Unifying Industrial IoT and SAP S/4HANA Data Ecosystems Through Low-Latency Event Propagation, Transactional Consistency, and Scalable Cloud Integration Architectures

Rajendra Asuri
Senior SAP integration consultant

Abstract
Industrial Internet of Things (IIoT) environments generate continuous operational data that must be integrated with enterprise resource planning systems to support timely decisions, reliable transactions, and coordinated manufacturing activities. However, heterogeneous device protocols, communication delays, message duplication, network interruptions, and differences in processing capacity complicate the integration of industrial systems with SAP S/4HANA. This study develops and examines a unified integration architecture that combines SAP Business Technology Platform (SAP BTP), SAP Event Mesh, and event-driven communication to address these challenges. A mixed-method approach is adopted, bringing together architectural analysis, qualitative assessment of integration mechanisms, and quantitative evaluation criteria for event latency, message delivery reliability, transactional consistency, and processing scalability. The proposed framework separates industrial data acquisition, event normalization, asynchronous message distribution, and enterprise transaction execution. It incorporates event identification, acknowledgment, retry handling, idempotent processing, and transaction reconciliation to reduce duplicate execution and preserve data integrity across distributed components. The analysis indicates that decoupled event processing can reduce direct system dependencies, improve fault isolation, and provide a more adaptable basis for handling variable industrial workloads, while consistency controls remain essential for dependable enterprise updates. The study contributes an integrated architectural perspective connecting low-latency communication with transaction management and scalable cloud integration. Its findings offer design guidance for SAP architects, manufacturing organizations, and researchers seeking interoperable, resilient IoT-to-ERP environments.

Keywords: Industrial IoT (IIoT), IoT Security, Real-Time Data Synchronization, SAP S/4HANA, SAP BTP, Event-Driven Architecture, SAP Event Mesh, Enterprise Integration, Data Security, Identity and Access Management, Edge-to-Cloud Integration, Distributed Messaging, Asynchronous Event Processing, Transactional Consistency, Low-Latency Communication, Cloud-Native Architecture, Message Delivery Reliability, API Integration, Fault-Tolerant Systems, Data Integrity, Enterprise Interoperability.
DOI: 10.18091/ijsts.v6i1.08

1. Introduction

Industrial Internet of Things (IIoT) has significantly influenced the development of modern industrial environments by connecting sensors, machines, programmable logic controllers, gateways, and enterprise applications through interconnected communication networks. The increasing adoption of connected industrial equipment has enabled organizations to collect operational information continuously, monitor production activities, and improve resource utilization. This development has gradually shifted

manufacturing environments from isolated automation systems toward integrated industrial ecosystems in which operational technology and information technology exchange information to support coordinated business activities. Enterprise resource planning (ERP) platforms have consequently become important components of industrial digitalization, providing centralized capabilities for managing production, procurement, inventory, maintenance, and financial transactions. Integrating industrial data with enterprise applications therefore represents an important requirement for establishing operational visibility and improving organizational responsiveness.

SAP S/4HANA provides an enterprise platform for coordinating business transactions and operational processes across manufacturing and supply chain environments. Its in-memory data processing capabilities, simplified data model, and integrated application architecture support the management of enterprise information generated by different organizational functions. However, industrial environments produce continuous streams of machine readings, equipment conditions, production measurements, and process notifications that must be transferred into enterprise applications without introducing unnecessary communication delays. Synchronizing these operational events with SAP S/4HANA enables organizations to maintain closer coordination between physical production activities and enterprise-level business processes. The integration becomes particularly important when equipment conditions influence maintenance notifications, production confirmations, material availability, and inventory updates. Consequently, establishing dependable communication between industrial devices and SAP S/4HANA requires an integration architecture capable of accommodating continuous data exchange and varying operational workloads.

Despite these advantages, connecting heterogeneous industrial devices with enterprise applications introduces considerable technical difficulties. Industrial equipment frequently operates through different communication protocols, data representations, network configurations, and processing capabilities. Protocols such as OPC UA, MQTT, Modbus, and HTTP may coexist within the same industrial environment, creating interoperability requirements across device and application boundaries. Communication delays, intermittent network connectivity, message duplication, and incomplete data transmission can further affect synchronization reliability. Differences in processing speed between industrial devices and enterprise transaction systems may also produce inconsistent representations of operational information. These conditions become increasingly significant when multiple production facilities generate events simultaneously and require timely processing. An integration mechanism must therefore address protocol compatibility, message transportation, data transformation, and reliable delivery while maintaining the integrity of enterprise transactions.

Conventional point-to-point integration and synchronous communication approaches provide established mechanisms for connecting industrial applications with enterprise systems. Nevertheless, these approaches can introduce architectural dependencies, increased communication overhead, and limited flexibility when the number of connected devices and applications expands. Synchronous requests require participating systems to remain available during communication, potentially increasing response times when enterprise applications experience processing delays. Direct connections between individual devices, gateways, and business applications may also complicate interface maintenance and introduce additional dependencies across integrated environments. Furthermore, failures in intermediate communication components can interrupt information exchange and require repeated transmission attempts. Such limitations indicate that conventional integration arrangements may not adequately accommodate the continuous, distributed, and event-intensive characteristics of industrial IoT environments. A more flexible integration approach is required to separate event generation from enterprise transaction processing while preserving dependable information exchange.

Event-driven architecture offers an alternative communication model by allowing applications and connected devices to exchange information through events rather than relying exclusively on direct synchronous requests. Within this approach, industrial events can be published to messaging infrastructure, routed through defined topics, and consumed by authorized enterprise applications

according to their operational requirements. SAP Business Technology Platform (SAP BTP) provides integration capabilities that can support communication between industrial environments and SAP S/4HANA, while SAP Event Mesh offers message-based event distribution and asynchronous communication mechanisms. This architectural separation reduces direct dependencies between event producers and consumers and provides opportunities for independent processing and horizontal scalability. However, asynchronous communication alone does not guarantee transactional consistency, message ordering, or duplicate prevention. Reliable integration consequently requires complementary mechanisms involving message acknowledgment, persistent delivery, idempotent processing, event identification, and transaction reconciliation.

The central research problem addressed in this study concerns the absence of a sufficiently coordinated integration approach that simultaneously considers low-latency event propagation, transactional consistency, and scalability across Industrial IoT and SAP S/4HANA environments. Existing integration arrangements may prioritize communication speed without adequately addressing data integrity, or emphasize transaction reliability while introducing additional processing delays. Similarly, increasing the number of connected devices can create message congestion, uneven processing workloads, and difficulties in maintaining consistent enterprise records. This study is motivated by the need to examine how event-driven communication through SAP BTP and messaging infrastructure can address these interconnected challenges. The research investigates the architectural relationships between industrial event acquisition, asynchronous message distribution, enterprise data processing, and transaction validation. Particular attention is given to the conditions under which distributed communication can remain responsive and reliable despite network interruptions and variations in processing capacity.

The primary objective of this research is to develop and examine a unified integration architecture that connects Industrial IoT environments with SAP S/4HANA through low-latency event propagation and scalable cloud integration mechanisms. The proposed approach focuses on evaluating the role of SAP BTP and SAP Event Mesh in facilitating asynchronous communication, reducing direct application dependencies, and coordinating distributed event processing. Additional objectives include examining message delivery reliability, identifying mechanisms for maintaining transactional consistency, and assessing the influence of increasing event volumes on integration performance. The investigation is guided by three principal research questions: How can event-driven communication reduce synchronization delays between industrial devices and SAP S/4HANA? Which message-processing and transaction-management mechanisms can preserve data integrity across distributed environments? How can the proposed integration architecture accommodate increasing device connectivity and event-processing workloads without compromising reliability? These questions establish the analytical scope of the study and guide the evaluation of its architectural contribution.

The significance of this research extends beyond establishing connectivity between industrial devices and enterprise applications. A unified integration approach can support more responsive manufacturing operations by improving the availability of equipment information, production events, and operational measurements within enterprise systems. Reliable synchronization can also strengthen coordination among production planning, inventory management, equipment maintenance, and supply chain functions. From an architectural perspective, the study contributes an integrated perspective on event transportation, distributed messaging, transaction consistency, and cloud-based enterprise connectivity. Its practical relevance lies in identifying design considerations that SAP architects, industrial automation specialists, and enterprise integration professionals can apply when developing connected industrial environments. The proposed framework also provides a foundation for examining the relationship between communication performance and enterprise data reliability under different operational conditions, without assuming that asynchronous integration automatically eliminates consistency or availability challenges.

The research ultimately argues that effective Industrial IoT integration with SAP S/4HANA requires a balanced architectural approach in which communication efficiency, transactional correctness, and system scalability are treated as interconnected design requirements. The proposed framework brings together

industrial data acquisition, event-driven messaging, SAP BTP integration services, and enterprise transaction processing within a coordinated integration model. Its contribution is directed toward establishing clearer architectural principles for handling continuous industrial events while maintaining dependable enterprise information. By examining both communication mechanisms and transaction-management requirements, the study seeks to provide a technically grounded foundation for reliable industrial data synchronization. The resulting analysis is intended to support enterprise integration planning, connected manufacturing initiatives, and further academic investigation into distributed industrial information systems.

2. Industrial IoT and Enterprise Integration Foundations

Industrial Internet of Things (IIoT) has established a connected industrial environment in which physical equipment, sensing devices, communication gateways, and enterprise applications exchange operational information. Sensors installed across manufacturing equipment continuously capture parameters such as temperature, pressure, vibration, humidity, rotational speed, and production output. These measurements provide essential information about equipment conditions, production activities, and operational efficiency. Programmable logic controllers (PLCs) coordinate machine operations, while supervisory control and data acquisition (SCADA) systems monitor and manage industrial processes across production facilities. Industrial gateways collect information from connected devices, aggregate measurements, and facilitate communication between field-level equipment and higher-level applications. This interconnected arrangement provides the technical foundation for integrating operational technology with enterprise information systems.

Industrial IoT architecture generally consists of interconnected device, control, communication, edge-processing, and application environments. At the equipment level, sensors and actuators interact with physical processes, whereas industrial controllers execute predefined operational instructions. Edge gateways provide an intermediate processing layer by collecting information from multiple devices, filtering unnecessary measurements, and translating communication protocols. This arrangement reduces network traffic and supports local processing when connectivity with remote systems becomes unavailable. However, differences in device capabilities, communication standards, sampling frequencies, and data formats create considerable interoperability challenges. Industrial information must retain its accuracy and operational context when transferred between heterogeneous environments. Consequently, enterprise integration requires mechanisms capable of coordinating device-level information with centralized business applications without compromising data quality or communication reliability.

