Fortifying Cloud-Based ERP Solutions: A Secure and Efficient Integration Approach

Udita Malhotra[1,*], Ritu[1]

[1]Department of Computer Science and Engineering, Guru Jambheshwar University of Science & Technology, Hisar-Haryana, India

Emails: drmalhotraudita@gmail.com; ritunagpal1973@gmail.com

Abstract

Cloud-based Enterprise Resource Planning (ERP) systems have become essential to organizational operations in today's digital environment, acting as the cornerstone for managing sensitive corporate data. ERP system integration with third-party apps, however, poses serious security risks because businesses cannot afford data breaches or illegal access that could jeopardize financial records, operational integrity, and reputation. Because ERP systems are appealing targets for cybercriminals looking to obtain sensitive company data, ensuring secure data exchange is an urgent concern. ERP integration security is still a problem, despite the numerous security frameworks and measures that have been put forth. Current methods frequently fall short of effectively addressing new threats. To guarantee the safe and smooth integration of cloud-based ERP solutions with external systems, this study presents an extensible security framework. The framework reduces the risk of data interception and unauthorized access by utilizing functional and technical security measures to produce a strong, adaptable security model. To prevent data leaks and unauthorized changes, the implementation is divided into two phases: (1) securing outbound data flow from the ERP portal to third-party systems, and (2) securing inbound data flow from third-party systems into the ERP portal, which protects against malicious intrusions and breaches of data integrity.

Keywords: ERP; Third-party system; Interception; Extensible; Security; Integration; Portal

1. Introduction

With the advancing technology, ERP (Enterprise Resource Planning) systems have been a transformational force in the modern workplace. ERP solutions are now part of daily operations, improving workflows and productivity for all enterprises, immediately making the systems vital to businesses across all sectors, including education, technology, defense, aerospace, and healthcare [6]. Microsoft Dynamics AX is perhaps the most popular of all ERP solutions, especially with medium and large enterprises. AX allows organizations to track steps and improve productivity, as well as to sustain competitive advantages globally. AX automates the core functions of business, including supply chain, financials and business intelligence, and allows a seamless execution of business and strategic plans [1].

AX can be outlined as a multi-language, customizable, and multi-currency enterprise resource planning solution. AX is very beneficial like service industries, wholesale, manufacturing, and e-business in various fields. AX has been a unique and strong solution with extensible technical and functional features [1].

Web API can import and export data from AX to the portal. There is an integration framework required to maintain consistent integration between AX and the portal [2]. The suggested framework creates a robust defence mechanism that allows enterprises to integrate their ERP systems with confidence without sacrificing data security by strengthening security at both ends of the integration process. The study emphasizes the importance of real-time security models and shows how ERP solutions' technological developments can make it easier to implement stronger security measures. By providing a scalable, flexible, and all-encompassing method of securing ERP integration, the study's conclusions help to advance ERP security by guaranteeing the secure and effective transfer of data between cloud-based ERP platforms and external systems [3].

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

There is a need to propose a secure and compatible model for integration, which would be applicable for all third-party systems. The integration framework would be a generic solution and applicable for all modules.

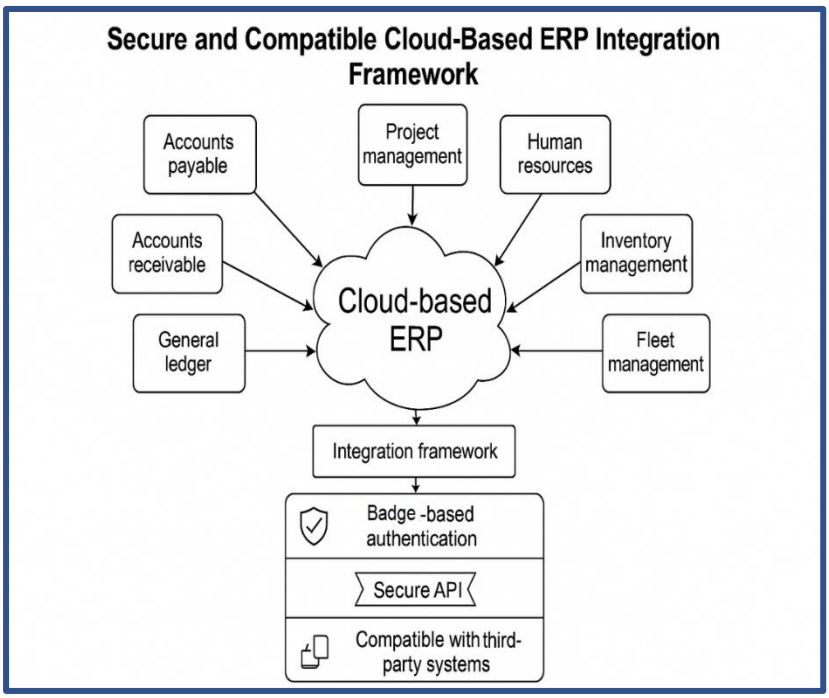

Figure 1. Modules of cloud-based ERP system

Figure 1 shows that there is a generic framework required for the integration of cloud-based ERP systems with third-party system. Here, badge framework will be used for add on security for the data in transit. In addition, this methodology needs to be applicable for all modules, i.e., accounts payable, project management, fleet management, human resources, accounts receivables, general ledger, payroll, warehouse, inventory management, sales and marketing, purchase and procurement, contract management, warranty and service, etc.

2. Problem Formulation

Although there have been many techniques and research methodologies proposed in the sector of secure integration of cloud-based ERP systems there are various limitations of these researches. Therefore, a better, customizable, flexible mechanism can be proposed to overcome the limitations of traditional approaches. The technical and functional features need to be integrated to formulate a customizable framework for integration security to avoid interceptions. This research work could provide a better and more efficient solution to achieving the aim of secure and compatible integration of cloud-based ERP systems.

3. Related Work

There are several types of research related to the integration security implementations for cloud-based ERP solutions. The below segment provides a literature review related to the integration of a cloud-based ERP system with a third-party system using blockchain technology and an IoT-based smart retailing system. The methods used in these are blockchain technology, cloud, radio frequency identification technology, and several data-mining methods.

