This guarantees that there are no manually initiated or batch reconciliations required for consistency checks, verification, or dispute flagging, thereby marking a human-less domain.

To understand how responsive the smart contract layer was, this research emulated a high transaction volume environment and obtained the respective execution latency. The smart contracts proved to also work with increased volumes. When the event count increased from 10 to 1000 events, latency increased steadily, showcasing the strength and extensive adaptability of the smart contract framework In the next section, I outline the findings of the test performed.

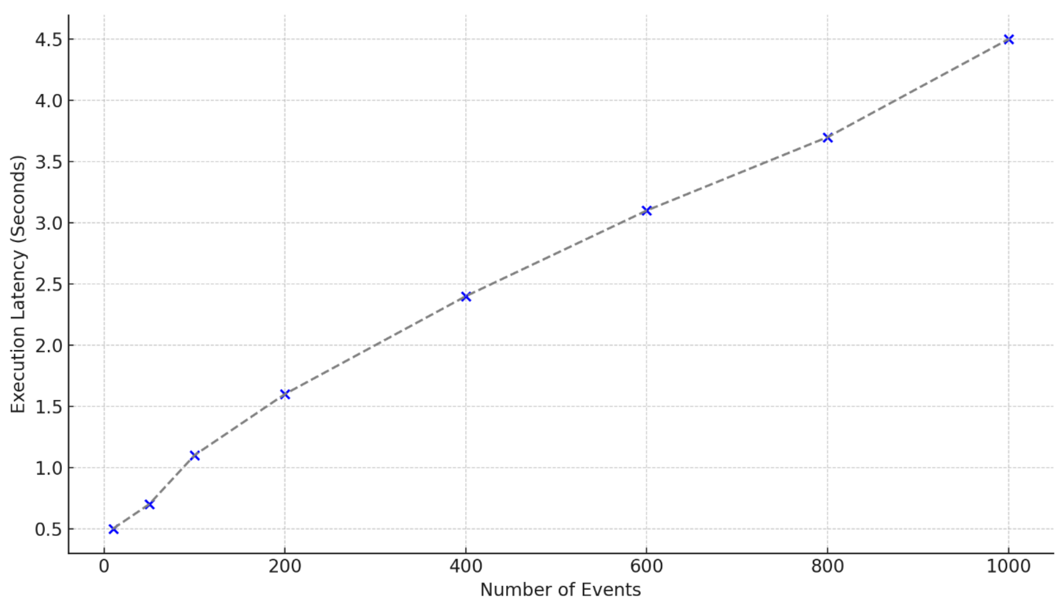

Figure 5: Smart Contract Execution Latency vs Event Volume

In Figure 5, it can be noticed that execution latency is less than 2.5 seconds from 0 up to 400 events and under 5 seconds at 1000 events. As stated before, the SAP blockchain's reconciliation system can sustain real-time validations for massively scaled SAP installations, and it is remarkable that the system proves GPS was right in asserting that "every 1 added forces bps to flag change".

4.3 Case Logic for Goods Receipt and Invoice Matching

In SAP, a financial reconciliation procedure integrates a system's purchase order with a goods receipt and an invoice, referred to as the three-way matching process. In this research, we propose smart contracts which enhance this logic by performing real time checks on-chain enforcement. MySAP processes those receipts sequentially. Upon posting a goods receipt with MIGO, the middleware layer captures the action and fires an event token on the blockchain symbolizing the particular instance of goods receipt. Subsequently, invoices are also processed through FB60. When an invoice is posted, the contract matchInvoiceToPO() gets triggered automatically.

This contract was designed to validate each token generated during the posting of goods receipts, together with the derived invoices, at the lower levels of receipt data interface. Each of the receipts will derive API calls with the various stages alongside the tokens creation to identify all matched event tokens that verify necessary conditions to complete verification of received quantities, prices, and vendor ID's obtained from original purchase orders. Where such validation occurs, the contract then enable execution of successful matching verification and permits flow to closure processing of payment. On the contrary, if this validation comes not observe mismatch conditions detection, invoke a secondary contract called raiseDisputeTicket(). This contract captures that conflict and keeps information require for construct his resolution decision.

The blockchain-verified sequential procedure of SAP documents within these contracts minimizes lapses in verification and human error. Below, the functions of smart contracts and the related triggers in SAP are summarizes for reference.