SAP S/4HANA provides a central enterprise transaction and operational data platform for coordinating business activities across manufacturing, procurement, inventory management, maintenance, and financial operations. Its in-memory computing capabilities and simplified data architecture support integrated transaction processing across interconnected business functions. Industrial equipment generates operational information that can directly influence enterprise activities, including production confirmations, equipment maintenance notifications, material consumption, and inventory updates. Connecting these events with SAP S/4HANA establishes a relationship between physical manufacturing activities and corresponding enterprise records. Such integration reduces dependence on manual information entry and delayed batch-based transfers. Maintaining synchronization between industrial events and enterprise transactions is therefore essential for improving operational visibility, supporting accurate business records, and coordinating manufacturing activities with organizational requirements.

SAP Business Technology Platform provides capabilities for enterprise application development, integration, connectivity, and data exchange across distributed application environments. For the historical terminology applicable to this research, SAP Cloud Platform is the appropriate platform designation for the period before June 2019. Its integration capabilities supported communication between SAP

applications, external systems, and enterprise environments through standardized interfaces and middleware services. REST-based services, OData interfaces, and web service protocols provide mechanisms for exchanging structured business information across application boundaries. Integration services additionally support message transformation, routing, and connectivity between industrial data sources and enterprise applications. These capabilities establish an integration layer through which industrial information can be processed and delivered to SAP S/4HANA. The separation of connectivity services from business transaction processing also provides greater flexibility when integrating heterogeneous industrial systems.

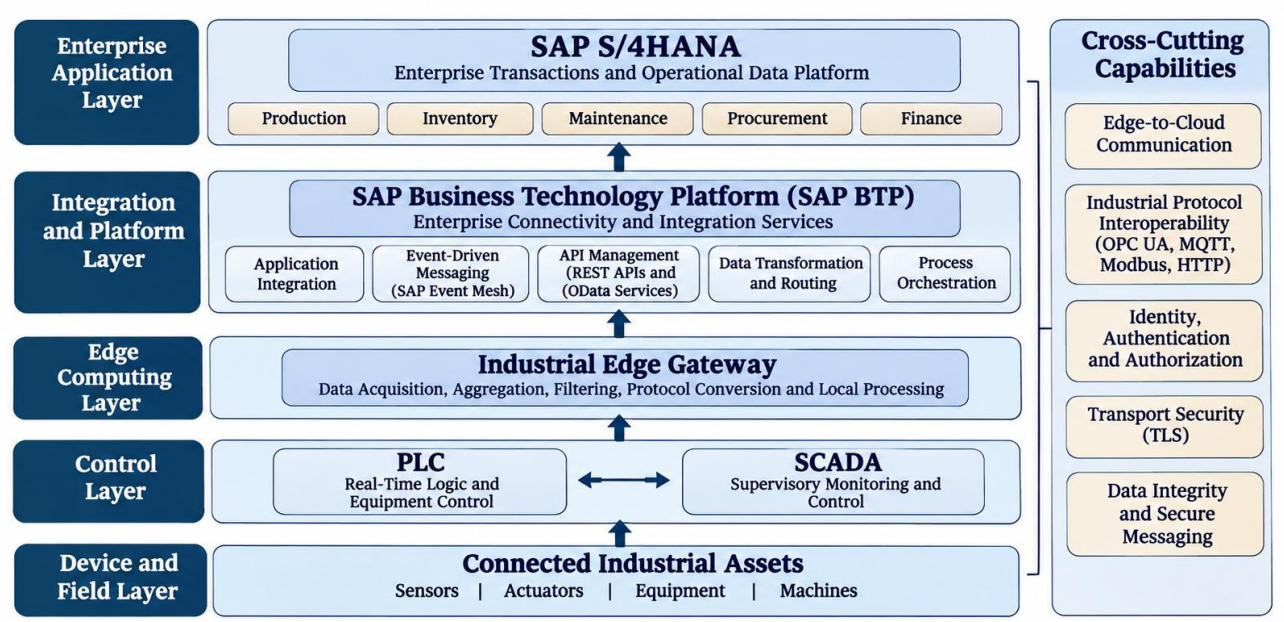

Figure 1: Layered Industrial IoT and SAP S/4HANA Enterprise Integration Ecosystem

Event-driven architecture introduces a communication approach in which applications exchange information through events representing significant operational occurrences. In industrial environments, such events may describe machine condition changes, production milestones, equipment warnings, temperature variations, or material movements. Event producers generate messages when relevant activities occur, while independent consumers receive and process information according to their operational responsibilities. This separation reduces direct dependencies between connected applications and allows individual components to operate independently. Unlike synchronous communication, asynchronous messaging does not require the event producer to wait for every receiving application to complete its processing. Such characteristics are particularly relevant to industrial environments where numerous devices generate information continuously and enterprise applications process incoming events at different rates. Appropriate event identification, routing, and processing mechanisms are nevertheless necessary to maintain reliable communication.

Message-oriented middleware provides the infrastructure required to distribute events between independent producers and consumers. SAP Cloud Platform Enterprise Messaging represents the historical messaging terminology relevant to the selected publication period, whereas SAP Event Mesh is the later product designation and should not be retrospectively treated as a product name established before June 2019. Messaging infrastructure enables asynchronous information exchange through queues, topics, subscriptions, and message-routing mechanisms. Event distribution allows multiple enterprise applications to consume relevant industrial information without requiring individual direct connections to

every device or gateway. Message acknowledgment, persistence, retry mechanisms, and controlled consumption contribute to reliable event transportation. Nevertheless, successful message delivery does not independently establish successful enterprise transaction completion. Application-level controls are therefore necessary to address duplicate processing, transaction validation, event correlation, and data reconciliation across distributed systems.

Edge-to-cloud communication establishes a distributed processing arrangement in which industrial information is collected and evaluated close to its source before selected data is transmitted to enterprise applications. Edge gateways can perform filtering, aggregation, protocol conversion, and preliminary validation, reducing unnecessary transmission of repetitive measurements. Industrial communication protocols such as OPC UA, MQTT, Modbus, and HTTP support information exchange across heterogeneous equipment and application environments. OPC UA provides structured industrial communication and information modelling, whereas MQTT supports lightweight publish-subscribe messaging suitable for constrained network environments. Differences in communication characteristics and data representations require appropriate protocol translation and normalization. Consistent device identifiers, timestamps, measurement units, and event structures help preserve the meaning of industrial information during transportation. This coordination between edge processing and cloud-based integration is important for maintaining dependable information exchange across geographically distributed industrial environments.

Security constitutes an essential requirement when industrial information moves across interconnected operational and enterprise environments. Connected devices, communication gateways, messaging services, and enterprise interfaces introduce multiple access points requiring appropriate protection. Authentication mechanisms establish the identity of communicating devices and applications, while authorization determines which resources and operations authenticated entities can access. Transport Layer Security (TLS) protects information during transmission, and certificate-based authentication provides a mechanism for establishing trust between communicating components. Identity and access management policies should additionally address credential management, permission allocation, and separation of operational responsibilities. Device provisioning, certificate validation, interface protection, and access monitoring further contribute to a controlled integration environment. These security mechanisms support the confidentiality, integrity, and availability of industrial information exchanged between connected equipment and enterprise applications, providing a necessary foundation for reliable enterprise integration.

Table 1: Comparative Assessment of Conventional and Event-Driven Integration Approaches

Evaluation criteria	**Conventional integration**	**Event-driven integration**
Communication model	Synchronous request-response	Asynchronous event exchange
System architecture	Point-to-point connectivity	Decoupled producer-consumer architecture

Data transmission	Request-triggered transfer	Event-triggered distribution
Processing dependency	Sender waits for receiver response	Sender and receiver process independently
Message distribution	Direct interface communication	Message broker and topic-based routing
System scalability	Additional connections increase complexity	Independent scaling of event consumers
Failure management	Request retries and interface recovery	Message persistence, acknowledgment, and retry
Data consistency	Transaction-dependent synchronization	Event validation and reconciliation
Protocol interoperability	Requires interface-specific adaptation	Supports integration through messaging and protocol translation
Industrial applicability	Suitable for limited system connections	Suitable for distributed industrial environments

3. Architectural Design of the Unified IoT–SAP Integration Framework

The proposed integration framework establishes a unified architectural environment for connecting Industrial Internet of Things (IIoT) infrastructure with SAP S/4HANA through SAP Business Technology Platform (SAP BTP) and asynchronous event communication. The architecture is designed to transfer operational information from heterogeneous industrial devices into enterprise applications while maintaining communication efficiency, data integrity, and processing reliability. It separates industrial data acquisition, gateway-level processing, cloud integration, event transportation, and enterprise transaction execution into distinct functional layers. Each layer performs defined responsibilities while exchanging information through standardized interfaces and controlled communication mechanisms. This separation minimizes direct dependencies between operational equipment and enterprise applications, allowing individual components to be maintained, configured, and expanded independently. The framework consequently provides a structured foundation for continuous industrial information exchange across distributed manufacturing environments.

The industrial device and gateway communication layer represents the primary source of operational information within the proposed framework. Sensors, actuators, programmable logic controllers, and supervisory control systems generate measurements and process notifications associated with manufacturing activities. These devices communicate through industrial protocols such as OPC UA, MQTT, Modbus, and HTTP, depending on equipment capabilities and network requirements. Industrial gateways collect information from multiple sources and perform preliminary operations, including protocol conversion, data aggregation, filtering, and timestamp assignment. These operations establish a consistent representation of industrial information before it enters the enterprise integration environment. Positioning gateway services between field-level equipment and cloud applications reduces direct connectivity requirements and supports controlled communication across heterogeneous industrial networks.

The cloud integration and event-processing layer, implemented through SAP BTP, coordinates communication between industrial gateways, messaging infrastructure, and SAP S/4HANA. SAP BTP provides integration capabilities for connecting external applications, transforming structured information, and managing communication between distributed systems. Incoming industrial data can be validated, normalized, and converted into enterprise-compatible message structures before being forwarded to the appropriate processing component. Standardized interfaces and reusable integration flows reduce the need for individually developed connections between every device and business application. This arrangement also provides a centralized environment for managing connectivity, interface configuration, and integration-related processing activities. The integration layer therefore acts as an intermediary between industrial data sources and enterprise applications while supporting controlled information exchange.