3.1 ERP Integration with AIS

The combination of Enterprise Resource Planning (ERP) systems with Accounting Information Systems (AIS) has the potential to be significantly improved with the use of blockchain technology [7]. Although AIS is one of several components of ERP systems, it is often regarded as the backbone of ERP. With the development of distributed ledger technologies like blockchain, there are increased opportunities for improvements in security and privacy. Several articles have cited the benefits of integrating blockchain to secure integration of systems. The multi-layered architecture of blockchain allows for improving integration in multiple areas of operation such as compliance and auditing. A case study method was used to review the functionality of the e-procurement module, which demonstrated that systems using Distributed Ledger Technology (DLT), Financial Technology (FinTech), or Decentralized Finance (DeFi) could achieve integration of AIS and ERP systems. Integrations improve efficiency, productivity, and security in this model in which transactions are grouped into blocks and managed in a decentralized manner.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

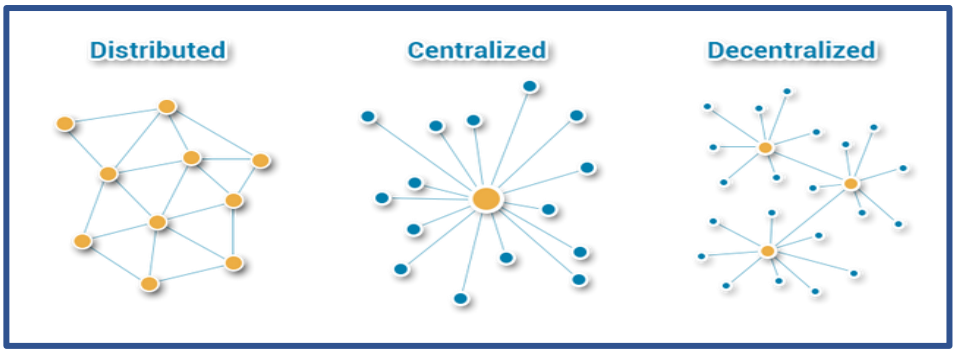

Figure 2. Distributed vs Centralized vs Decentralized systems

Figure 2 compares three types of network architectures: Distributed, where each node communicates with multiple others; Centralized, where all nodes connect to a single central node; and Decentralized, where multiple central nodes manage smaller clusters, offering a balance of control and resilience [7].

The Accounting Information Systems (AIS) serve as an essential platform for supporting the e-procurement system [7]. The main components of AIS includes the trial balance, which accounts for the sum of net debit and credit amounts within every ledger account for the financial year; "bookkeeping" and the recording of a particular ledger entry; financial statements that used for reconciling the income statements with the balance sheets; and analysis of the financial statements, which helps in assessing the overall financial position and performance of an entity.

Smart contracts serve an important role in e-procurement that works with blockchain technology. They provide security and compliance with the terms of the contract by providing an automated, self-executing structure for contracts created between entities or individuals. These contracts are created and stored on the blockchain, and they are automatically triggered once the previously defined conditions are met. All parties must sign digitally to validate the agreement and to create enforceability.

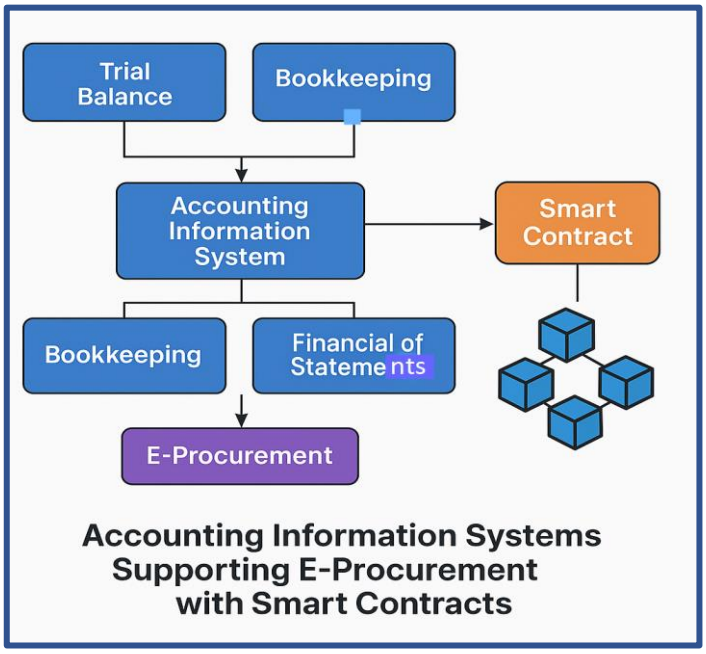

Figure 3. Accounting Information Systems

This diagram illustrates how Accounting Information Systems (AIS) integrate with e-procurement systems using smart contracts. Core AIS components like trial balance and bookkeeping feed into the system, enabling automated, secure, and transparent procurement processes governed by blockchain-based smart contracts [7].

3.2 ERP Integration with Procurement System

Blockchain technology is a strong foundation for connecting ERP systems with procurement components. This connection gets even smarter when we factor in IoT technology and it is even more secure when we add RFID technology. In the supply chain space, blockchain is able to create advanced features like self-billing, which streamline checkout processes and reduce the time it takes for customers to make payments. Even with technology

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

automating these processes, security is met with RFID technology. The RFID trackers can recognize unpaid items when people leave a store, eliminating theft and ensuring secure and reliable transactions [8].

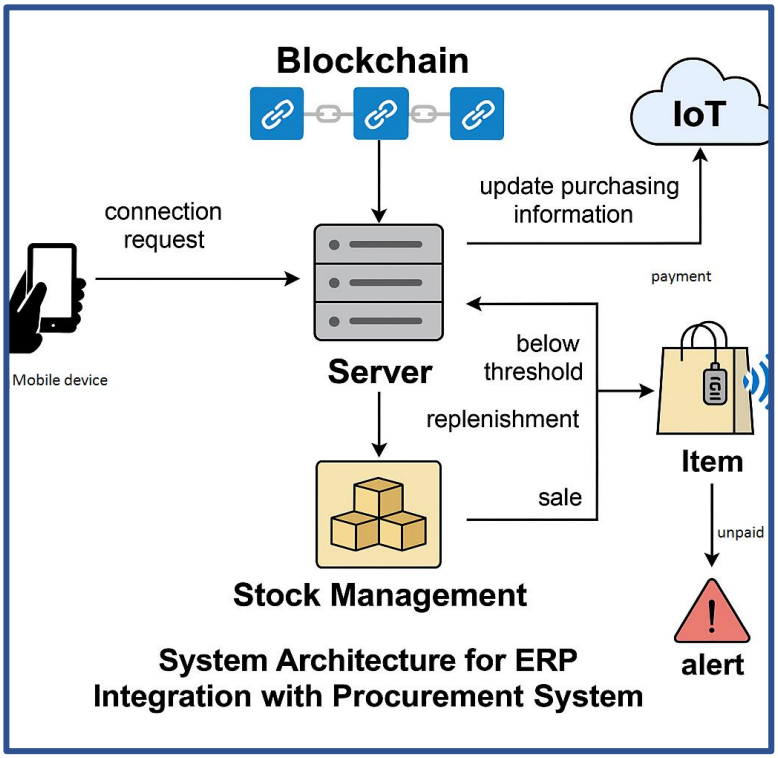

Figure 4. System Architecture for ERP Integration with Procurement System

Figure 4 represents a System Architecture for ERP Integration with Procurement System, displaying how Blockchain, IoT, and RFID enhance automation and security. A mobile device initiates a connection with the server, which interacts with stock management and updates purchasing info. Blockchain ensures secure transactions, IoT tracks inventory levels, and unpaid items trigger alerts through RFID, preventing unauthorized checkouts.