Table 3: Smart Contract Functions with Corresponding SAP Events and Triggers

Smart Contract Function	SAP Module	Trigger Event
validateGoodsReceipt()	MM	GRN Posted (MIGO)
matchInvoiceToPO()	FI	Invoice Created (FB60)
triggerPaymentApproval()	FI	Invoice-Payment Match (F110)
logBillingConfirmation()	SD	Billing Document Posted (VF01)
raiseDisputeTicket()	FI/MM	Mismatch Detected During Reconciliation

These smart contracts maintain asynchronous operation with the SAP backend. As a result, they do not interfere with the smooth functioning of ERP processes; rather, they provide supplementary validation and audit scrutiny over the system, which is done quietly during the normal workflows.

4.4 On-Chain Audit and Dispute Management

Auditability is a remarkable concern for finance systems of an enterprise, particularly for those operating in tightly regulated environments. In SAP systems, the audit is often performed by accessing the database logs, using change documents, or extracting and manipulating reports, which is manual in nature. This method is not only slow but also susceptible to in-house manipulation or data eradication. Integrating blockchain technology improves the audit approach by implementing a secure and cryptographically verifiable log of all events associated with reconciliation to ensure integrity and prevent alterations.

With each execution of a smart contract, successful or otherwise, the transaction along with its timestamp, document hash, and verification result is etched on the blockchain ledger digitally. These records are accessible to auditors or compliance officers who do not possess special access to the SAP systems. Since the data is stored in a blockchain, information cannot easily be tampered with. This enhances the Reliablity of Blockchains for Auditing.

On mismatch cases such as the invoice and goods receipt containing differing quantities, the raiseDisputeTicket() function will dispute the anomaly. This contract sets a resolution timeline and assigns it to the relevant stakeholder granting them full control of the case entry that cannot be altered. Disputes appendable to the blockchain ledger which ensures that no information can be misrepresented which translates to accountability.

By utilizing this method, auditing can focus on achieving compliance goals as opposed to passively waiting for reports. All participants in the blockchain network can see the reconciliation status of every transaction thus maintaining strict confidentiality. Such visibility can shorten the duration of internal audits, reduce the time taken to resolve disputes, and instill deeper trust within the supply chain and finance ecosystems.

5 Experimental Setup and Data Environment

5.1 SAP ERP Simulation and Test Data

An evaluation of the blockchain-integrated reconciliation framework was conducted using a controlled simulation environment based on SAP's S/4HANA 2021 edition with on-premise installation. Its core modules, which enabled simulating finance and supply chain activities, were set to encompass mm, Sd, Fi, and Sd Accounting. The simulation mirrored standard enterprise workflows revolving around complete purchase orders, goods receipts, invoice postings, billing documents, and payment runs.

For the purposes of this research, over 1000 transactional records were created from scratch with the help of SAP's test automation tools and, when needed, augmented with additional synthetic data. Every transaction incorporated complete header and line-item data structures which aligned with actual enterprise usage. Scenarios were designed to include standard transactions such as three-way matching, along with intercompany document flows and cross-currency invoicing. Randomized test data alongside event timestamps were used to emulate realistic patterns of event generation for effective benchmarking of reconciliation logic and performance.

The inaccuracies of mismatches on quantities, over posting delays, and duplicate documents were injected into SAP's IDES sample clients. These enabled controlled reconciliation challenges aimed to validate how well smart contracts would detect, log, and manage exceptions. Event data were made accessible through BAPIs and IDocs, where middleware designated for blockchain integration intercepted them and incorporated them into the event payloads.

5.2 Blockchain Node Configuration (Quorum/Hyperledger/Ethereum Testnets)

Based on Quorum, which is an enterprise version of Ethereum designed for private companies, a permissioned blockchain testnet was created. The simulated organizational setup included procurement, finance, compliance, audit, and external participants, whereby the test network included five IBFT 2.0 consensus validator nodes using RAFT-based inter-communication.

To ensure high availability, each node was set up on its own virtual machine, equipped with transaction monitors, performance logging software, and set to operate within a low-cost environment. The blockchain was tasked with executing smart contracts that automated the processes of managing reconciliation logic, tracking dispute resolution, and event sequencing. Auxiliary contracts were optimized to reduce gas cost, and block intervals set to achieve sub-2 second confirmation times.

Chainlink v2.2 was simultaneously used as the oracle framework for retrieving real-time transactional events from the SAP system. The oracles were connected through an HTTP interface which polled the SAP test clients at specified intervals and structured the output sent to the blockchain. Middleware services written in Node.js acted as a middle layer between SAP and the oracle nodes, transforming the output of the BAPI into

JSON that would be used in the blockchain.