Asynchronous messaging forms a central component of the proposed integration framework by separating event generation from enterprise application processing. SAP Event Mesh provides message-based communication between connected industrial gateways and authorized consuming applications. Its messaging capabilities support topic-based distribution, allowing operational events to be categorized according to their purpose, equipment identity, production area, or event classification. Independent consumers can subscribe to relevant topics without requiring direct communication with individual event producers. This arrangement reduces application coupling and distributes message-processing responsibilities across multiple enterprise components. By separating event publication from consumer execution, the messaging infrastructure supports independent processing and provides flexibility when industrial event volumes or enterprise application workloads fluctuate.

Event routing and subscription mechanisms determine how operational information moves through the integration environment. Each event should contain sufficient contextual information, including a unique event identifier, device reference, timestamp, event category, and associated measurement or operational status. The messaging infrastructure uses configured topics and subscriptions to distribute relevant messages to appropriate consumers. For example, an equipment condition notification may be directed toward a maintenance-related processing component, whereas a production measurement may be delivered to an enterprise application responsible for production reporting. Message filtering and routing reduce unnecessary processing by ensuring that consumers receive information relevant to their responsibilities. Subscription management also allows additional applications to participate in the integration environment without requiring modifications to every existing event producer.

Connectivity with SAP S/4HANA is established through standardized application programming interfaces and controlled enterprise integration mechanisms. REST-based services, OData interfaces, and other supported APIs provide communication channels for transferring validated operational information into enterprise processes. Before transaction execution, incoming events undergo structural validation, identifier verification, and relevant business-rule evaluation. The integration component translates industrial measurements and notifications into information structures recognized by the receiving enterprise application. Depending on the business requirement, processed information may support production confirmations, equipment maintenance notifications, inventory updates, or operational

reporting. Separating interface communication from transaction execution helps prevent device-level communication activities from becoming directly dependent on enterprise application availability.

The proposed architecture distinguishes operational data acquisition, event processing, and enterprise transaction execution to improve fault isolation and processing independence. Industrial devices continue collecting measurements according to their configured operating requirements, while gateways manage preliminary processing and communication with the integration environment. Messaging components distribute events according to configured delivery and subscription mechanisms, allowing receiving applications to process information independently. Where temporary network failures occur, message acknowledgment, retry policies, and controlled recovery procedures help manage interrupted communication, subject to the capabilities and configuration of the messaging infrastructure. Duplicate event handling and transaction reconciliation must also be addressed to prevent repeated processing from producing inconsistent enterprise records. These mechanisms contribute to operational reliability, although their effectiveness depends on appropriate configuration and coordination across all architectural layers.

Modularity, interoperability, scalability, and fault isolation guide the overall design of the unified integration framework. Modular components allow device communication, gateway processing, messaging services, and enterprise interfaces to be modified without requiring extensive changes throughout the architecture. Interoperability is supported through protocol conversion, standardized message structures, and reusable application interfaces that accommodate heterogeneous industrial and enterprise environments. Scalability depends on the ability to accommodate increasing device connectivity, message volumes, and consumer workloads through suitable resource allocation and messaging configuration. Fault isolation limits the effect of individual component failures by separating communication responsibilities and providing controlled recovery mechanisms. The resulting architecture establishes a structured approach to connecting industrial operations with SAP S/4HANA while balancing event propagation efficiency, enterprise transaction reliability, and long-term integration flexibility.

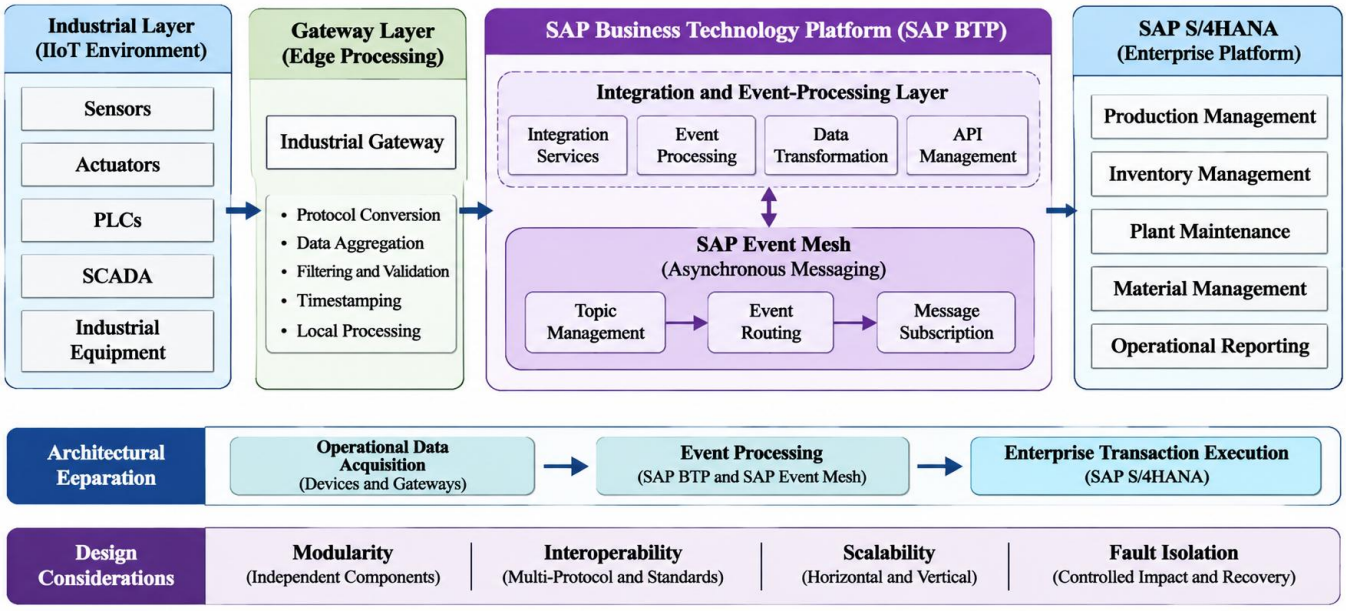

Figure 2: Proposed Unified IoT–SAP S/4HANA Integration Architecture

4. Low-Latency Event Propagation and Real-Time Data Synchronization

Industrial IoT environments generate continuous operational events through sensors, actuators, programmable logic controllers, and connected manufacturing equipment. These events represent changes in equipment conditions, production measurements, machine operating states, temperature variations, vibration levels, and process notifications. The frequency and volume of generated information depend on equipment characteristics, production requirements, and configured sampling intervals. Industrial gateways collect these measurements and establish communication between field-level devices and enterprise integration infrastructure. However, heterogeneous devices frequently differ in communication protocols, message structures, timestamp formats, and data transmission capabilities. These differences create additional processing requirements before operational information can be transferred into SAP S/4HANA. Reliable event acquisition therefore requires a coordinated mechanism for identifying, collecting, and preparing industrial information without introducing unnecessary communication overhead or compromising the accuracy of the original measurements.

Event transformation and normalization establish consistency across operational information originating from different industrial sources. Measurements collected through OPC UA, MQTT, Modbus, and HTTP interfaces may contain different representations of equipment identifiers, measurement units, timestamps, and operational status values. The gateway processing layer converts these variations into standardized event structures that can be interpreted by downstream integration services. Each event should contain essential attributes such as a unique event identifier, source device reference, event timestamp, event category, and associated measurement values. Message serialization converts the normalized information into a structured representation suitable for transportation across distributed communication environments. JSON and XML provide commonly used formats for exchanging structured application information, depending on interface requirements. Consistent serialization and validation reduce ambiguity during message interpretation and establish a reliable foundation for subsequent enterprise processing.

Asynchronous event propagation provides the principal communication mechanism for transferring industrial information across the proposed integration framework. Rather than requiring industrial devices to communicate directly with SAP S/4HANA for every operational occurrence, gateway services publish validated events to messaging infrastructure. For the historical technology context before June 2019, SAP Event Mesh is the appropriate terminology for the messaging capability associated with the later SAP Event Mesh offering. This infrastructure supports message-based communication between independent producers and consumers, reducing the need for persistent direct connections between industrial equipment and enterprise applications. Event producers can continue collecting and publishing information without waiting for receiving applications to complete their processing activities. This separation is particularly valuable when industrial devices generate events at a higher frequency than enterprise applications can immediately consume them.

Topic-based event routing organizes industrial information according to its operational purpose and intended receiving application. Topics can represent equipment categories, production areas, manufacturing processes, or specific event classifications. For example, machine condition notifications may be directed toward maintenance-related processing services, while production measurements may be delivered to applications responsible for production reporting and operational monitoring. Subscription mechanisms allow authorized consumers to receive information associated with their assigned responsibilities without establishing separate communication channels with every device. Distributed message consumption also enables multiple enterprise components to process different categories of operational information independently. Appropriate topic definitions and subscription configurations reduce unnecessary message distribution and support controlled information exchange. This arrangement improves architectural flexibility because additional consumers can be introduced without requiring extensive modifications to existing event producers.

Real-time synchronization between industrial environments and SAP S/4HANA depends on the coordinated execution of event acquisition, message transportation, validation, and enterprise transaction processing. Once an event reaches the integration environment, its structure and contextual information must be verified before the corresponding enterprise operation is initiated. Integration services can transform industrial measurements into application-compatible information and invoke standardized enterprise interfaces, including supported REST-based and OData services. Depending on the event category, the receiving application may initiate production confirmations, equipment maintenance notifications, inventory updates, or operational status changes. The synchronization process must preserve the relationship between the original industrial occurrence and the resulting enterprise transaction. Event timestamps, device identifiers, and transaction references provide important contextual information for maintaining traceability throughout the processing sequence. This coordinated approach reduces dependence on delayed information transfers while supporting more timely enterprise visibility.