The proposed system architecture begins with a mobile device sending a connection request to the server. The server, after the connection is made, will update the customer's purchasing information and communicate with the stock management module to adjust the current stock levels, depending on what items are below their threshold. If there is an item that went below its threshold, the system can also automatically generate a new order of stock to replenish that item [1]. The system also allows for transparency because it lets customers extract and view item information specific to that customer. All purchasing activities will also be saved in the server for future reference. Payment will be made, based on whatever method the user prefers after completing the transaction the RFID tag on the item purchased would be deactivated which would allow the customer to leave the store without any interruptions. If that item remains unpaid then the tag associated will not deactivate and an alert will be created when the customer leaves the store with the unpaid item. This improves store security and helps to prevent goods from being removed from the store without paying [8].

Table 1: Algorithm for ERP integration with the Procurement system

A mobile device submits a connection request to the server.
The server records and controls purchasing data.
The stock management module refreshes inventory counts and begins restocking if required.
When payment is successfully performed, the RFID tag is deactivated and the customer is permi exit.
When payment is not completed, the RFID tag remains active and an alarm will be initiated if a is attempted.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

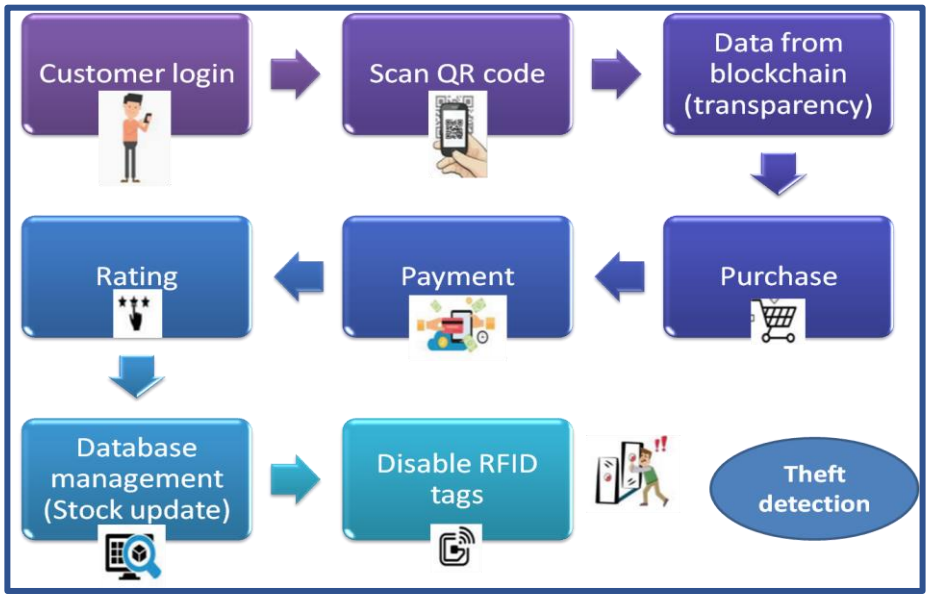

Figure 5. System design for ERP integration with the Procurement system

This diagram illustrates a Smart Retail Workflow Using Blockchain and RFID for Theft Detection. It begins with customer login and QR code scanning, and then retrieves transparent data via blockchain to enable purchases. After payment, RFID tags are disabled, updating the stock database. If an item remains unpaid, theft is detected automatically, enhancing security and operational efficiency [8].

A. Methods based on blockchain technology for procurement

a.) Bumble Bee Foods: this SAP portal is a cloud-based ERP solution. It makes use of blockchain technology so that the customers can fetch the product origin and history details by scanning a QR code that is present on the item packet via their smartphones.

b.) Starbucks: They are functional with the Microsoft team to implement a system based on blockchain technology that can be used to maintain tracking information of supply chain management, which the customers can use to get transparent information, related to the purchased beans and coffee [11].

c.) Food security observation in the Procurement module based on HACCP: This involves a supply chain system providing a transparent, secure, and reliable platform for users. The system has been developed using blockchain, IoT, and HACCP, which helped to make an innovative decentralized system to ensure openness [8].

d) Walmart: This involves a supply chain management for food items that is based on blockchain technology and Linux Foundation's Hyperledger Fabric. This involves tracking the origin of food items from various vendors and their end-to-end supply.

B.) Smart procurement systems using IoT

a.) Merchandising Security: This has been used by several marketers to secure their worthy items like mobiles, cameras, tablets, and watches. The same has been achieved by a security stand that lies above their showcased products.

b) Management Tool based on Smart Inventory: In this technique, retailers use smart barcode scanners to analyze and monitor the inventory stock. This has helped them to avoid stealing things. They keep an eye on the stock counts and observe keenly in case any discrepancy is found [1].

c) Electronic Article Surveillance (EAS): Many retailers use this technology in which security tags are attached to the items and an alarm is triggered if any customer tries to steal the product and walks out of the store with the tag still on. This is achieved with the help of electronic sensors present at the outlets.

d) Cameras and Video Analytics: Under this technique, the software is used to detect any malicious activity or suspicious movement of the customers. In case any such activity is found, an alert can be generated instantly [21].

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

Table 2: Literature Review

S.No	Authors	Research paper	Proposed work	Research gap
1.	Ahn, Byungchan, and Hyunchul Ahn, (2020)	"Factors Affecting Intention to Adopt CloudBased ERP from a Comprehensive Approach" [10]	Identified that data security is a significant concern in cloud-based ERP systems.	Lacks a suitable implementation technique to address the security challenges.
2.	Mahmood F., Khan, A.Z. and Bokhari, R.H., (2020)	"ERP issues and challenges: a research synthesis" [9]	Highlighted common ERP issues including security, data migration, and system integration.	No implementation strategy was proposed to resolve these integration issues.
3.	Faccia, Alessio, and Pythagoras Petratos. (2021)	"Blockchain, Enterprise Resource Planning (ERP) and Accounting Information Systems (AIS): Research on e Procurement and System Integration" [7]	Proposed blockchain as a mechanism to support ERP-AIS integration.	No automation method (e.g., e-invoicing) or security framework for APIs/web services was addressed.
4.	Mandal, S., & Khan, D. A. (2020)	"A Study of Security Threats in Cloud: Passive Impact of COVID-19 Pandemic" [17]	Discussed increased cybersecurity threats due to higher cloud usage during the pandemic.	Did not suggest an actionable or technical solution for enhancing cloud ERP security.
5.	J. Shree, N. R. Kanimozhi, G. A. Dhanush (2020)	"To Design Smart and Secure Purchasing System integrated with ERP using Block chain Technology" [8]	Demonstrated ERP integration with procurement using IoT, RFID, and blockchain.	No general secure framework proposed for integration with diverse third-party systems.
6.	Baraa K. Muslmani, Saif Kazakzeh, Eyad Ayoubi, and Shadi Aljawarneh (2018)	"Reducing integration complexity of cloud-based ERP systems" [11]	Suggested ways to minimize complexity in traditional to cloud ERP integration.	Integration methodology applicable only for integration with traditional ERP systems. No generic solution provided.