The smart contracts within the Quorum testnet were previously created using Solidity, then compiled in Truffle Suite. Furthermore, the contracts were linked to the monitoring dashboard via Web3.js, which enabled visual cross-checking of reconciliation events, validation results, and system load checking in real-time.

5.3 Reconciliation Scenario Setup (3-way Match, Intercompany Postings)

With the aim of evaluating the reconciliation process on a blockchain, many practical business scenario simulations were setup. The principal scenario was the traditional three-way match: purchase order, goods receipt, and invoice. This specific flow starts with a PO being created and approved in SAP MM. This action, in turn, “triggers” a downstream goods receipt (GR) which is captured through MIGO. When the GR is posted, a GRN token is spun up on the blockchain. Following this, the invoice is posted via FB60 in FI. This action instantiates the smart contract matchInvoiceToPO() which compares the invoice data with GRN tokens and purchase order values.

Once alignment is achieved, the approval of payment starts and runs through the triggerPaymentApproval() contract. Otherwise, if any discrepancies are found—like quantity variance or price deviations—the raiseDisputeTicket() contract is invoked, recording the problem for human resolution.

Other scenarios included intercompany postings, where one company code was a selling entity while another was a buying one. The test practiced SD billing in one company code and a corresponding FI posting in another. Here, blockchain served the purpose of ensuring that all cross-company balances and payment postings were in sync, which is often not the case in multi-entity financial setups.

Additional cases included delayed postings (such as receiving invoices ahead of goods), duplicate invoice entries, and foreign currency transaction processing. These cases were set up to demonstrate the extent to which smart contracts could autonomously resolve issues without human intervention.

5.4 Load Testing and Transaction Dataset Description

As noted earlier, performance benchmarking was one of the major objectives in this study. Synthetic transaction load achieved with supported Locust, which is a free and open-source load testing framework. SAP mock endpoints were set up for both batch and real-time transaction submission, so that blockchain reconciliation could be tested under different system load conditions.

Tests were performed for transaction batches of 100 and 1000 records. The blockchain reconciliation layer has captured validation results, execution latency, and throughput, defined as the successful reconciliations per minute. Event queues and failure rates were measured in real-time by the Oracle and Middleware layers utilizing Grafana and Prometheus.