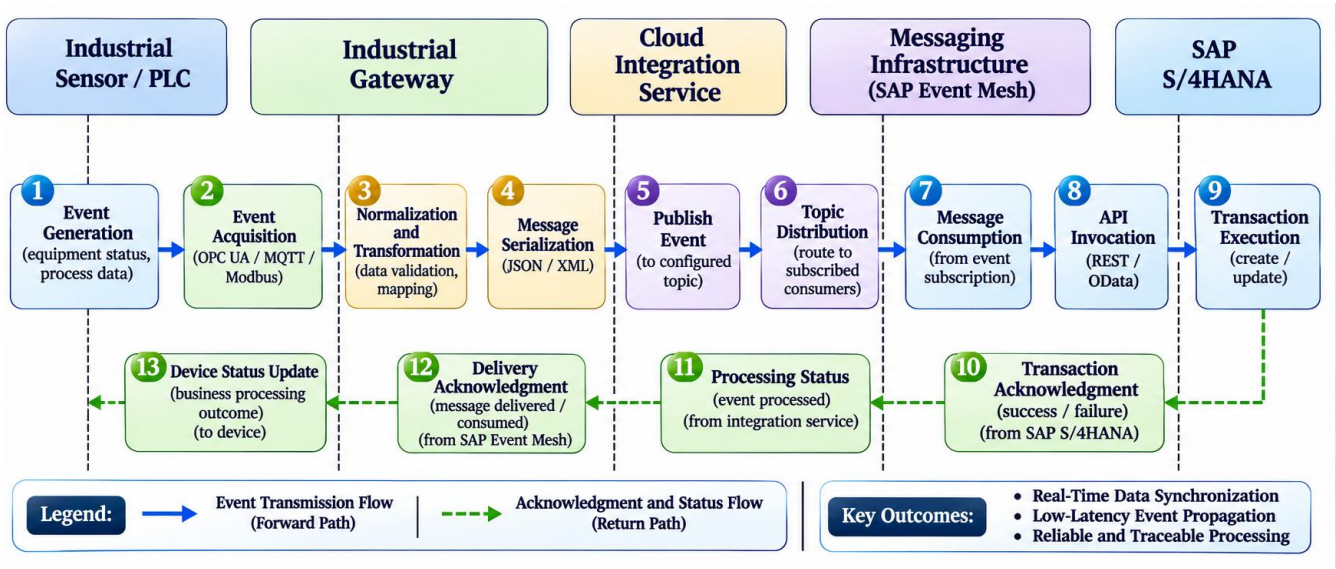

Figure 3: End-to-End Industrial Event Propagation and Synchronization Sequence for IoT–SAP S/4HANA Integration

Latency reduction depends on minimizing unnecessary communication dependencies and distributing processing responsibilities across appropriate architectural components. Conventional synchronous integration can introduce waiting periods when a receiving application is unavailable or requires additional processing time before responding. Asynchronous messaging separates event publication from consumer execution, allowing communication and application processing to proceed independently. Gateway-level filtering and aggregation can further reduce the volume of information transmitted to cloud integration services by eliminating repetitive measurements that do not require individual enterprise processing. Message routing and independent consumer execution also prevent unrelated application workloads from unnecessarily interfering with one another. Nevertheless, reduced communication dependency does not automatically guarantee low end-to-end latency. Network conditions, message queue depth, serialization overhead, broker processing capacity, and enterprise interface response times continue to influence overall synchronization performance.

Reliable event propagation additionally requires mechanisms for maintaining message ordering, preventing duplicate processing, and confirming successful delivery. Industrial events may arrive out of sequence because of network variations, gateway processing delays, or differences in communication paths. Event timestamps, sequence identifiers, and source-specific ordering information can assist

receiving applications in determining the appropriate processing sequence. Duplicate detection is equally important because message retries and repeated transmissions may cause the same operational event to reach an enterprise application more than once. Idempotent processing mechanisms allow repeated event submissions to be identified and handled without generating unintended duplicate business transactions. Delivery acknowledgment provides information about message receipt, while retry policies support recovery from temporary communication failures. Persistent messaging and controlled error handling further contribute to dependable event transportation, although transaction completion must be verified separately from message delivery.

Network interruptions, processing delays, and message congestion remain significant challenges in distributed industrial synchronization environments. Temporary connectivity failures can interrupt communication between industrial gateways, cloud integration services, and enterprise applications, resulting in delayed event transmission. Increasing event volumes may also create message backlogs when consumer processing capacity becomes insufficient to handle incoming information. These conditions can increase synchronization time and create differences between actual equipment conditions and corresponding enterprise records. Controlled buffering, message persistence, configurable retry intervals, consumer workload management, and dead-letter handling provide mechanisms for managing such situations. Monitoring queue depth, processing duration, failed deliveries, and transaction completion status can help identify emerging bottlenecks. Effective recovery procedures must also include reconciliation mechanisms to identify events that were delayed, rejected, or incompletely processed during communication disruptions.

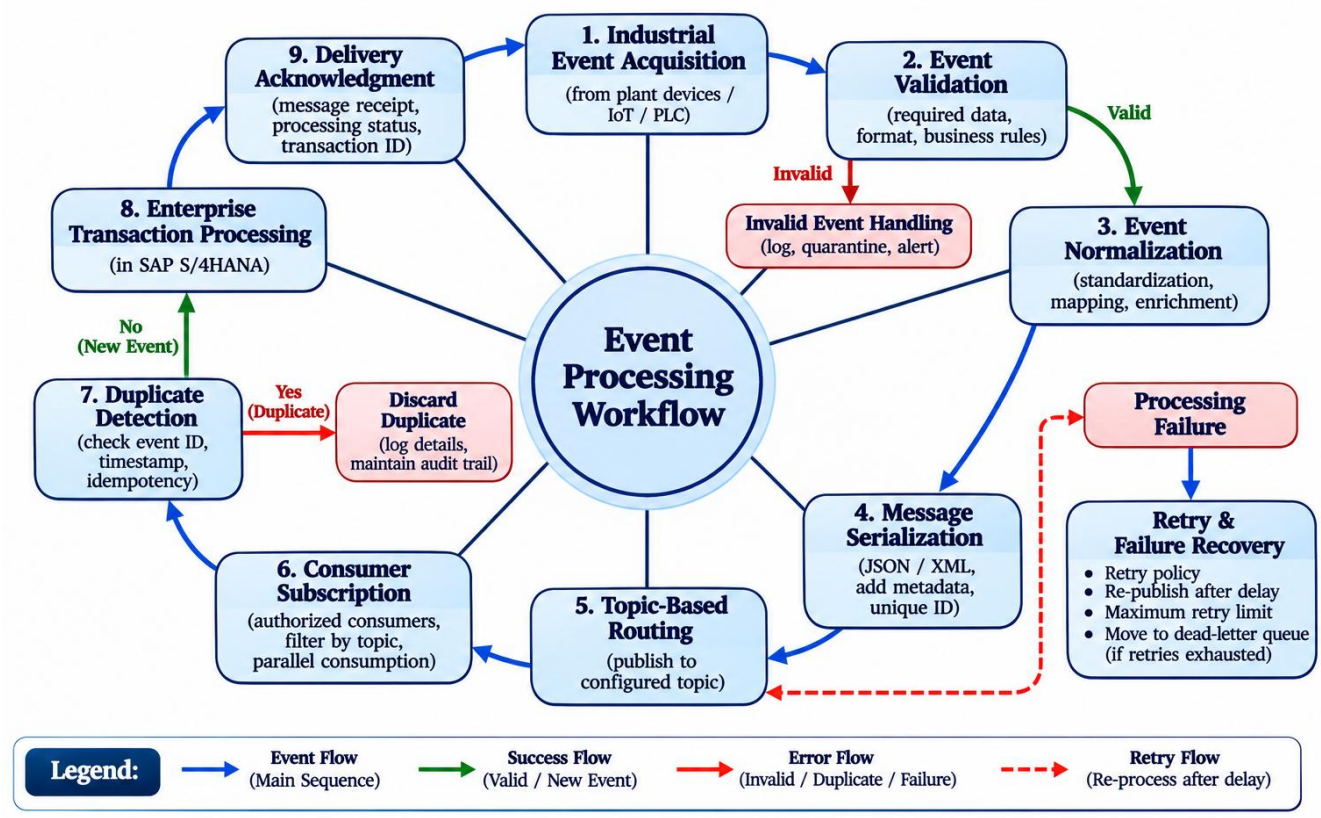

Figure 4: Distributed Event Routing and Message Processing Workflow

The proposed event propagation approach establishes a coordinated mechanism for transferring industrial information into SAP S/4HANA while balancing communication responsiveness and processing reliability. Its principal contribution lies in combining standardized event acquisition, structured message transformation, asynchronous distribution, and controlled enterprise transaction execution within a unified

synchronization workflow. The separation of event producers and consumers provides flexibility when industrial workloads increase or individual enterprise applications experience temporary processing delays. Message identification, acknowledgment, duplicate detection, and recovery mechanisms further support the preservation of operational information throughout distributed communication. However, synchronization performance remains dependent on the configuration of messaging infrastructure, network capacity, gateway processing, and enterprise interface responsiveness. These considerations provide the basis for evaluating the proposed architecture against measurable indicators of event propagation latency, message throughput, delivery reliability, and synchronization consistency.

5. Transactional Consistency, Data Integrity, and Integration Reliability

Transactional consistency represents a fundamental requirement when integrating Industrial Internet of Things (IIoT) environments with enterprise resource planning systems such as SAP S/4HANA. Industrial devices continuously generate operational events that may initiate corresponding enterprise activities, including production confirmations, inventory adjustments, equipment maintenance notifications, and material movement records. These events must be transferred and processed without introducing conflicting, incomplete, or inaccurate business information. However, industrial devices, communication gateways, messaging infrastructure, and enterprise applications operate as independent components with different processing capabilities and availability conditions. Consequently, an event successfully generated at the equipment level does not necessarily indicate that its corresponding enterprise transaction has been completed. Maintaining consistency across these distributed components requires coordinated validation, message management, transaction identification, and reconciliation mechanisms.

Eventual consistency presents an important consideration in asynchronous integration environments because operational information may become available in different systems at different times. Unlike tightly coordinated synchronous transactions, asynchronous processing allows event producers and consumers to execute independently, potentially creating temporary differences between industrial measurements and enterprise records. For example, a production event may be successfully published by an industrial gateway while its corresponding inventory update remains pending in the enterprise application. Such temporary differences do not necessarily indicate data corruption, provided that the integration architecture maintains appropriate processing records and eventually reconciles the affected transactions. Nevertheless, prolonged processing delays, interrupted communication, and repeated message delivery can create inconsistencies if recovery mechanisms are insufficient. The proposed framework therefore considers eventual consistency alongside event traceability and controlled transaction completion.