4. Tools and Methodology used in research work

Research and academia are two of the many learning domains that make use of the AX ERP portal. Because of its adaptable source code, it provides an ERP solution that is flexible enough to be customized as needed. X++, an object-oriented programming language that is similar to C#, is used to create the platform. X++, which was first created as a superset of Java, has strong data access features. It is anticipated that the integration of functional and technical customizations using X++ will improve efficiency in the proposed work by lowering time consumption while guaranteeing high-quality results.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

5. Proposed System

There is an integration framework required to maintain consistent integration between AX and the portal. Web API is used to import and export data from AX to the dealer's portal. Sync status is used in AX to indicate the synchronization status of the data. There are three possible values:

- AX only: This indicates that data is present only in AX and has not been shared with the portal.
- AX and portal: This indicates that the data has been shared with the portal.
- AX Updated: This indicates that data was shared with the portal but afterward, some modifications have been done to the data, which are not shared with the portal.

Therefore, when the API is called to share the data from AX to the portal, only the data, which comes under 'AX Only' and 'AX Updated', will be shared.

Hosting of URLs is done in the portal (e.g., dealers' portal). Initiation of requests either GET or PUT is always from AX. We can prevent the declining operative speed of the computers or servers with the help of batch jobs so that server performance is not impacted during busy working hours. The history log for batch jobs will also be maintained which will provide various informative fields like start and end date time. API integration setup will also be done in which badge details will also be configured.

Outbound Algorithm

1. Fetch the data from main tables that need to be shared with third-party systems.
2. Validate and extract the selected records as per the requirements.
3. Create an XML document for the extracted data where the portal sync status is 'AX only' or 'AX updated'.
4. Save the XML document created at a specified destination path.
5. Request the URL provided with the appropriate content type i.e., POST.
6. Update the portal sync status of the selected records to 'AX and portal'.

Inbound Algorithm

1. Request the URL provided with the appropriate content type i.e., GET.
2. Read the response for the request triggered.
3. The data is extracted from the response file which can be in the desired format (for example XML).
4. Iterate each record present in the data file.
5. Assign the staging field values from the data file before inserting the data values directly into the main tables database.
6. Validate the data fields if the data present in the staging is accurate and can be inserted into the main tables without having any adverse effects.
7. Process the successfully validated data further.

Applying security features to the third-party integration

The concept of badges is used for authentication between AX and the portal. A badge may be defined as a string of 255 characters that is used to allow and authenticate the sharing of data between AX and the portal.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

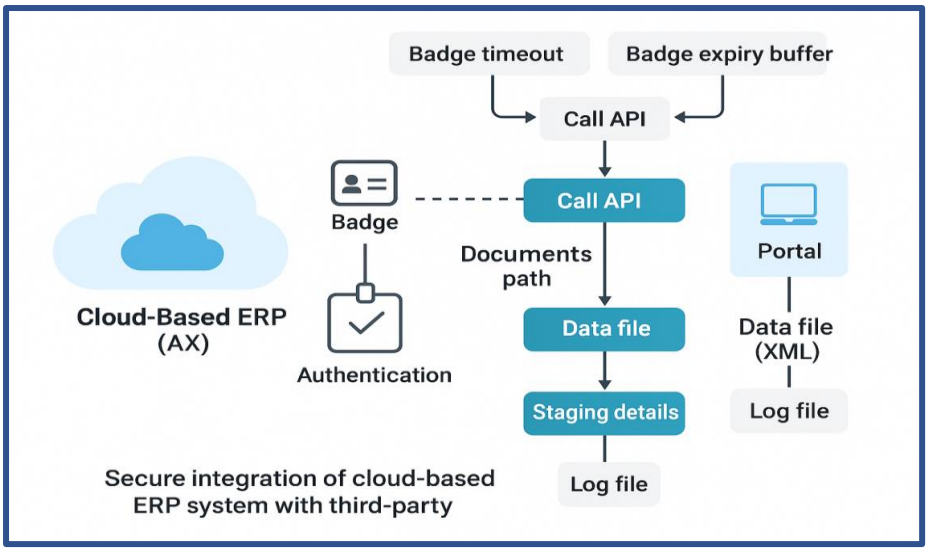

Figure 6. Secure integration of cloud-based ERP system with third party system

Figure 6 illustrates the secure integration of a cloud-based ERP system with a third-party portal. It displays how authentication via a badge enables secure API calls that manage data file transfers. These files undergo staging and logging processes before being transmitted to the portal in XML format, ensuring traceability and compliance through log file management.

Badge timeout: Time between the sending of the request and the arrival of the response.

Badge expiry buffer: The time after which the badge will not be valid.

Base URL: URL of the API which is going to be hit.

Document path: Path of the location where XML files generated will be saved.

API nature: This can be inbound or outbound. In case the data is shared from AX to the portal, outbound API will be used. On the other hand, in case the data is shared from the portal to AX, an inbound API will be used.

Data file: This is the XML file generated.

Log file: This file contains the error id and description, if any.

Records: Number of records that will be sent at a time. (Used only in case of outbound APIs)

Call API: Request goes to the portal. Although, it is preferred to run the batch job instead.

Staging details: It contains the inbound data. When data comes from the portal, it first gets stored in a separate table called the staging table. Then a separate job may be used to transfer the data to the respective location.

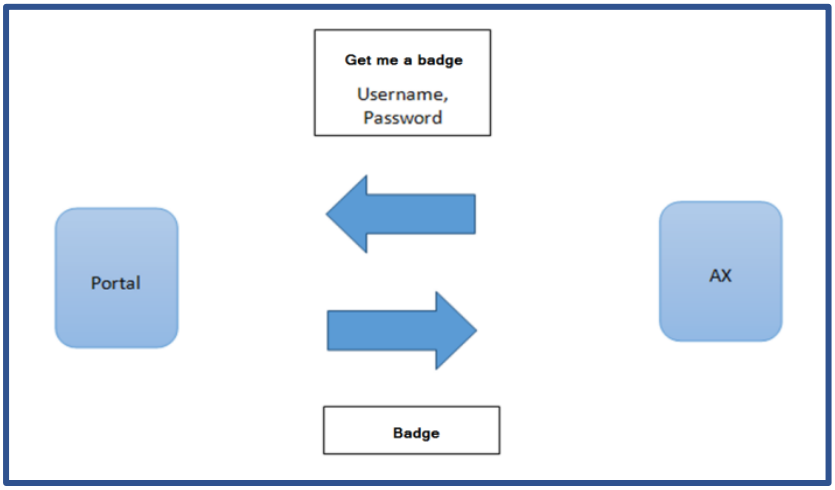

Figure 7. Applying security features to the third-party integration

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

Figure 7 represents a basic authentication flow between a portal and the AX ERP system. The user provides a username and password to request a badge (token). Once authenticated, the badge is exchanged between the portal and AX system, enabling secure communication for data exchange.

