

(REVIEW ARTICLE)



Integration frameworks for harmonizing CRM and ERP in real-time revenue operations

Ansha Bundela *

University of Cincinnati, Cincinnati, Ohio, USA.

World Journal of Advanced Engineering Technology and Sciences, 2026, 20(01), 037-048

Publication history: Received on 23 April 2026; revised on 30 June 2026; accepted on 02 July 2026

Article DOI: <https://doi.org/10.30574/wjaets.2026.20.1.0296>

Abstract

The combination of customer relationship management (CRM) and enterprise resource planning (ERP) has shifted from a back-office interoperability issue to a strategic requirement for real-time revenue operations. Revenue execution now depends upon the continuous integration of lead qualification, quoting, order capture, contract administration, invoicing, collections, renewals and service-led expansion. Revenue leakage, forecast distortion, and poor operational accountability are caused by fragmented data models, inconsistent transaction ownership, and slow synchronization within this environment. This review discusses integration frameworks that integrate CRM and ERP in near real time with particular focus on architectural patterns, semantic alignment, process orchestration, governance controls and performance implications. The literature suggests that the quality of integration is not dependent on the raw connectivity but rather designed event flows, canonical business objects, exception management and stewardship structures. The review also finds that the scholarly base remains fragmented as there is little direct peer-reviewed research concerning the revenue operations with an evidence base scattered within the enterprise systems, process management, data governance and digital platform research. The field requires stronger conceptual consolidation, comparative empirical studies, and better measurement of revenue-operational outcomes across integration designs.

Keywords: CRM-ERP Integration; Event-Driven Architecture; Real-Time Revenue Operations; Revenue Data Harmonization; Sales-to-Cash Orchestration

1. Introduction

The integration of CRM and ERP has been given a new strategic weight as revenue execution has become a cross-system, cross-functional and increasingly time-sensitive operating challenge. The traditional organizational barriers of customer-facing operations and back-office transaction execution made batch interfaces and periodic reconciliation tolerable in most environments. This assumption is no longer valid. The existing revenue models are based on rapid quotations, dynamic pricing, recurring billing, subscription amendments, service entitlements, and usage-based invoicing, and renewal prediction. All these activities initiate state changes that are created in one system but can gain economic significance only when they are coordinated across multiple systems. Opportunity, account, pipeline, activity, and commercial-intent records are normally found in CRM environments and inventory, order status, fulfillment, invoicing, financial posting and revenue recognition records are found in ERP environments. Real-time revenue operations therefore require an integrated form of coordination structure that can go beyond mere data exchange and support synchronized commercial activity in the case of low-latency circumstances.

The issue is significant because revenue operations are increasingly defined by temporal coherence, not only functional completeness. Revenue exposure may be modified instantly in reaction to a business event such as product setup, acceptance of an offer, adjustment of a contract, service upgrade or contract renewal. When CRM and ERP are loosely

* Corresponding author: Ansha Bundela.

coupled, downstream effects are often experienced as a strain between quote and order, invoice errors, provisioning latency, credit disputes, or unstable forecasts. Delay in transmission is not the only problem. The deeper issue is connected with discordant business semantics, competing data ownership, asymmetric rate of updates and bad visibility of where the state of the revenue is at any given moment. A quote may be commercially approved in CRM while the corresponding pricing logic in ERP remains outdated or inconsistent. Sales commitments may be reflected in the pipeline before contractual, tax, or fulfilment requirements are validated. A service-led signal may indicate an upsell opportunity but may remain trapped in case-management activity and never reaches orchestration of the order. In such a situation, the latency issue is as much organizational as technical because any asynchronous latency is a revenue loss and a lack of clarity in accountability.