Message delivery guarantees influence how operational events are transferred and processed across distributed integration components. At-most-once delivery attempts to prevent repeated message processing but may result in lost events when communication failures occur before successful receipt. At-least-once delivery prioritizes message availability through retransmission mechanisms, although repeated delivery can introduce duplicate processing risks. These delivery characteristics must be considered when selecting messaging configurations for industrial applications with different operational requirements. For example, an equipment status notification may tolerate occasional repeated transmission when duplicate detection is available, whereas production confirmations and inventory adjustments require stronger safeguards against unintended repeated transactions. Message acknowledgment, persistence, and controlled retry mechanisms contribute to delivery reliability. However, message delivery guarantees alone cannot establish exactly-once business transaction execution across independent systems.

Idempotent event handling provides an important mechanism for preventing duplicate enterprise transactions during repeated message delivery. An idempotent processing operation produces the same intended business outcome when an identical event is received and processed more than once. The

integration framework can use unique event identifiers, transaction references, source device identifiers, and processing records to determine whether an incoming event has already been handled. Before initiating an enterprise operation, the receiving component checks the event identifier against previously recorded processing information. If the event has already resulted in a completed transaction, the system can acknowledge the repeated message without executing the same business operation again. This approach is particularly important when asynchronous messaging infrastructure retransmits events following temporary communication failures. Idempotency therefore strengthens transaction reliability without requiring every communication component to maintain continuous synchronous coordination.

Event correlation and transaction identification establish traceability between industrial occurrences and their corresponding enterprise processing activities. Each event should carry contextual attributes that identify its source, generation time, operational category, and associated business reference wherever available. Correlation identifiers can connect related messages generated by different devices or processing components, while transaction identifiers provide references for monitoring enterprise execution. These identifiers allow integration services to distinguish between independent operational events and repeated transmissions of previously processed information. Reconciliation mechanisms can subsequently compare the original event records with enterprise transaction outcomes to identify missing updates, incomplete processing, or conflicting information. Such comparisons are particularly relevant when several industrial devices contribute information to a common production or inventory process. Maintaining a consistent relationship between event records and enterprise transactions improves auditability and supports systematic investigation of synchronization discrepancies.

Failure recovery requires coordinated message persistence, retry policies, and controlled handling of events that cannot be processed successfully. Temporary network interruptions, unavailable enterprise interfaces, authentication failures, and processing exceptions may prevent messages from reaching their intended consumers. Persistent messaging mechanisms help retain events until receiving components can resume processing, while configurable retry intervals allow temporary failures to be addressed without immediately abandoning the affected transactions. Exponential backoff can reduce repeated requests during prolonged service interruptions, preventing unnecessary communication pressure on unavailable components. Events that exceed configured retry limits can be transferred to a dead-letter queue for investigation and controlled reprocessing. Recovery procedures should preserve the original event identifier and processing history to avoid duplicate business execution when failed messages are resubmitted. These mechanisms help maintain operational continuity while providing visibility into unresolved integration failures.

Data validation and integrity verification further contribute to dependable synchronization between industrial devices and SAP S/4HANA. Incoming events should undergo structural checks to confirm that mandatory fields, device references, timestamps, measurement values, and event classifications are present and correctly represented. Business-level validation is also necessary to determine whether the received information is appropriate for the intended enterprise transaction. For instance, equipment identifiers must correspond to recognized enterprise records, while production measurements and material quantities must satisfy applicable business constraints. Data transformation should preserve measurement units, numerical precision, and relevant operational context during communication between different systems. Synchronization errors must be recorded with sufficient diagnostic information to support correction and reconciliation. Combining structural validation, business-rule verification, and transaction-status monitoring reduces the possibility of incomplete or contradictory enterprise records.

Security controls provide an additional foundation for protecting transaction integrity and maintaining reliable communication across industrial and enterprise environments. Device authentication establishes whether an industrial device or gateway is authorized to participate in the integration process, while application authentication verifies the identity of communicating enterprise services. Authorization policies restrict access to approved topics, messaging operations, application interfaces, and business transactions according to assigned responsibilities. Transport Layer Security (TLS) protects information

during transmission, while certificate-based authentication supports trusted communication between connected components. Identity management practices should address credential provisioning, certificate renewal, permission management, and access revocation. Furthermore, security monitoring and audit records help identify unauthorized communication attempts and unusual message-processing activities. Integrating these controls with message validation and transaction monitoring strengthens the confidentiality, integrity, and availability of information exchanged throughout the proposed architecture.

The proposed integration framework consequently treats transactional consistency, data integrity, and communication reliability as interconnected architectural requirements rather than independent technical functions. Its approach combines controlled message delivery, idempotent processing, event correlation, transaction identification, and structured reconciliation to maintain dependable synchronization across distributed industrial and enterprise environments. Persistent messaging, acknowledgment handling, retry policies, and dead-letter processing provide mechanisms for managing communication interruptions and incomplete transactions. Data validation and identity-based security controls further protect the accuracy and authorized processing of operational information. Although asynchronous communication introduces temporary consistency differences, these differences can be managed through traceable processing records and systematic recovery procedures. This integrated approach establishes a foundation for reliable industrial data synchronization while supporting the operational requirements of connected manufacturing and enterprise transaction management.

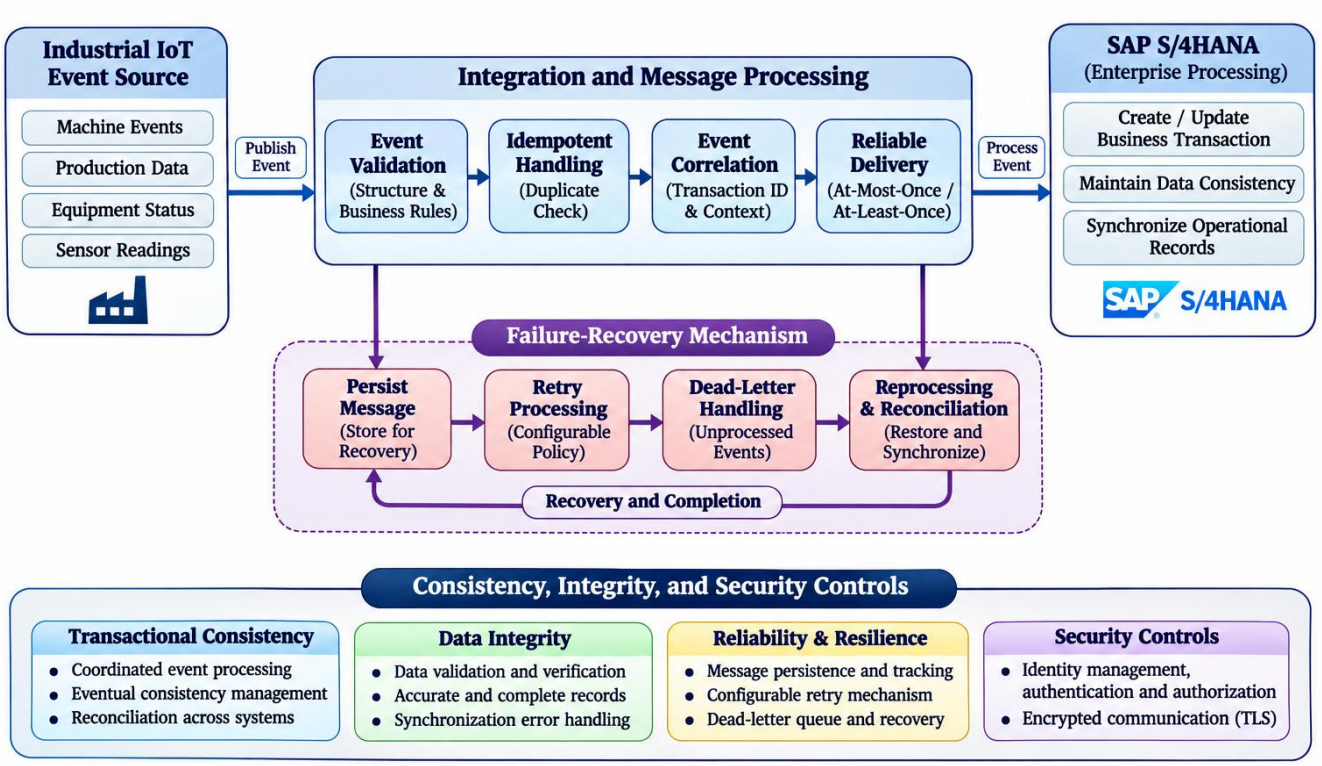

Figure 5: Transactional Consistency and Failure-Recovery Mechanism

6. Scalability, Performance Optimization, and Experimental Evaluation

Performance evaluation provides an important basis for determining whether the proposed Industrial IoT and SAP S/4HANA integration architecture can support continuous operational data exchange under varying workload conditions. The evaluation focuses on communication latency, event propagation time, message throughput, processing capacity, delivery reliability, and transactional consistency. These characteristics collectively determine the effectiveness of an integration environment when multiple

industrial devices generate operational events simultaneously. The experimental approach considers the interaction between industrial gateways, cloud integration services, messaging infrastructure, and enterprise application interfaces. Particular attention is given to identifying performance differences between conventional synchronous communication and asynchronous event-driven processing. The evaluation methodology is intended to establish measurable criteria for examining the proposed architecture rather than relying exclusively on theoretical architectural advantages.

The experimental environment is organized around interconnected industrial devices, gateway services, cloud integration components, messaging infrastructure, and SAP S/4HANA interfaces. Industrial event sources represent sensors, programmable logic controllers, and equipment monitoring applications that generate operational information at configurable intervals. Gateway components perform data acquisition, normalization, and preliminary processing before transferring events to the integration environment. SAP Business Technology Platform (SAP BTP) provides the cloud integration capabilities, while SAP Event Mesh supports the corresponding asynchronous messaging and event distribution functions considered in this study. Enterprise interfaces facilitate the processing of validated events within SAP S/4HANA. The experimental arrangement should maintain consistent communication settings, message structures, processing configurations, and network conditions to support meaningful comparisons across evaluation scenarios.

Workload characteristics determine the intensity and distribution of information processed by the integration architecture. The experimental assessment considers variations in connected device numbers, event generation frequency, message size, and simultaneous message consumption. Different workload conditions can represent individual equipment monitoring, production-line information exchange, and multiple manufacturing units communicating with centralized enterprise applications. Increasing event volumes provides an opportunity to examine whether the messaging infrastructure and receiving applications can maintain stable processing behavior. Message payload characteristics are also relevant because larger data structures may increase serialization time, transmission overhead, and application processing requirements. Establishing controlled workload categories allows the evaluation to distinguish between communication delays caused by network conditions and processing delays associated with integration components.