Algorithm:

1. A request is initiated from the ERP side with the appropriate authentication card information to the portal side.
2. The portal then checks and authenticates the credentials from the ERP request.
3. The portal then generates and returns the access badge.
4. Assign the time for which the badge is valid.
5. The ERP then includes the access badge in the authorization header of the sent request.
6. The data can be shared between ERP and the portal until the badge is expired.

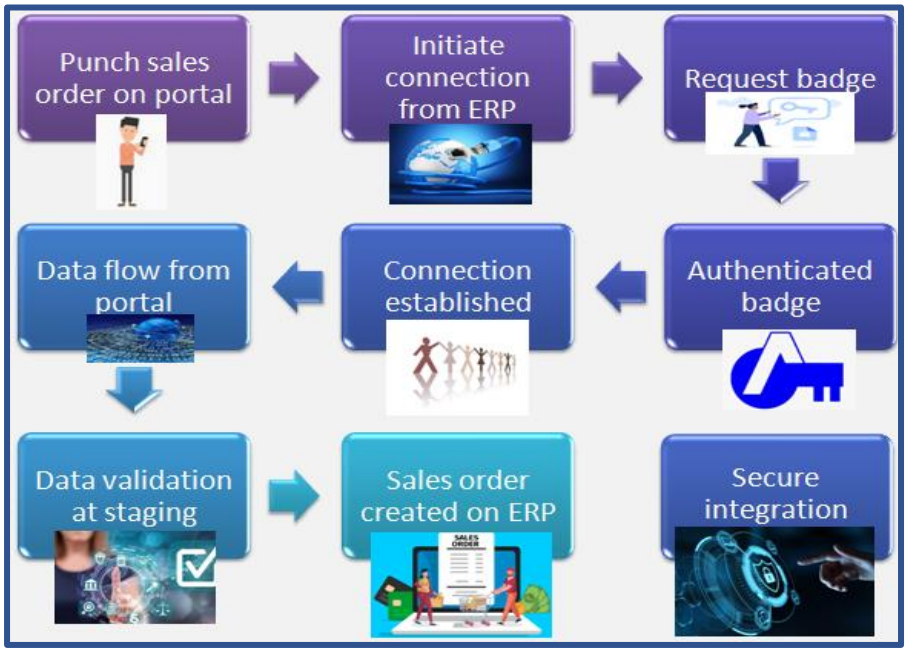

Figure 8. Sales order secure integration of cloud-based ERP system with portal

Figure 8 represents a secure integration process for sales order transfer from a portal to an ERP system. It includes badge-based authentication, connection establishment, data validation, and final sales order creation in ERP.

6. Analysis parameters and equations

To ensure a complete assessment of the integration process, a wide variety of parameters is examined for inclusion. These parameters confirm that the process meets not only functional specifications, but also other functional properties, such as performance, security, and compatibility, are equally factored in.

1. Integration Time: The time duration spent in carrying out the integration is an important parameter of inquiry. Integration time includes time taken for initialization, data migration into the new system, and ongoing synchronization [13].
2. Data Consistency: An important parameter specifies data consistency across systems. This parameter pertains to the accuracy and reliability of business data, transferred from the new ERP system, into third-party systems [19]. The parameters are:

   Error rate: Percentage of data transactions that contain errors

   Error rate = (Erroneous transactions/Total transactions) x 100
3. Data consistency rate: The percentage of data that remains consistent across systems [7].

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

4. System Performance: The system performance reflects the integration process's impact on the performance of the system in terms of response times, transaction speeds, and overall system throughput [14-16] .
5. Scalability: The solution viability is assessed for capacity for increased amounts of data and numbers of users. Scalability is needed for an increasing or growing business and system operations [17-19].
6. Security: Most important is how safe the data will be while in transit and after storage. This involves the use of the encryption mechanism, access control, and data protection compliance. For experimental purposes, a parameter that will be used is the vulnerability score [20]. The vulnerability score can be assessed based on three primary factors: number of security breaches, severity of vulnerabilities discovered, and frequency of security updates [21-23].

- Number of Security Breaches (N): A count of the number of security breaches that occurred in the month.
- The severity of Vulnerabilities (S): A score based on the average severity of vulnerabilities found, typically rated from 1 to 10.
- Frequency of Security Updates (F): A score based on the frequency of security updates, rated from 1 to 10.

Vulnerability Score = $(N/10) \times (S/10) \times (11-F)$

7. Cost Efficiency: The cost-effectiveness of the integration solution is analyzed by taking into account implementation costs, maintenance costs, and potential cost savings that occur through enhanced efficiencies [24-27].
8. User Satisfaction: User views and satisfaction levels are measured to understand the real-world usability and acceptance of the integration. Parameters that are used include:

- Ease of Use (EoU)
- User Satisfaction (US)
- System Responsiveness (SR)

User Experience Score = $(EoU/10) \times (US/10) \times (SR/10)$

7. Comparative Analysis of Integrations methodologies

To validate the effectiveness of the proposed methodology, we compare it against blockchain-based and IoT-based integration approaches. The comparison is based on the key parameters identified in the previous section.

Security Assessment

Security is a primary concern in cloud-based ERP integration. Table 3 presents the results of our security assessment experiment.

Table 3: Security Comparison of Integration Approaches

Methodology	Security breaches (N)	Severity of vulnerabilities (S)	Frequency of Security updates (F)	Vulnerability Score
Blockchain-based	2	6 (high)	7 (moderate)	0.48
IoT-based	5	5 (medium)	6 (low)	1.25
Proposed methodology	1	3 (low)	9 (very frequent)	0.06

The proposed methodology significantly outperforms both blockchain and IoT-based integration approaches in terms of security, with the lowest vulnerability score (0.06).

Data Integrity and Consistency

Ensuring data consistency across integrated systems is crucial. Table 4 compares the data integrity results.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

Table 4: Data Consistency Comparison

Methodology	Total transactions	Erroneous transactions	Error Rate	Data consistency
Blockchain-based	10,000	23	0.23	98
IoT-based	10,000	45	0.45	96.5
Proposed methodology	10,000	7	0.07	98.9

The proposed model demonstrates superior data consistency (98.9%) and the lowest error rate (0.07%), making it a more reliable choice for ERP integration.

User Experience Evaluation

User experience is assessed based on three factors: ease of use, user satisfaction, and system responsiveness.

Table 5: User Experience Comparison

Methodology	Ease of Use (EoU)	User Satisfaction (US)	System Responsiveness (SR)	User Experience Score
Blockchain-based	6	7	5	0.21
IoT-based	5	6	6	0.18
Proposed methodology	8	8	7	0.448

The proposed model achieves the highest user experience score (0.448), suggesting better usability and overall user satisfaction.