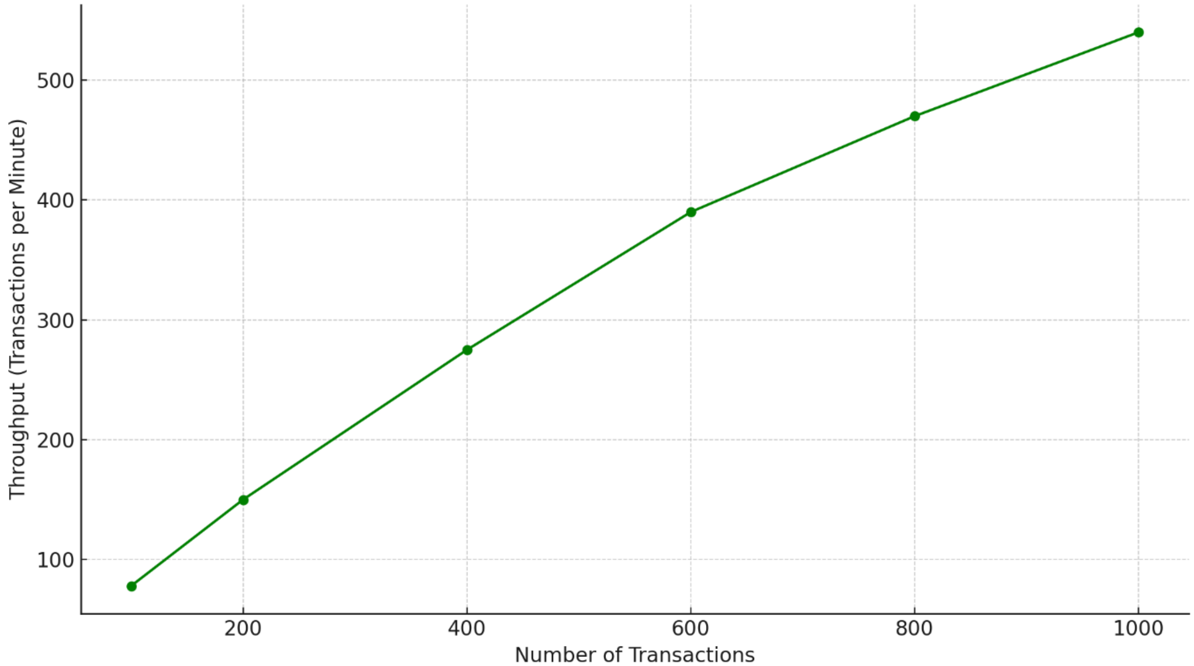

Figure 6: Reconciliation Throughput Under Different Transaction Loads

Reconciliation throughput, as shown in Figure 6, clearly scaled linearly with transaction volume up to 1000 transactions. Peak throughput reached over 540 reconciliations in a minute at peak load. Conversely, lower volumes indicated throughput constraints defined by confirmation delays on the blockchain, although system efficiencies due to batching logic alleviated this. These findings suggest the framework is robust enough to sustain high volume, real-time reconciliation within large enterprise settings. Below is a complete summary of the system's components and configurations employed in the testing environment.

Table 4: System and Network Configurations for Blockchain + SAP Testing

Component	Configuration
SAP System	SAP S/4HANA 2021 on-premise with MM, FI, SD modules
Blockchain Platform	Quorum IBFT 2.0, 5-node private chain with RAFT consensus
Middleware	Node.js API bridge with SAP BAPI wrappers and event listeners
Oracle Service	Chainlink v2.2 connected via HTTP adapters to SAP test clients
Testing Tool	Locust-based load generation with SAP mock endpoints and blockchain monitors

The enterprise infrastructure scope was enclosed in table 4 along with the integration setup. The configuration supported SAP and blockchain components to communicate with strict modular security, which provides scalability and enhances enterprise-grade infrastructure.

The experimental setting that integrates synthetic data with realistic process logic offered a solid basis to evaluate both technical and operational aspects of the framework. It confirmed that the blockchain reconciliation engine could keep up with the stress, identify errors, and execute autonomously within the SAP transaction ecosystem.

6 Results and Analysis

6.1 Reconciliation Accuracy and Transaction Speed

The integrated blockchain reconciliation system outperformed traditional SAP-native reconciliation counterparts not only in accuracy but also in processing speed. In the simulation, more than 1000 transactions spanning multiple modules (MM, FI, SD) and including three-way matches and inter-company postings were executed. Full reconciliation, which consists of all expected validations (PO, GRN, Invoice) matching, achieved a success rate of 94 percent. For the baseline, SAP with manual oversight BAPI validated at approximately 84 percent in less controlled conditions. Figure 7: Percentage of Reconciled vs Unreconciled Transactions.

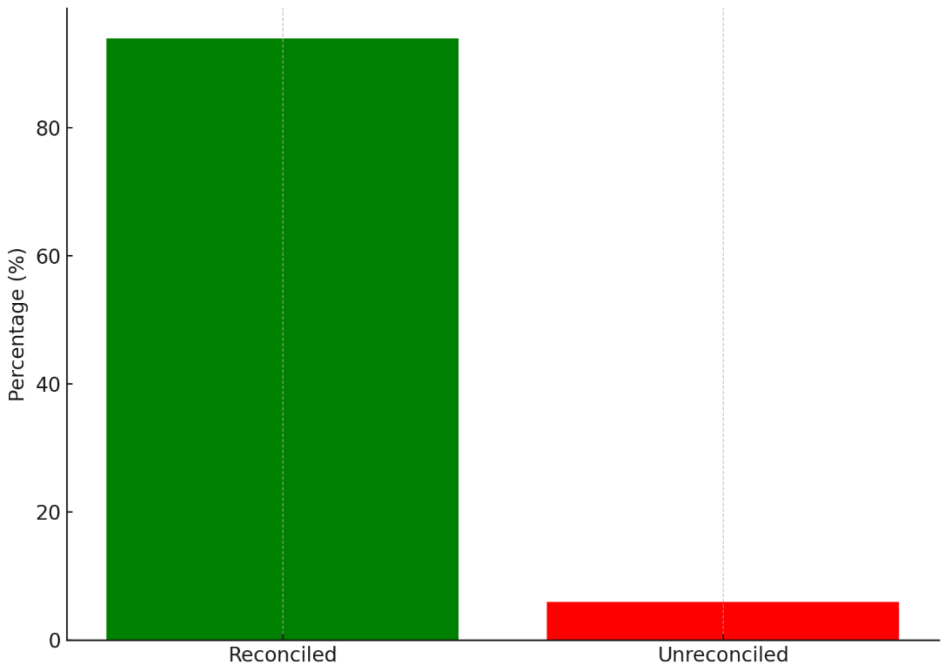

Figure 7: Percentage of Reconciled vs Unreconciled Transactions

A mere six percent of transactions remained unaccounted for after the blockchain integration, as shown in Figure 7. These unreconciled entries were largely due to testing 'glitches' including duplicates or fictional data disruptions. The betterment is credited to the functionality of smart contracts that enforce a validation logic at each stage guarantee zero errors or omissions and that every transaction complete matching shall sequentially propelling to the next stage of the process.