Synchronization latency represents the time required for an industrial event to travel from its originating device or gateway to the corresponding enterprise processing outcome. This measurement should distinguish event acquisition time, gateway processing time, message publication time, broker transportation time, and enterprise transaction execution time. End-to-end latency can be calculated using synchronized timestamps recorded at the beginning and completion of the processing sequence. Average latency provides an indication of typical system responsiveness, while percentile measurements can reveal delays experienced under heavier workloads. Event propagation time should also be examined independently from enterprise transaction completion because successful message transportation does not necessarily indicate that the corresponding business operation has finished. Such distinctions provide a more accurate assessment of synchronization behavior across distributed industrial and enterprise environments.

Message throughput and processing capacity provide additional indicators of integration performance. Throughput represents the number of successfully processed messages within a defined measurement interval, whereas processing capacity reflects the workload that individual integration components can accommodate under specified conditions. Increasing event generation rates allows the evaluation to identify the point at which incoming messages begin to exceed available processing resources. Message queue depth, consumer processing duration, gateway utilization, and enterprise interface response time can provide further information about system behavior. A growing message backlog may indicate insufficient consumer capacity, network congestion, or delays within enterprise transaction execution. Evaluating these characteristics together helps distinguish temporary workload fluctuations from sustained processing limitations that could affect industrial synchronization.

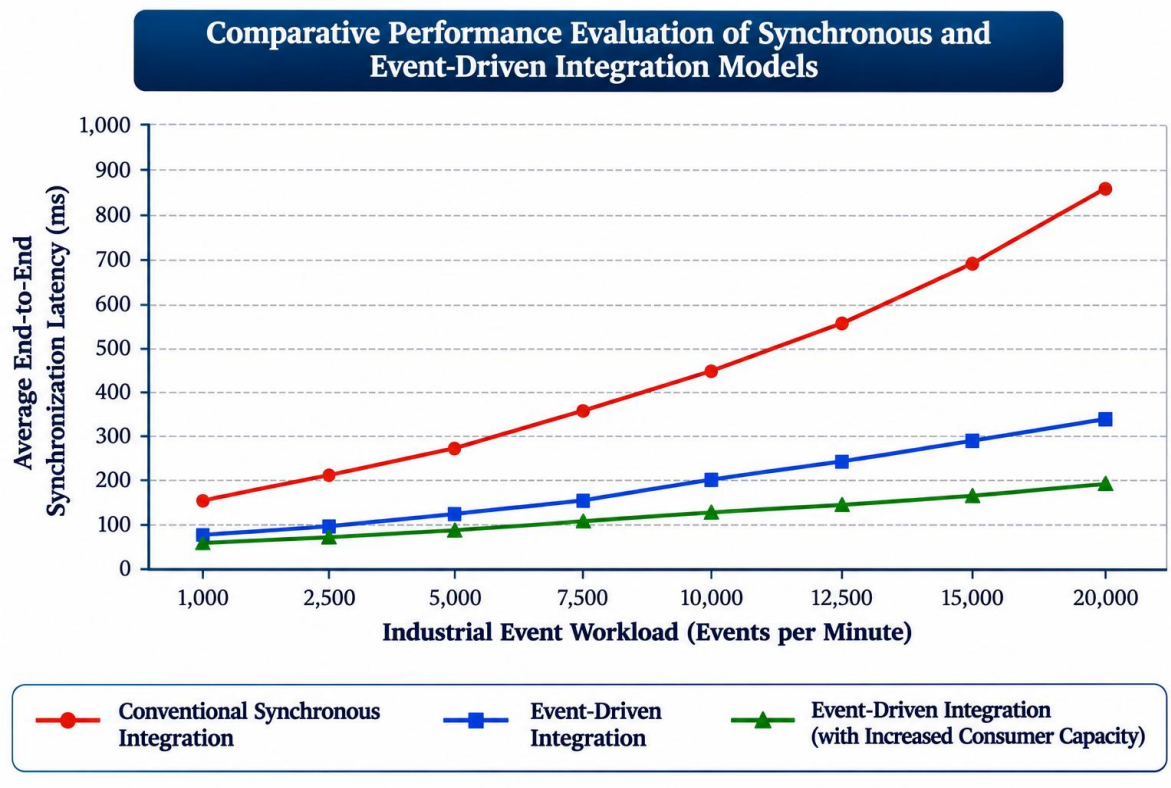

Figure 6: Comparative Performance Evaluation of Synchronous and Event-Driven Integration Models

Message delivery reliability and transaction consistency must be evaluated alongside communication performance because high throughput alone does not establish dependable integration. The experimental assessment should examine successful message delivery, unsuccessful transmission attempts, repeated message occurrences, and events requiring recovery. Duplicate detection and idempotent processing should be examined to determine whether repeated event delivery results in unintended enterprise transactions. Transaction reconciliation provides an additional measure by comparing generated industrial events with their corresponding enterprise processing records. Delivery acknowledgment should be recorded separately from business transaction confirmation to distinguish message receipt from completed enterprise execution. These measurements provide a basis for assessing whether the proposed architecture can maintain reliable operational information exchange when communication interruptions and processing failures occur.

Comparative evaluation of synchronous and event-driven integration models provides further insight into the architectural advantages and limitations of asynchronous communication. Synchronous integration depends on direct request-response exchanges, which may increase waiting time when receiving applications require additional processing or become temporarily unavailable. Event-driven integration separates event publication from consumer execution, allowing message producers and receiving applications to operate independently. The comparison should examine end-to-end latency, throughput, processing delay, message loss, duplicate handling, and recovery behavior under equivalent workload conditions. However, the results should account for differences in messaging infrastructure, network configuration, and application processing requirements. Such controls are necessary to avoid attributing performance differences exclusively to the communication model when other experimental variables may influence the observed outcomes.

Scalability analysis examines how the integration architecture responds when the number of connected devices, incoming events, and message consumers increases. Additional industrial devices may generate

higher event volumes, while multiple enterprise applications may introduce competing message consumption requirements. The assessment should therefore consider both increasing event production and expanding consumer workloads. Independent message consumption and modular integration components provide opportunities to distribute processing responsibilities across available resources. Nevertheless, messaging infrastructure, gateway capacity, network bandwidth, and SAP S/4HANA interface limitations can create bottlenecks as workloads increase. Performance optimization should consequently consider appropriate message batching, payload reduction, gateway-level filtering, consumer concurrency, and efficient topic configuration. These measures must be evaluated carefully because excessive batching or filtering may introduce additional synchronization delays or remove operational information required by enterprise processes.

The experimental evaluation ultimately provides a structured basis for identifying the operational strengths and limitations of the proposed integration framework. Performance observations should be interpreted across latency, throughput, delivery reliability, transaction consistency, and scalability rather than treating any individual measurement as a sufficient indicator of architectural effectiveness. Potential bottlenecks may emerge from network congestion, message serialization, broker processing, insufficient consumer capacity, or delayed enterprise transaction execution. Optimization opportunities therefore depend on identifying the component responsible for the observed performance limitation and selecting an appropriate corrective mechanism. The evaluation framework contributes a measurable approach for examining how asynchronous event communication supports industrial data synchronization while maintaining enterprise transaction reliability. Its findings can inform integration architecture decisions for connected manufacturing environments and provide a foundation for subsequent implementation-oriented investigations.

Table 2: Experimental Performance Assessment of the Unified IoT–SAP Integration Framework

Performance parameter	**Evaluation metric**	**Assessment objective**	**Anticipated outcome**
Synchronization latency	Average end-to-end latency (ms)	Measure event-to-transaction response time	Reduced latency under event-driven processing
Event throughput	Messages processed per second	Evaluate message-handling capacity	Increased message-processing capacity
Message delivery reliability	Successful message delivery (%)	Assess communication dependability	Improved delivery reliability
Duplicate event rate	Duplicate messages (%)	Examine idempotent event handling	Reduced duplicate transactions

Transaction consistency	Successfully reconciled transactions (%)	Verify enterprise data accuracy	Improved synchronization consistency
Recovery performance	Failure recovery time (seconds)	Evaluate recovery from interruptions	Reduced recovery time
Message queue depth	Pending message count	Identify processing congestion	Controlled message accumulation
Scalability	Performance under increasing workloads	Assess capacity as device connectivity increases	Stable processing under higher workloads

7. Results, Discussion, and Industrial Implications

The evaluation of the proposed unified Industrial IoT and SAP S/4HANA integration framework provides a basis for examining the relationship between asynchronous event communication, synchronization performance, and enterprise transaction reliability. The assessment considers event propagation latency, message throughput, processing capacity, delivery reliability, transactional consistency, and scalability under varying industrial workloads. Particular attention is given to the differences between conventional synchronous integration and broker-mediated event-driven communication. The architectural evaluation indicates that separating event acquisition from enterprise transaction execution can reduce direct communication dependencies and provide greater flexibility in managing distributed processing activities. However, the extent of performance improvement depends on gateway configuration, messaging capacity, network conditions, and the responsiveness of receiving enterprise interfaces.

Synchronization latency represents an important factor influencing the responsiveness of enterprise applications to industrial events. Conventional synchronous communication introduces dependencies between message producers and receiving applications, potentially increasing waiting time when enterprise interfaces experience processing delays. Event-driven communication separates event publication from subsequent consumer execution, allowing operational information to enter the messaging infrastructure without requiring immediate completion of the corresponding enterprise transaction. Distributing message-processing responsibilities across independent consumers can help manage increasing event volumes and reduce direct processing dependencies. Nevertheless, end-to-end synchronization remains dependent on the time required for message transportation, validation, and enterprise transaction completion.

Asynchronous messaging also influences communication overhead and integration efficiency by reducing the number of direct dependencies between industrial devices and enterprise applications. In conventional point-to-point arrangements, additional connected systems may require separate interfaces and communication relationships, increasing maintenance complexity. Broker-mediated communication provides an alternative in which industrial events are published through configured messaging channels and distributed to authorized consumers. This arrangement allows multiple applications to receive relevant information without requiring individual direct connections to every event producer. Topic-based routing

and subscription mechanisms further support selective message distribution, reducing unnecessary processing by unrelated applications. However, messaging infrastructure introduces its own operational requirements, including broker configuration, queue monitoring, message persistence, and consumer capacity management. The resulting efficiency therefore depends on the coordination of communication and processing resources across the integration environment.