Scalability Evaluation

Scalability is a critical factor for cloud-based ERP integrations. The proposed methodology was tested with 20 different third-party systems, compared to limited integrations in existing studies. Faccia [7] model integrated with AIS only. J. Shree [8] framework was able to provide integration solution with purchasing systems only. Ahn [10] model can be used for intergation with traditional ERP systems. On the other hand, proposed model is able to successfully achieve integration with 20 different third-party systems.

Integration Complexity

Integration complexity is assessed by measuring setup time. A statistical analysis was performed to compare the setup times of the proposed model with the J. Shree model [8]. The results are presented in Table 6.

Table 6: Integration Setup Time Comparison

Methodology	Average Setup Time (Minutes)
J.Shree Model	57.55
Proposed Model	49.45

A statistical F-test was conducted with the following results:

Calculated F-value: 18.32

Critical F-value ($\alpha = 0.05$): 4.96

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

Since the calculated F-value is greater than the critical F-value, the null hypothesis is rejected, confirming that the proposed methodology significantly reduces integration setup time.

Performance metrics

Integration points refer to the number of connections or interfaces that a system has to communicate or exchange data with other systems, applications, or services. In the context of integrating a cloud-based ERP system with third-party systems, integration points represent the various touchpoints or endpoints where data is exchanged between the ERP system and external systems [4].

Key metrics analyzed:

- Integration points
- Data exchanges
- Transactions
- Operations between systems
- Response time

The more integration points there are, the more data exchanges, transactions, or operations occur between the systems as illustrated in figure 9.

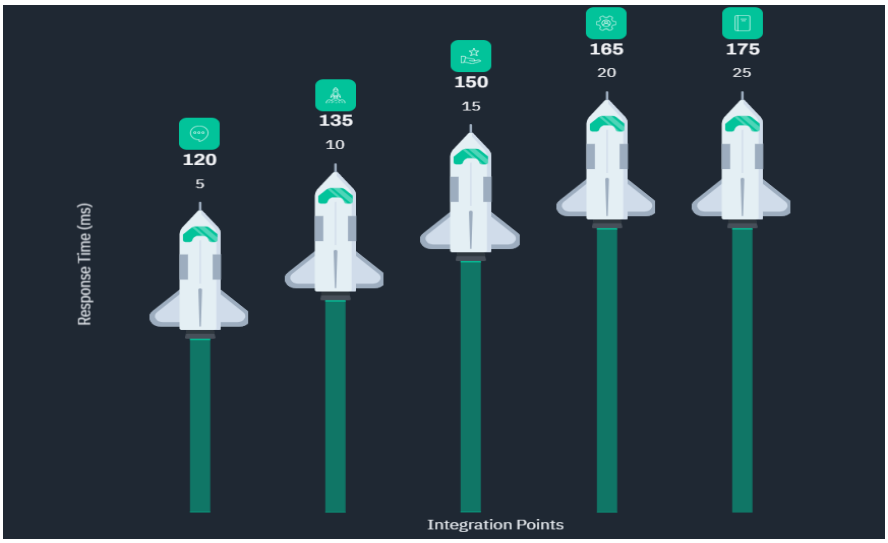

Figure 9. Performance metrics

Regression Analysis Results

A regression analysis was conducted to understand the relationship between the number of integration points and the response time in the proposed model. The results show a positive correlation, meaning that as the number of integration points increases, the response time also increases.

Statistical Analysis:

Mean of Integration Points ($\bar{X}$) and Response Time ($\bar{Y}$): 15 and 147.2, respectively.

Regression Coefficient (b): 2.56, indicating an increase in response time per additional integration point.

Pearson's Correlation Coefficient (r): 0.9866, signifying a strong linear relationship.

Coefficient of Determination (r^2): integration points explain 0.9734, meaning 97.34% of response time variation.

Hypothesis testing (p-value for Regression): $p < 0.001$, confirming statistical significance.

These results indicate a strong positive correlation between the number of integration points and response time. The hypothesis that response time increases with more integration points is statistically supported.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

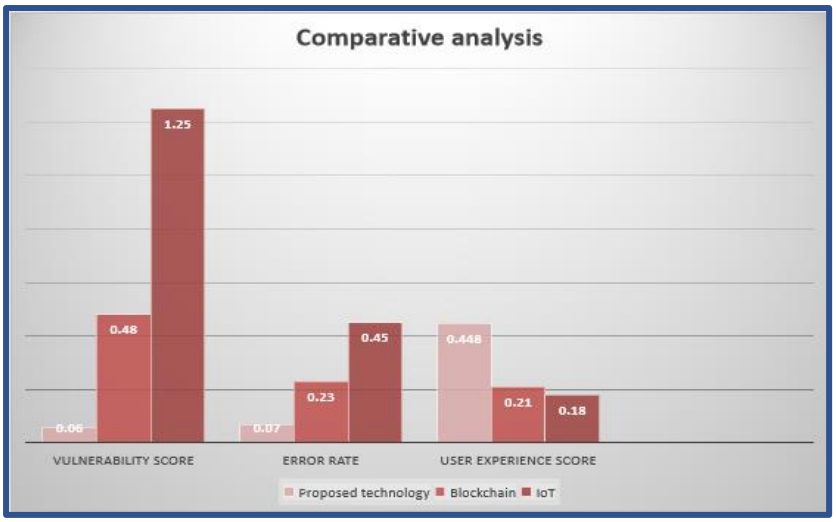

Figure 10. Comparative analysis

As illustrated in figure 10, the analysis highlights the advantages of the proposed model, particularly in security, performance, and scalability, making it an effective solution for cloud-based ERP integration.

8. Results of Implementation

First, an item needs to be created on the cloud-based ERP portal, which needs to be released. As this unit is newly created and not yet shared with the portal, its sync status would be 'AX Only'. This makes it eligible for outbound data sharing.

After that, the unit master API will be run through which a cloud-based ERP solution will try to establish a connection with the dealer's portal for outbound data sharing. If the connection is authorized and an accurate badge is generated, then the data will be shared successfully and the dealer's portal will reflect the item from a cloud-based ERP solution.

A sales order can be created on the dealer's portal to check the inbound data flow. Unit order API will be triggered to initiate a connection of the dealer's portal with the cloud-based ERP solution. If the authorization is successful, the generated badge will be used for the inbound data flow. The data will be inserted into the staging, validated, and then pushed to the main tables.

In case, some unauthorized user tries to establish a connection, an alert message will be generated and the data will not be shared. If incorrect badge information is used for data sharing, then the connection between the cloud-based ERP solution and the dealer's portal will not be established.