The existing bodies of knowledge approach this issue in a disjointed way. The information systems scholarship adopts the issues of enterprise integration, API management, microservices, iPaaS, business process orchestration, process mining, data quality and master data management. Management and operations scholarship examines commercial coordination, forecasting, and performance visibility. The digital transformation research discusses the platformization, cross-functional redesign, and modular architecture. The phrase itself, when used directly by peer-reviewed authors, is rare, and direct scholarly discussion of CRM in harmony with ERP to perform real-time revenue execution is thus more sparse than discussion by practitioners might indicate. Relevant knowledge is spread over adjacent literatures that may not necessarily have concepts, outcome measures and unit of analysis. The message patterns and interoperability can be in the focus of a system-design paper. A process-management paper may emphasize exception traces and flow conformance. A data-governance paper may focus on stewardship, identifiers, and quality control. The quality of the forecasts, speed of conversion and reduction of leakages can be highlighted in a commercial-operations paper. Lack of a common integrative framework has slowed down the build-up of similar evidence.

A number of unanswered questions warrant a special study. Firstly, current literature still regards CRM-ERP integration as a connectivity challenge despite real-time revenue processes equally depending on the canonical business objects, transaction authority scopes, time-based sequencing, and observability. Second, the reported benefits are not necessarily talked about in terms of actual revenue, but in a generic digital-transformation terms, i.e., accuracy of quotes, integrity of bookings, fallout rate of orders, invoice dispute frequency, capture of renewals, or latency to reconcile. Third, architecture is still a contentious matter. APIs, middleware hubs, iPaaS orchestration, master-data hubs, and event-based patterns may all support agility, but they distribute control and complexity differently. Fourth, the practices of methodology are not even. Case studies predominate architectural discourse and there is little rigorous comparative evidence on the revenue results of alternative patterns of integration. Fifth, new business models like subscription, consumption-based pricing, hybrid service bundles and partner-led ecosystems add further strain on the traditional CRM-ERP interfaces as price, entitlement, contract, billing and service cues no longer flow in a linear fashion.

This review aims at summarizing and critically reviewing the academic literature that best applies to the integration models to synchronize CRM and ERP in real-time revenue processes. The review uses the concept of real-time revenue operations as a coordinated execution of lead-to-cash, quote-to-cash, contract-to-bill, renewal and service-led expansion processes in the context that they need to be semantically consistent and process visible in near real-time on both the customer-facing and transactional systems. The subsequent sections cover the literature in thematic clusters, build a topic-specific conceptual framework, summarize methodological strategies, report patterns and contradictions, and priorities in research that could result in the analytical sophistication of the field.

2. Literature Review

The recent literature relevant to CRM-ERP harmonization of real-time revenue operations, converges around five thematic areas: architectural integration patterns, semantic data harmonization, process orchestration and observability, governance and control, and performance implications. The architectural stream is an account of a transition from tightly coupled point-to-point interfaces to mediated and modular form of integration. Middleware based designs reduced proliferation of direct dependencies, but were more likely to have transformation bottlenecks in the middle. iPaaS and API-led designs offered quicker implementation and enhanced reuse of services, in particular in heterogeneous SaaS environments. The request-response exchange also became the focus of event driven models to propagate the states with publish-subscribe patterns. In the case of revenue operations, this architectural transformation is important since revenue-relevant events are very unlikely to be in a clean serial chain. The occurrence of opportunity stage movement, discount approval, order creation, shipment release, service entitlement activation, billing adjustment, and renewal risk detection usually takes place in overlapping cycles. This body of literature demonstrates that a technically modern integration stack is necessary but not sufficient. It is only by the architecture also addressing semantic consistency, replayability, exception routing and authority of competing updates that real time operating value becomes a possibility.

The second major stream concerns data harmonization. CRM and ERP will tend to differ in the structure and purpose of basic objects such as customer, account structure, product, price, contract, order, invoice and entitlement. The breakage is not by chance. CRM models are concerned with the sales context and commercial intent, and the ERP models are concerned with transaction completeness, the internal control and the financial effect. Field mapping is thus not enough when it comes to harmonization. The literature on canonical models, master-data governance, identity resolution, and metadata discipline are recurrent themes as the foundation of long-lasting integration. This requirement is enhanced by revenue operations since commercial and financial truth must come together fast enough to enable the integrity of booking and operational responsiveness. The type of customer can be a prospect, legal entity, bill-to party, ship-to party, site, reseller, payer or service location. A bundle of CRM can be a product, a stocked product of ERP and a revenue-performance need of downstream financial logic. Real-time integration may only accelerate inconsistency without formal semantic alignment.