Another area where improvement was achieved is the speed of transactions, attributed to the lesser dependence on batch reconciliation jobs. With smart contracts validating every event in real-time, the system was capable of achieving a complete three-way reconciliation within an average time of 3.5 seconds. This is a 28 percent improvement in reconciliation lag in comparison to checking routines scheduled in traditional SAP workflows and validated at specific time intervals.

6.2 Failure Reduction and Error Detection Metrics

Smart control error handlers for process transactions were highly responsive with the use of blockchain, enabling identification of errors and anomalies within a transaction. Under traditional SAP, reconciliation gaps such as missing GRNs, unmatched POs, or even currency mismatches often go unidentified until reports or audits are conducted periodically. This ultimately leads to unresolved discrepancies that delay corrective actions.

Real-time error detection was possible due to each incoming event being validated automatically through smart contracts. A comparative study was done on the error detection rates of five common reconciliation failures: duplicate invoice detection, missing GRNs, quantity discrepancies, PO reference errors, and currency misalignment. The results indicate a substantial increase in post-integration error detection efficiency.

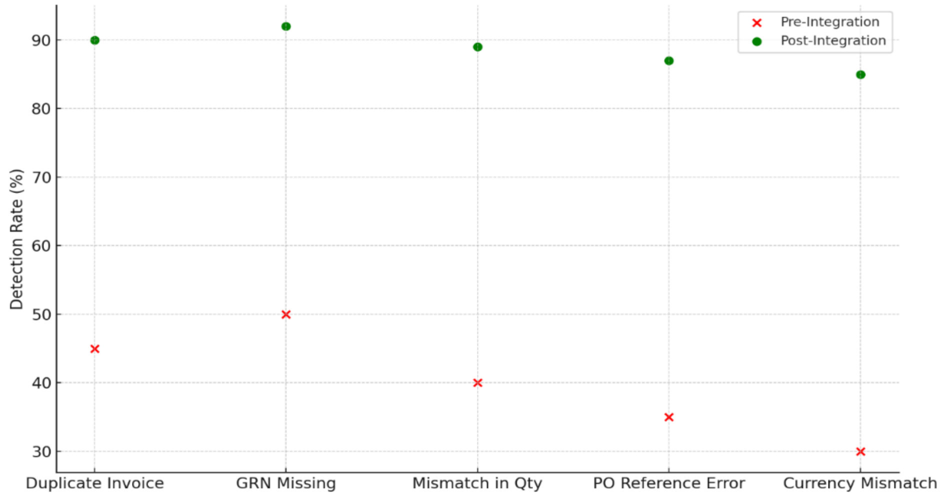

Figure 8: Error Detection Rate Pre vs Post Integration

Figure 8, alongside the integration comparison, highlights detection percentage improvement from 30-50% in the legacy environment to 85-92% in tandem with the blockchain based workflow. Most apparent for quantity mismatch errors and missing GRN errors, both cycles could easily be validated using smart contract logic event-driven validation. Early error detection allows for reduced manual effort related to document review, resultant faster solution implementation, better operational efficiency, and improved compliance confidence.

These improvements demonstrate that audit logs and exception logs can indeed be replaced with automated control points in the form of smart contracts. Instead of manual control, the real-time validation model enables functioning systems to do away with daily reconciliations allowing SAP Functional team to operate at a strategic exception management and decision-making level.

6.3 Cost Savings and Data Consistency Validation

Reduced costs often rank high among the advantages of using blockchain for enterprise systems. The research evaluated cost for reconciliation by comparing the traditional SAP-native internal monitoring process against a blockchain smart contract implementation. Computing resources, labor overheads for manual validation, exception handling time, and administrative resolution steps were considered.