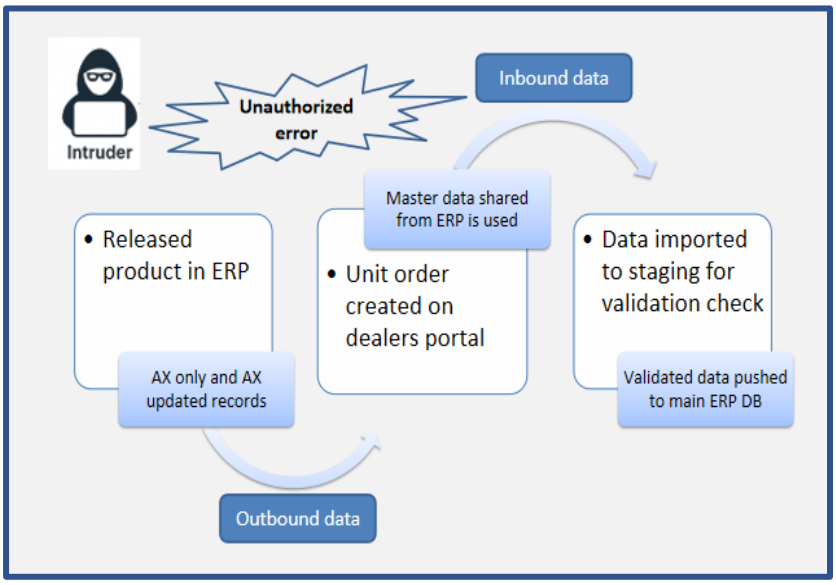

Figure 11. Secure integration with the help of badge framework

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

Figure 11 shows a secure integration framework. When an item is released in ERP, and has the relevant status required for synchronization, will be shared with the dealer's portal via outbound unit master API. User can create transactional data on dealer's portal like a unit order with the help of master data received. This transactional data is shared with the ERP portal using inbound API and the data is inserted into the staging for validation checks before inserting the data into main ERP database. If an intruder tries to access the data in transit, whether inbound or outbound, he will get the unauthorized error and our data will stay secure.

9. Evaluating and Findings

Through the comparative analysis of the different integration methods, the efficiency and security of the proposed means of integrating cloud-based ERP systems and third-party applications have been demonstrated. Overall, both statistical analyses of integration points and response time revealed a robust positive correlation demonstrating that as integration increases in complexity, the response time of the system is impacted. The results indicated a high correlation coefficient and low p-value demonstrating that the relationship is statistically significant. Meaning, that if an organization intends to reduce latency on response time, it needs to factor in integration complexity. In terms of security, the proposed model outperforms existing models by adding AES-256 encryption, SSL/TLS protocols, multi-factor authentication, and role-based access control (RBAC). The t-test analysis of the effectiveness of the security using encryption between the proposed and blockchain-based ERP integration measurements confirms a statistically significant difference in measured security. The proposed model increases encryption standards for data that will allow for enhanced security and protection for the company from the vulnerabilities of traditional blockchain and API-based integration methods. The comparative analysis including manual integration, middleware integration, API-based integration, and point-to-point integration shows how the proposed methodology provides benefits in terms of automation, scalability, and data consistency. Despite the challenges of scalability and an error-prone process associated with manual and point-to-point integrations, middleware solutions and API-based integrations introduce latency and dependence on third-party applications. In contrast, the proposed integration methodology optimizes performance by minimizing the integration overhead while preserving high-performance security and system integrity. In measuring performance indicators, across blockchain-based, IoT-based, and proposed integration models, we have demonstrated that the proposed system has better security, fewer errors, and improved user satisfaction. A lower risk score and higher consistency ratio demonstrate that the proposed methodology is capable of maintaining data integrity across integrated applications. User satisfaction metrics additionally demonstrated usability scores and efficiency scores and thus the proposed solution can be justified as practical. In conclusion, this study shows that the suggested integration is a secure, scalable, and efficient method of cloud-based ERP for employees for practical uses. Through addressing identified limitations of existing integration methods by enhancing the performance measurements, risk or customer satisfaction measurements presented the suggested integration model is a secure, scalable, and efficient methodology for smooth integration of third-party systems, thus improving operational efficiency or data security or both in enterprise contexts.

10. Conclusion

In the contemporary world of changing technological demands, cloud-based ERP solutions have become a necessity for organizations wanting to easily store and manage their data. An ERP System serves as a virtual repository of business-critical information and is subject to a variety of security threats, which is why protecting the stored data must be the highest priority. In the event of a security breach within an ERP system, the implications are serious and significant, including the possibility of financial loss or damage to the organization's reputation. The security framework proposed is an effort to facilitate seamless, safe, and efficient integration of an ERP system with reputable third-style applications, and advocating for integration with any application. The limitations of the traditional approach to security will be addressed by providing a more thorough, comprehensive, and adaptable mechanism for assurance of data flow between two or more interconnected systems. A benefit of this security framework is the ability to group users based on their job tasks or responsibilities and enhances security by being able to utilize a unique customized source code security offering an efficient and easier approach to utilizing security for an ERP System in parallel with the third-party applications, which ultimately moves towards enhancing not only security, but systems performance and managing access control. Drawing attention to an existing state of vulnerabilities and providing a structured and secured integration will move the conversations forward of being proactive in the realm of security and assurance for ERP Systems. The integration of functional and technical customizations of coding provides organizations with options for ensuring that their preferred applications can potentially be implemented and secure while they exchange data with the ERP System.

The integration of functional and technical customizations using X++ code enabled the security implementation in the ERP system. API and badge initiation need to be established by the developer in the developer workspace whereas badge timeout needs to be configured by the system administrator in the functional workspace.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

11. Future Scope

The paper serves as a systematic overview of a flexible security framework to improve the interoperability of ERP solutions and third-party systems. This research addresses existing risks associated with cloud-based ERP software and assists a broad industry audience in addressing critical aspects of security solutions faster and better. Were presented primarily the common limitations and risks/enablers organizational contexts possess in and utilizing ERP portals and key gaps in research. We offer a security framework that protects ERP systems against cyber threats, denies potentially harmful intrusions, and provides a secure digital environment for organizations. Organizations can use a framework of this type to protect their data better, reduce integration vulnerabilities, and improve overall resilience to the ERP ecosystem.- Research - discussed providing theory, lessons learned, tools, methods, and techniques for the design, evaluation, operation, governance, and management of ERP systems while managing/enforcing the access control of systems and data. This research also represents an overview of current studies related to cloud security and is a summary of both functional and technical components of existing security frameworks. The article summarizes the current research limitations, challenges, and opportunities of the current research landscape. Meaningful contributions to the literature by providing coherent research for future investigators wanting to develop new solutions for organizations that can leverage cloud security mechanisms.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare that there are no conflicts of interest regarding the publication of this research."

References

[1] U. Malhotra, Ritu, and Amandeep, "Secure and Compatible Integration of Cloud-Based ERP Solution: A Review," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 9s, pp. 695–707, 2023.

[2] M. Patel and R. Kumar, "Challenges in Cloud-Based ERP Systems: A Review," *International Journal of Cloud Computing and Services Science*, vol. 13, no. 3, pp. 215-225, 2023.