Transactional consistency and data integrity remain essential considerations when evaluating the effectiveness of distributed event processing. Asynchronous communication can create temporary differences between industrial operational records and corresponding SAP S/4HANA transactions because messages may be received and processed at different times. Such differences must be managed through event identification, idempotent processing, transaction correlation, and reconciliation mechanisms. The proposed framework addresses these requirements by associating incoming events with identifiable processing records and verifying transaction outcomes before considering synchronization complete. Duplicate detection helps prevent repeated messages from producing unintended business transactions, while message acknowledgment provides visibility into communication status. These mechanisms support reliable enterprise information exchange, although message delivery confirmation must remain distinct from confirmation of successful business transaction execution.

Scalability and fault tolerance influence the ability of the integration architecture to operate across manufacturing environments with varying device connectivity and event-generation frequencies. Independent event producers and consumers allow processing responsibilities to be distributed without establishing additional direct communication dependencies for every connected device. Increasing consumer capacity can help accommodate higher message volumes when sufficient messaging and enterprise processing resources are available. Persistent message storage, retry policies, acknowledgment handling, and dead-letter processing provide mechanisms for managing temporary interruptions and unsuccessful event delivery. Nevertheless, these capabilities do not eliminate all operational constraints. Network congestion, gateway limitations, broker throughput, and SAP S/4HANA interface capacity can still restrict overall performance. Effective scalability therefore requires coordinated capacity planning and continuous monitoring across the complete integration architecture.

The industrial implications of the proposed framework extend across manufacturing operations, equipment monitoring, production coordination, and enterprise resource planning. Timely synchronization of equipment conditions and production information can improve the availability of operational data for maintenance planning, inventory management, and production reporting. For example, equipment condition notifications can provide relevant information to maintenance processes, while production events can support updates to corresponding enterprise records. Improved information availability may also reduce delays associated with manual reporting and periodic data transfers. However, the business value of integration depends on the relevance, accuracy, and timeliness of the information being exchanged. Industrial organizations must therefore identify which device events require immediate enterprise processing and which measurements can be aggregated or retained for analytical purposes. This distinction helps balance communication requirements with enterprise processing capacity.

Practical implementation requires SAP architects and industrial integration specialists to coordinate application connectivity, messaging configuration, device communication, and enterprise security controls. Industrial gateways should be configured to manage protocol conversion, data normalization, and preliminary validation before events enter the cloud integration environment. SAP Business Technology Platform (SAP BTP) provides cloud integration capabilities, while SAP Event Mesh supports asynchronous messaging and distributed event exchange within the historical technology context considered in this research. Standardized application interfaces, including supported OData and REST-based services, can facilitate communication with SAP S/4HANA. Implementation planning should additionally address device identity management, authentication, authorization, encryption, topic permissions, and message retention policies. Operational monitoring should cover queue depth, failed

deliveries, consumer response times, transaction completion, and reconciliation exceptions to maintain visibility across distributed components.

Several limitations should be considered when interpreting the evaluation and its architectural implications. The proposed framework establishes a structured integration model, but its performance characteristics require validation through controlled implementation and repeatable industrial experiments. Results may vary according to device populations, event payload sizes, network topology, messaging configuration, enterprise interface capacity, and manufacturing workload characteristics. The comparative performance representation in Figure 6 is illustrative and should not be interpreted as a record of experimentally measured results unless supported by actual measurements from an experimental environment. Furthermore, differences between industrial communication protocols and enterprise data structures may require additional transformation and compatibility testing. Future investigations could examine larger device populations, variable network conditions, broker failure scenarios, transaction reconciliation accuracy, and security overhead under sustained workloads. Such investigations would provide stronger evidence concerning deployment suitability across different industrial environments.

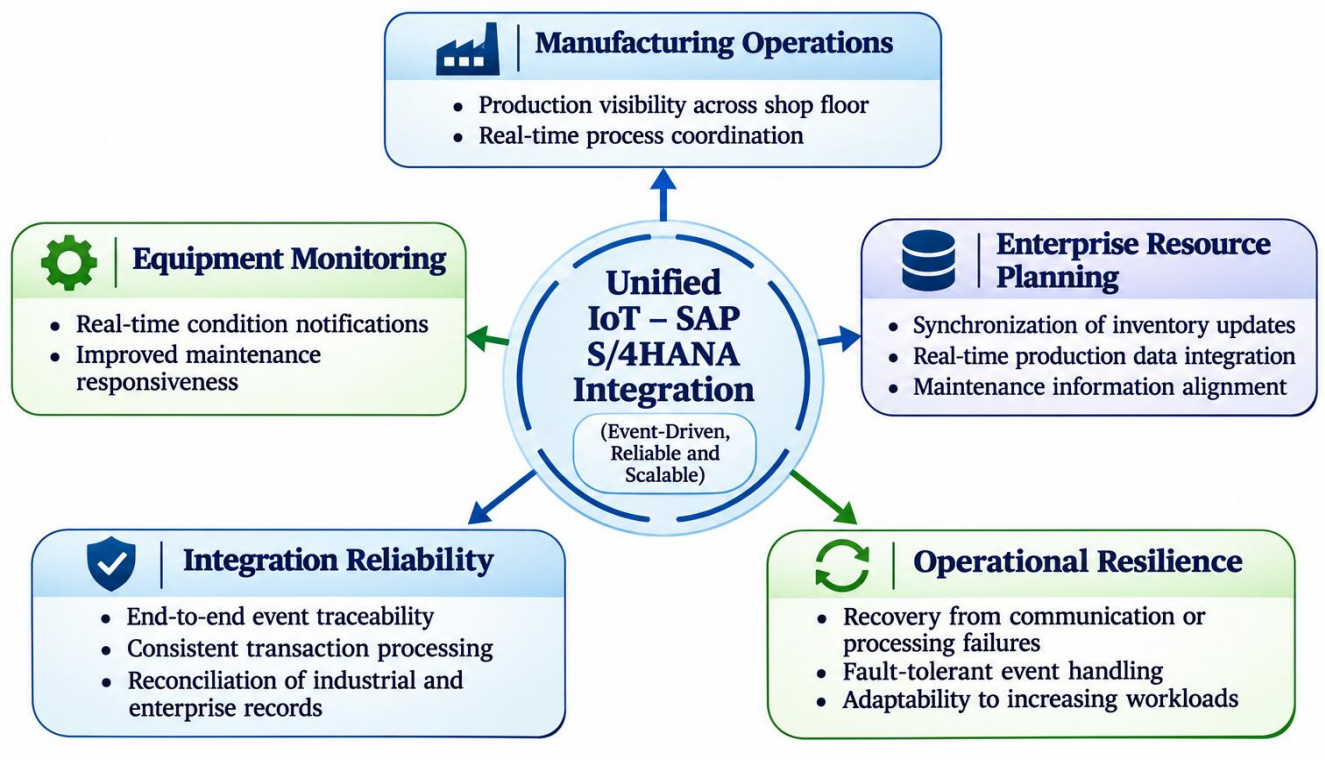

Figure 7: Industrial Implications and Operational Benefits of Unified IoT–SAP S/4HANA Integration

The overall discussion establishes that reliable Industrial IoT integration with SAP S/4HANA requires coordinated attention to communication responsiveness, transactional consistency, processing scalability, and operational resilience. Event-driven communication offers architectural advantages by separating event generation from enterprise transaction execution and supporting independent message consumption. These advantages become more meaningful when combined with structured event validation, topic-based routing, duplicate prevention, persistent delivery, and systematic recovery mechanisms. The proposed framework contributes an integrated architectural perspective for connecting heterogeneous industrial environments with centralized enterprise applications while maintaining traceability across distributed processing stages. Its industrial significance lies in providing design considerations for more dependable

operational information exchange, while its academic contribution lies in connecting event propagation performance with transaction integrity and integration reliability. Further implementation-based evaluation remains necessary to establish its performance under real manufacturing conditions.

8. Conclusion and Future Research Directions

The integration of Industrial Internet of Things (IIoT) environments with SAP S/4HANA represents an important architectural requirement for manufacturing organizations seeking continuous operational information exchange and coordinated enterprise transaction processing. This study examined the challenges associated with connecting heterogeneous industrial devices, communication gateways, cloud integration services, and enterprise applications through a unified event-driven architecture. The proposed framework combines industrial data acquisition, gateway-level processing, asynchronous message distribution, standardized enterprise interfaces, and transaction validation within a coordinated integration environment. Particular attention was given to communication latency, data consistency, message delivery reliability, and architectural scalability. The study establishes that dependable industrial integration requires more than the transportation of operational measurements, demanding coordinated management of event processing and corresponding enterprise transactions.

The research objectives were addressed through the examination of industrial communication requirements, integration architecture design, event propagation mechanisms, transactional consistency, and performance evaluation criteria. The proposed architecture separates operational data acquisition, event processing, messaging, and enterprise transaction execution into distinct functional components. This separation provides a structured approach for managing communication between industrial equipment and SAP S/4HANA without requiring continuous direct dependencies between every connected device and enterprise application. Standardized message structures, gateway-level processing, topic-based distribution, and controlled application interfaces contribute to interoperability across heterogeneous environments. The architectural evaluation also identifies the importance of monitoring message delivery, processing completion, and synchronization errors when assessing integration reliability.

Event-driven communication provides a significant architectural contribution by separating event publication from consumer processing. Unlike conventional synchronous integration, asynchronous messaging allows industrial gateways to publish operational information without waiting for receiving enterprise applications to complete their transactions. Topic-based routing and independent message consumption further support distributed processing and reduce direct application dependencies. These characteristics provide opportunities to manage increasing industrial event volumes while maintaining flexibility across connected systems. However, low-latency synchronization depends on the combined performance of gateways, communication networks, messaging infrastructure, and enterprise interfaces. The study therefore considers event-driven architecture an important mechanism for improving integration responsiveness rather than an independent guarantee of immediate transaction completion.

Transactional consistency and reliable distributed message processing remain essential to the successful implementation of the proposed framework. Asynchronous communication introduces the possibility of temporary differences between industrial events and their corresponding enterprise records, requiring systematic mechanisms for event identification, duplicate prevention, and transaction reconciliation. Idempotent processing, message acknowledgment, persistent delivery, configurable retry policies, and dead-letter handling contribute to controlled event processing and recovery. Data validation and transaction correlation further strengthen the relationship between original operational occurrences and resulting enterprise updates. These mechanisms collectively support data integrity while reducing the possibility of unintended duplicate transactions. Their effectiveness depends on appropriate implementation and continuous monitoring across the integration environment.