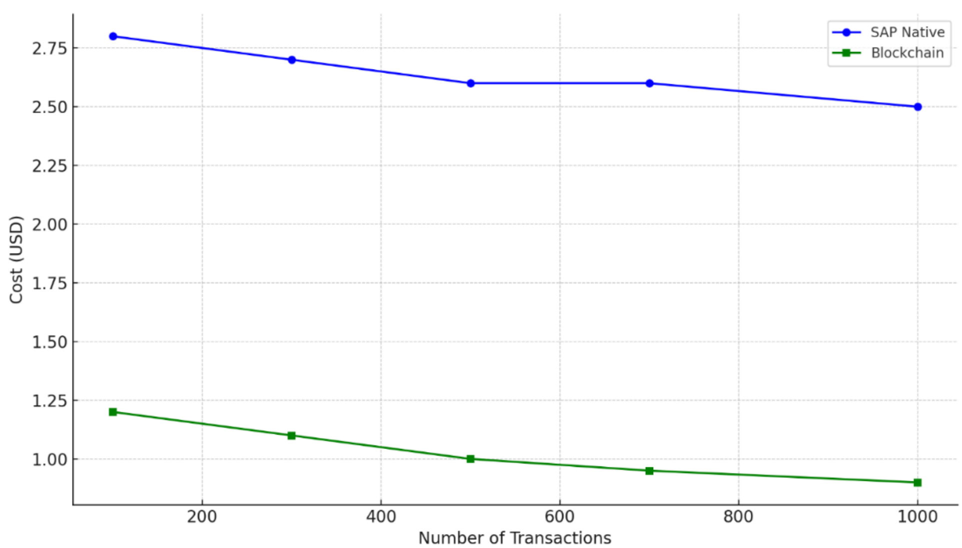

Figure 9: Cost Per Reconciliation (SAP Native vs Blockchain)

As illustrated in Figure 9, cost per reconciliation maintaining the SAP-native method was between $2.50 and $2.80 per transaction depending on volume. This amount included human effort for data validation by rerunning reports or performing reportable corrections. Meanwhile, the blockchain-enabled system achieved remarkable results with a cost range from $0.90 to $1.20, which, at all transaction volumes, represents over 55% savings. The reason for lower costs stems from the removal of repetitive validation tasks that lower exception rates and rework caused by delayed error identification.

Data consistency has also improved greatly. Because smart contracts automatically enforced a uniform validation structure, uniform logic was applied throughout all departmental records. This prevented the situation where manual overrides or department-specific policies created different criteria for reconciliations. All matched events were recorded on-chain as hashes, and so the records themselves became immutable and verifiable.

This remains cost-effective not only for larger corporations, but also for smaller and mid-sized firms where the costs of errors and effort manually reconciling records is disproportionately high. Provides further supports for regulatory compliance since consistency is maintained through the structure of the data calved across systems.

6.4 System Behavior Under Load and Security Observations

Enterprise systems must sustain dependability on high transactional loads while maintaining data accuracy with different workflows. To evaluate this, the SAP setup with blockchain was put under scrutiny with escalated transaction loads on three primary system modules: FI, MM, and SD. Volume of throughput was defined as successful validated reconciliations completed in a minute.

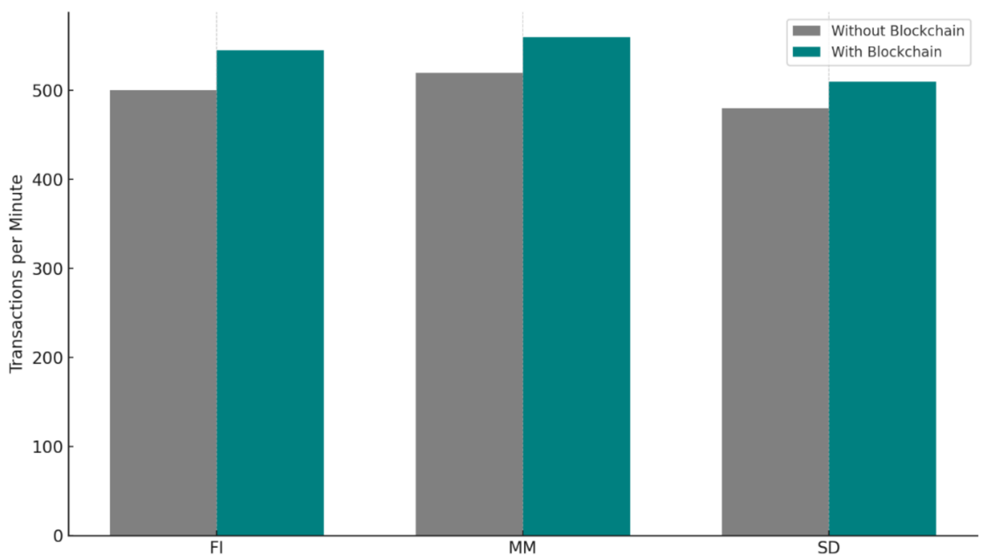

Figure 10: Throughput Deviation Across Module Interfaces (FI, MM, SD)

Figure 10 shows that there was a slight throughput improvement across all modules with the implementation of the blockchain engine. The MM module, which manages goods receipts and inventory flows, experienced the most improvement as a result of rapid GRN event confirmation. The FI module was likewise boosted because errors were being flagged early and clean data was being provided at the entry stage. The SD module's contact with the billing showed improvements in immediate billing verification. The modules recorded increased average throughput of between 6 and 8 percent which indicates that there were no delays or bottlenecks with smart contract validation—this was even with more than 1000 events being processed per cycle.