[3] J. Doe and L. White, "Integrating ERP Systems with Cloud Technologies: A New Approach," *Journal of Software Engineering and Applications*, vol. 15, no. 4, pp. 300-310, 2022.

[4] P. Brown and S. Green, "E-invoicing Solutions for Cloud-Based ERP: A Case Study," *International Journal of Business Information Systems*, vol. 30, no. 2, pp. 120-135, 2021.

[5] U. Malhotra and R. Nagpal, "Secure and Compatible Integration of Cloud-Based ERP Solution: A Comprehensive Survey," in *International Conference on Applied Technologies (ICAT 2023)*, M. Botto-Tobar, M. Zambrano Vizuete, S. Montes León, P. Torres-Carrión, and B. Durakovic, Eds., Communications in Computer and Information Science, vol. 2051. Cham, Switzerland: Springer, 2024. doi: 10.1007/978-3-031-58950-8_17.

[6] A. Smith and B. Johnson, "A Comprehensive Review of Cybersecurity Frameworks for IoT Devices," *Journal of Information Security*, vol. 12, no. 1, pp. 45-58, 2023.

[7] A. S. Chellathurai, U. Malhotra, S. Thapasimuthu Rajeswari, and S. Thachankurichy Natesan, "Healthcare security in cloud-based wireless sensor networks: Botnet attack detection via autoencoder-aided goal-based artificial intelligent agent," *Concurrency Comput. Pract. Exp.*, vol. 36, no. 19, p. e8152, 2024, doi: 10.1002/cpe.8152.

[8] A. Faccia and P. Petratos, "Blockchain, Enterprise Resource Planning (ERP) and Accounting Information Systems (AIS): Research on e-Procurement and System Integration," *Appl. Sci.*, vol. 11, no. 15, p. 6792, 2021.

[9] Baskar, S., Mohamed Shakeel, P., Kumar, R., Burhanuddin, M.A., & Sampath, R., "A dynamic and interoperable communication framework for controlling the operations of wearable sensors in smart healthcare applications," *Computer Communications*, vol. 149, pp. 17-26, 2020.

[10] F. Mahmood, A. Z. Khan, and R. H. Bokhari, "ERP Issues and Challenges: A Research Synthesis," *Kybernetes*, vol. 49, no. 3, pp. 629–659, 2020.

[11] Srinivasan, V., Singh, J., Pandi-Perumal, S.R., Brown, G.M., Spence, D.W., & Cardinali, D.P., "Jet lag, circadian rhythm sleep disturbances, and depression: The role of melatonin and its analogs," *Advances in Therapy*, vol. 27, no. 11, pp. 796-813, 2010.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025

[12] Shukla P.K., Roy V., Chandanan A.K., Sarathe V.K., Mishra P.K., "A Wavelet Features and Machine Learning Founded Error Analysis of Sound and Trembling Signal," *SN Computer Science*, vol. 4, no. 6, art. no. 717, 2023, doi: 10.1007/s42979-023-02189-y.

[13] Selvam, C., Prabu, S.L., Jordan, B.C., Purushothaman, Y., Umamaheswari, A., Hosseini Zare, M.S. &, Thilagavathi, R., "Molecular mechanisms of curcumin and its analogs in colon cancer prevention and treatment," *Life Sciences*, vol. 239, 2019.

[14] R. Hrischev, "ERP Systems and Data Security," in *Mater. Sci. Eng.*, 9th Int. Sci. Conf., 2020.

[15] Jayachitra, S., & Prasanth, A., "Multi-Feature Analysis for Automated Brain Stroke Classification Using Weighted Gaussian Naïve Bayes Classifier," *Journal of Circuits, Systems and Computers*, vol. 30, no. 10, 2021.

[16] T. Anderson and V. Lee, "IoT Solutions for Healthcare: Challenges and Opportunities," *Journal of Health Informatics*, vol. 8, no. 2, pp. 100-115, 2022.

[17] R. Kumaraswamy, S. Latif, and T. Mather, "Chapter 7: Privacy," in *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, 1st ed., O'Reilly Media, 2009, p. 145.

[18] Kalaiselvi, C., & Nasira, G.M., "A new approach for diagnosis of diabetes and prediction of cancer using ANFIS," in *2014 Proceedings, World Congress on Computing and Communication Technologies*, WCCCT 2014, pp. 188-190.

[19] Jasti, V.D.P., Zamani, A.S., Arumugam, K., Naved, M., Pallathadka, H., Sammy, F., Raghuvanshi, A., & Kaliyaperumal, K., "Computational Technique Based on Machine Learning and Image Processing for Medical Image Analysis of Breast Cancer Diagnosis," *Security and Communication Networks*, 2022.

[20] Dey, N., Ashour, A.S., Beagum, S., Pistola, D.S., Gospodinov, M., Gospodinova, E.P., & Tavares, J.M.R., "Parameter optimization for local polynomial approximation based intersection confidence interval filter using genetic algorithm: An application for brain MRI image de-noising," *Journal of Imaging*, vol. 1, no. 1, pp. 60-84, 2015.

[21] Sasi, S.B., & Sivanandam, N., "A survey on cryptography using optimization algorithms in WSNs," *Indian Journal of Science and Technology*, vol. 8, no. 3, pp. 216-221, 2015.

[22] Sahu P., Viji A.J., Roy V., Roy L., Manogna D., Vasal S., "A Comprehensive Framework for Evaluating Cyber-Physical Threats in Energy Internet," in *2024 International Conference on Intelligent Systems and Advanced Applications, ICISAA 2024*, 2024, doi: 10.1109/ICISAA62385.2024.10828794.

[23] Yacin Sikkandar, M., Alrasheadi, B.A., Prakash, N.B., Hemalakshmi, G.R., & Mohanarathinam, A., "Deep learning based an automated skin lesion segmentation and intelligent classification model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3245-3255, 2021.

[24] S. Mandal and D. A. Khan, "A Study of Security Threats in Cloud: Passive Impact of COVID-19 Pandemic," in *Proc. Int. Conf. Smart Electron. Commun. (ICOSEC)*, 2020, pp. 837–842.

[25] R. Smith and K. Taylor, "Real-Time Data Processing in Cloud Environments for Healthcare Applications," *International Journal of Cloud Computing and Services Science*, vol. 14, no. 1, pp. 75-85, 2023.

[26] R. Hrischev, "ERP Systems and Data Security," *Mater. Sci. Eng.*, 9th Int. Sci. Conf., 2020.

[27] S. O. Kuyoro, F. Ibikunle, and O. Awodele, "Cloud Computing Security Issues and Challenges," *Int. J. Comput. Netw. (IJCN)*, vol. 3, no. 5, pp. 247–255, 2011.

DOI: https://doi.org/10.54216/JCIM.160205
Received: January 31, 2025 Revised: February 25, 2025 Accepted: April 05, 2025