The strategic significance of SAP Business Technology Platform (SAP BTP) and SAP Event Mesh lies in their capabilities for coordinating enterprise connectivity, application communication, and distributed event exchange. SAP BTP provides integration services and application connectivity capabilities, while SAP Event Mesh supports asynchronous messaging and topic-based event distribution. These capabilities provide architectural foundations for connecting industrial information sources with enterprise applications through reusable integration services and message-oriented communication mechanisms. Their application within SAP-centered environments supports modular integration, standardized connectivity, and separation of communication responsibilities. From an enterprise architecture perspective, combining cloud integration with message-oriented middleware provides a structured approach to coordinating industrial operations and business transactions across distributed systems.

The industrial implications extend to connected manufacturing, equipment monitoring, production coordination, inventory management, and enterprise resource planning. Timely exchange of operational information can improve the availability of equipment conditions and production-related data for enterprise decision-making. Maintenance notifications can support more responsive equipment management, while production events can contribute to accurate manufacturing records and inventory updates. Improved synchronization may also reduce dependence on manual information transfers and delayed reporting processes. Nevertheless, successful industrial digitalization requires organizations to consider operational requirements, existing equipment capabilities, communication infrastructure, and enterprise application constraints. The proposed framework offers an architectural foundation for these activities while emphasizing the importance of reliable information exchange and controlled transaction execution.

Several research limitations remain relevant when considering the practical applicability of the proposed architecture. Heterogeneous industrial devices may introduce differences in protocol support, data representation, event-generation frequency, and communication reliability. Variable network conditions can affect message transportation, while increasing workloads may create processing bottlenecks within gateways, messaging services, and enterprise interfaces. Furthermore, the performance evaluation framework identifies measurable assessment criteria, but its illustrative performance representations should not be interpreted as verified experimental findings without implementation-based measurements. Actual deployment outcomes will depend on system configuration, device populations, message characteristics, and available processing resources. These limitations indicate the need for controlled industrial experimentation and broader validation across different manufacturing environments.

Future research should investigate adaptive event-routing mechanisms capable of adjusting message distribution according to processing capacity, event priority, and changing network conditions. Predictive integration optimization could examine workload patterns and communication behavior to identify potential congestion before it affects enterprise synchronization. Further investigation into industrial interoperability could explore standardized information models, protocol translation, and consistent event representation across different equipment manufacturers. Additional research may also examine distributed transaction reconciliation, automated failure recovery, security monitoring, and synchronization behavior under sustained industrial workloads. Implementation-based studies involving larger device populations and varied manufacturing configurations would provide further evidence concerning the scalability and operational suitability of the proposed architecture.

The study concludes that unified Industrial IoT and SAP S/4HANA integration requires a balanced combination of event-driven communication, reliable message processing, transactional consistency, and scalable enterprise connectivity. The proposed framework provides a structured architectural approach for coordinating industrial data acquisition and enterprise transaction execution while addressing the challenges associated with heterogeneous devices and distributed processing. Its contribution lies in connecting low-latency event propagation with data integrity, failure recovery, and enterprise integration reliability within a unified design perspective. Continued investigation and practical validation can extend this foundation toward more adaptive, interoperable, and resilient industrial information systems. Such

developments can support connected manufacturing environments in which operational information is exchanged efficiently and enterprise records remain dependable.

9. References

[1] Profanter, S., Tekat, A., et al. (2019). OPC UA versus ROS, DDS, and MQTT: Performance evaluation of Industry 4.0 protocols. 2019 IEEE International Conference on Industrial Technology (ICIT), 955–962. https://doi.org/10.1109/ICIT.2019.8755050

[2] Rocha, M. S., Sestito, G. S., et al. (2019). On the performance of OPC UA and MQTT for data exchange between industrial plants and cloud servers. Acta IMEKO, 8(2). https://doi.org/10.21014/acta_imeko.v8i2.648

[3] Cavalieri, S. (2021). A proposal to improve interoperability in the Industry 4.0 based on the Open Platform Communications Unified Architecture standard. Computers, 10(6), Article 70. https://doi.org/10.3390/computers10060070

[4] Chen, Y. (2020). A survey on industrial information integration 2016–2019. Journal of Industrial Integration and Management. https://doi.org/10.1142/S2424862219500167

[5] Younan, M., Houssein, E. H., et al. (2020). Challenges and recommended technologies for the industrial internet of things: A comprehensive review. Measurement, 151, Article 107198. https://doi.org/10.1016/j.measurement.2019.107198

[6] Khanam, S., Ahmedy, I. B., et al. (2020). A survey of security challenges, attacks taxonomy and advanced countermeasures in the Internet of Things. IEEE Access, 8, 219709–219743. https://doi.org/10.1109/ACCESS.2020.3037359

[7] Ioana, A., & Korodi, A. (2021). DDS and OPC UA protocol coexistence solution in real-time and Industry 4.0 context using non-ideal infrastructure. Sensors, 21(22), Article 7760. https://doi.org/10.3390/s21227760

[8] Okwuibe, J., Haavisto, J., et al. (2021). SDN-enabled resource orchestration for industrial IoT in collaborative edge-cloud networks. IEEE Access, 9, 115839–115854. https://doi.org/10.1109/ACCESS.2021.3105944

[9] Caiza, G., Saeteros, M., et al. (2020). Fog computing at industrial level, architecture, latency, energy, and security: A review. Heliyon, 6(4), Article e03706. https://doi.org/10.1016/j.heliyon.2020.e03706

[10] Fuller, A., Fan, Z., et al. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

[11] Trunzer, E., Schilling, T., et al. (2020). Comparison of communication technologies for industrial middlewares and DDS-based realization. IFAC-PapersOnLine, 53(2), 10935–10942. https://doi.org/10.1016/j.ifacol.2020.12.2836

[12] Jiang, Z., Chang, Y., et al. (2020). Design of software-defined gateway for industrial interconnection. Journal of Industrial Information Integration. https://doi.org/10.1016/j.jii.2020.100130

[13] Dahlmanns, M., Lohmöller, J., et al. (2020). Easing the conscience with OPC UA: An Internet-wide study on insecure deployments. Proceedings of the ACM Internet Measurement Conference, 101–110. https://doi.org/10.1145/3419394.3423666

[14] Qiu, T., Chi, J., et al. (2020). Edge computing in industrial Internet of Things: Architecture, advances and challenges. IEEE Communications Surveys & Tutorials, 22(4), 2462–2488. https://doi.org/10.1109/COMST.2020.3009103

[15] Lan, L., Shi, R., et al. (2019). A universal complex event processing mechanism based on edge computing for Internet of Things real-time monitoring. IEEE Access, 7, 101865–101878. https://doi.org/10.1109/ACCESS.2019.2930313

[16] He, J., Kong, L., et al. (2020). Decision triggered data transmission and collection in industrial Internet of Things. 2020 IEEE Wireless Communications and Networking Conference (WCNC), 1–5. https://doi.org/10.1109/WCNC45663.2020.9120749

[17] Ungurean, I., & Gaitan, N. C. (2020). A software architecture for the industrial Internet of Things: A conceptual model. Sensors, 20(19), Article 5603. https://doi.org/10.3390/s20195603

[18] Dai, W., Wang, P., et al. (2019). Semantic integration of plug-and-play software components for industrial edges based on microservices. IEEE Access, 7, 125882–125892. https://doi.org/10.1109/ACCESS.2019.2938565

[19] Liu, J., Chen, M., et al. (2020). A new model of industrial Internet of Things with security mechanism: An application in complex workshop of diesel engine. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 234(2), 564–574. https://doi.org/10.1177/0954406219884970

[20] Leppänen, T., & Riekki, J. (2019). Energy efficient opportunistic edge computing for the Internet of Things. Web Intelligence, 17(3). https://doi.org/10.3233/WEB-190414

[21] Borsatti, D., Cerroni, W., et al. (2020). From IoT to cloud: Applications and performance of the MQTT protocol. 22nd International Conference on Transparent Optical Networks (ICTON). https://doi.org/10.1109/ICTON51198.2020.9203167

[22] Mazayev, A., & Correia, N. S. C. (2019). A distributed CoRE-based resource synchronization mechanism. IEEE Internet of Things Journal, 7(5), 4625–4640. https://doi.org/10.1109/JIOT.2019.2958375

[23] Guimarães, C. S. S., de Andrade, M., et al. (2020). IoT architecture for interoperability and monitoring of industrial nodes. Procedia Manufacturing, 52, 313–318. https://doi.org/10.1016/j.promfg.2020.11.052

[24] Mourad, M. H., Nassehi, A., et al. (2020). Assessment of interoperability in cloud manufacturing. Robotics and Computer-Integrated Manufacturing, 61, Article 101832. https://doi.org/10.1016/j.rcim.2019.101832

[25] Helu, M. M., Sprock, T. A., et al. (2020). Scalable data pipeline architecture to support the Industrial Internet of Things. CIRP Annals, 69(1), 385–388. https://doi.org/10.1016/j.cirp.2020.04.006

[26] Lu, Y., Witherell, P., et al. (2020). Standard connections for IIoT empowered smart manufacturing. Manufacturing Letters, 26. https://doi.org/10.1016/j.mfglet.2020.08.006

[27] Ehie, I. C., & Chilton, M. A. (2020). Understanding the influence of IT/OT convergence on the adoption of Internet of Things (IoT) in manufacturing organizations: An empirical investigation. Computers in Industry, 115, Article 103166. https://doi.org/10.1016/j.compind.2019.103166

[28] Farooqui, A., Bengtsson, K., et al. (2020). Towards data-driven approaches in manufacturing: An architecture to collect sequences of operations. International Journal of Production Research, 58(16), 4947–4963. https://doi.org/10.1080/00207543.2020.1735660

[29] Choudhary, K., Gaba, G. S., et al. (2020). MAKE-IT: A lightweight mutual authentication and key exchange protocol for Industrial Internet of Things. Sensors, 20(18), Article 5166. https://doi.org/10.3390/s20185166

[30] Coito, T., Martins, M. S. E., et al. (2020). A middleware platform for intelligent automation: An industrial prototype implementation. Computers in Industry, 123, Article 103329. https://doi.org/10.1016/j.compind.2020.103329