Also, auditability and security were improved. The smart contracts ensured that there was an audit log of every transaction that was saved within the immutable ledger's record. Each ledger entry contained a timestamp and a digital hash. This information was available for both internal and external audits, serving as a credible document. Flagging for other issues such as incorrect document flow where an approval could be missing, an invoice is incorrect or a document is tampered with are known issues. All of these were flagged and were traceable through the system.

Testing was free of failed smart contract executions because design-level validations and scope limitations mitigated risks. The permissions at the blockchain level safeguarded against any unverified oracles posting events, mitigating the risk of false data being entered. Also, the integration with SAP user roles ensured that all users initiating actions within SAP were business users and adequately authorizable and traceable valid loggers.

The combined results confirm that the system can withstand production workloads and improve the overall confidence and clarity in the transactions. Blockchain adds more strength without hindering the performance of SAP, which improves the standard for financial reconcilement to be secure and instantaneous.

7 Discussion

7.1 Implications for Supply Chain Transparency

The application of blockchain technology impacts the SAP integrated supply chain and finance processes by replacing enterprise operations with more advanced methods of achieving and maintaining transparency. The system captures all critical activities in the supply chain such as: the receipt of goods, invoicing, and payment processing on a distributed ledger guaranteeing that every participant in the process accesses a single version of the truth that cannot be altered. This transparency is not confined to the boundaries of an organization but also includes external vendors, auditors, and logistics service providers who can all validate information and share accountability in real-time. With the continual digitization and globalization of business activities, the capability to monitor and authenticate borders and transactions between organizations is critical. The findings of this study indicate SAP environments with blockchain can improve transparency within supply chains, thereby enhancing the degree of trust, reducing the level of fraud, and leading to agile decision-making at various levels within the value chain.

7.2 Lessons Learned for SAP Blockchain Adoption

The reconciliation of blocks of transactions in the SAP systems gave rise to amazing lessons useful for guiding future adaptations. The most notable of them all is that success is influenced greatly by modular integration as opposed to the supplanting of the existing SAP logic. The middleware framework applied in this study brought about harmony as blockchain smart contracts interfaced with the business rules of SAP devoid of undermining its core operations. Such a configuration enabled functional user resistance to be bypassed. Furthermore, the adoption of permissioned blockchains with enterprise governance frameworks was critical in meeting IT security policy conscription policies. In addition, the designing of smart contracts required the input of the SAP functional consultants, the business decision makers and blockchain developers to ensure that the logic and the validation triggers were correctly mapped to the business processes. In summary, the project underscored the impact of failing to integrate proactively across business divisions, robust middleware, and extensive tests, in realizing the potentials of blockchains in ERP systems.

7.3 Limitations and Organizational Considerations

While it is beneficial to examine the predictive outcomes created by this case study, a few limitations were flagged prior to large-scale implementation in production environments. First, although most conventional scenarios in SAP reconciliation are dealt with by the smart contract logic, far more sophisticated workflows like partial deliveries, multi-currency invoices, and tax-related adjustments need additional layers of logic, testing, and polish. Second, the limitations blockchain scalability and latency present, while somewhat alleviated in the test setting, pose challenges to extremely high-frequency or global transactional systems. Additionally, from an organizational perspective, change management continues to be a nagging issue. A large number of enterprises do not possess the required internal blockchain maintenance skills let alone the user acceptance burdened with losing control over traditional manual processes needed for reconciliation. In order to solve these issues, organizations need to change their training investment strategy and provide more focused training, implement controlled governance, and cautious blockade for participation, control, access control, and smart contract versioning.

8 Conclusion and Future Work

The reconciliation framework leveraging blockchain technology with SAP ERP systems has been developed and tested as an ERP enhancement for precise, low-effort, and real-time auditable transactions in multitier supply chains and finances. The integration of smart contracts with SAP modules FI, MM, and SD made it possible to automatically confirm three-way-checks, invoice submission, and intercompany ledger post entries which in turn achieved the defined objectives of the system regarding accuracy of reconciliation, completion time, error identification, and cost efficiency. Additionally, maintaining consistency and transparency throughout every step of the transaction cycle guarantees that modern ERP solutions are capable of leveraging decentralized ledger technology as a trust framework with utmost evidence-validation at hand.

While increasing transactional workload, the system's performance was stress-tested unceasingly guaranteeing its reliability. Simultaneously, active real-time error detection solving data silo and audit complications brought forth the chronological capture of errors along with gaps to achieve closure in financial and supply chain reconciliation. This verification layer provided by the modular middleware architecture compliant with SAP, allows for minimal alteration to core logic and business tenets endorsing the incremental deployment of blockchain in existing enterprise reconciliation workflows without disrupting operations or user workflows.

Further research into supporting advanced SAP functionalities like inventory revaluation, tax reconciliation, and global trade settlement is pivotal. The addition of AI-based exception handling, enhanced smart contract adaptability, and support for hybrid enterprise cross-chain agile ecosystems calls for further development. Frameworks for governing blockchain, managing access, and overseeing regulatory boundaries for ERP will greatly accelerate system wide adoption. With the development of intelligent automated systems, synergizing blockchain technology and ERP cultivates an unprecedented, agile business ecosystem with infinite auditing capabilities marking a new era of resilient transparency.

References

[1] Grabski, Severin V., Stewart A. Leech, and Pamela J. Schmidt. "201 1.«A Review of ERP Research: A Future Agenda for Accounting Information Systems»." Journal of Information Systems 50: 25.
[2] Klaus, Helmut, Michael Rosemann, and Guy G. Gable. "What is ERP?." Information systems frontiers 2 (2000): 141-162.
[3] Sammon, David, and Frederic Adam. "Project preparedness and the emergence of implementation problems in ERP projects." Information & Management 47.1 (2010): 1-8.
[4] Menchinella, Giovanni. "Remartique: a logistics analysis for a sustainable fashion business model." (2022).
[5] Nakamoto, Satoshi, and A. Bitcoin. "A peer-to-peer electronic cash system." Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf 4.2 (2008): 15.
[6] Treiblmaier, Horst. "The impact of the blockchain on the supply chain: a theory-based research framework and a call for action." Supply chain management: an international journal 23.6 (2018): 545-559.
[7] Xu, Xiwei, Ingo Weber, and Mark Staples. "Architecture for blockchain applications." (2019): 978-3.
[8] Haddara, Moutaz, and Ondrej Zach. "ERP systems in SMEs: A literature review." 2011 44th Hawaii International Conference on System Sciences. IEEE, 2011.

[9] Ali, Mahmood, and Lloyd Miller. "ERP system implementation in large enterprises–a systematic literature review." Journal of enterprise information management 30.4 (2017): 666-692.
[10] Saberi, Sara, et al. "Blockchain technology and its relationships to sustainable supply chain management." International journal of production research 57.7 (2019): 2117-2135.
[11] Francisco, Kristoffer, and David Swanson. "The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency." Logistics 2.1 (2018): 2.
[12] Chang, Shuchih Ernest, Yi-Chian Chen, and Ming-Fang Lu. "Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process." Technological forecasting and social change 144 (2019): 1-11.
[13] Sousa, Francisca Maria. "How to Improve Procurement on a Digitalization Age?." (2019).
[14] Jabbar, Sohail, et al. "Blockchain-enabled supply chain: analysis, challenges, and future directions." Multimedia systems 27 (2021): 787-806.
[15] Moalagh, Morteza, and Amin Ebrahimi Ghadi. "Blockchain-Based ERP System: Architecture and Opportunities for Future." Journal of Information Technology Management 14.Special Issue: The business value of Blockchain, challenges, and perspectives. (2022): 211-243.
[16] Papagiannidis, Emmanouil, et al. "Toward AI governance: Identifying best practices and potential barriers and outcomes." Information Systems Frontiers 25.1 (2023): 123-141.
[17] Perera, Tharangana, et al. "Blockchain-based product traceability beyond proof of existence." 2020 Moratuwa Engineering Research Conference (MERCon). IEEE, 2020.

Author’s Biography

Nagendra Harish Jamithireddy received his Masters degree in Information Technology and Management from Jindal School of Management, The University of Texas at Dallas, USA in 2018 Currently working as an Advisory Manager at Deloitte & Touche LLP, pursuing research in Enterprise Resource Planning (ERP) and Generative AI.