A third stream looks at orchestration and observability of processes. The idea of data synchronization was often a target in the past enterprise-integration models. More recent process-management and process-mining literature treats the integration layer as a carrier of executable business state. This is vital to the revenue operations because the economically relevant question is not whether a record was successfully transferred, but whether or not the commercial process was progressed without the loss of control. The literature identifies lead-to-order, quote-to-cash, order-to-fulfillment, contract-to-bill, incident-to-renewal, and service-to-expansion flows to be particularly susceptible to hidden handoffs and local workarounds. Diagnostic visibility is also introduced to process mining in terms of re-creating the sequence of events in the systems of the enterprise and identifying the deviations of conformance, work-around, or silent delays. Diagnostic visibility alone is not adequate to secure harmony in operations. Several studies indicate that the most helpful process transparency is used with the rule-conscious orchestration and the ownership of exceptions and event semantics which can distinguish between informative and revenue-related process transitions.

The governance stream makes an important modification to the purely architectural discourse. Weak data stewardship, security control, audit traceability, and change management can undermine real-time integration, and real-time integration can expose these weaknesses. Revenue operations generate particularly delicate coordination intersections due to the fact that pricing approval, discount control, order acceptance, credit release, tax validation, billing and revenue recognition each have compliance connotations. The literature links integration quality to the stewardship models, policy enforcement, access segmentation and traceable lineage across the platforms. The other general finding is that failure of governance is not typically the result of lack of rules but of diffused location of the rule. Pricing logic can be found partially in CRM, partially in ERP and partially in middleware. Local files (or service tools) may be used to correct customer hierarchies manually. The revenue status measures can be aggregated on the non-controlling analytics systems. Real time can be used to cover distributed inconsistency in such environments. Most convincing work in this group is that governance must be integrated within the framework of the integration, and not added to it once it is implemented technically.

Table 1 summarizes the key thematic findings from the reviewed literature. In spite of this drawback, there are certain trends. The response is more often than not being overstated in architectural papers and semantic friction understated. Data-governance papers focus on semantic control, yet are less specific on the low-latency orchestration. Process-based articles display spheres of implementation, but never go into the full responsibility of corrective action. This change of focus is one of the factors that have made the area still lag behind in having even footing topic-based model of aligning CRM and ERP in real-time revenue execution.

One last theme of the literature is the outcomes of performance and difficulty of evaluation. Reported benefits include fewer reconciliation cycles, reduced manual rekeying, improved order accuracy, greater forecast consistency, and more consistent customer records. The basis of comparison is, however, quite varied. Studies are available which are concerned with technical measures such as latency, throughput and recovery of failures. The others focus on the business results in terms of conversion velocity, billing accuracy, reduced disputes, and customer-response time. Few of them offer an entire chain of connection between architectural design as well as revenue implications by observing process alterations. This is a disadvantage to causal interpretation. It also explains why the centralization/modular autonomy, canonical/bounded-context models, and batch stabilization/near-real-time propagation remain unresolved scholarly issues. The literature indicates a tentative conclusion, however: the co-location of CRM and ERP to real-time revenue processes is no longer a one-integration choice but instead a joint design concern which cuts across architecture, semantics, process control and governance.

Table 1 Summary of key findings

Ref	Focus	Key Findings
[6]	API-led CRM–ERP integration for distributed commercial processes	API reuse improves system modularity and rollout speed, but revenue-state consistency deteriorates when object semantics remain locally defined and version control is weak.
[7]	Event-driven enterprise integration in SaaS-heavy operating environments	Publish-subscribe patterns reduce synchronization delay and support scalable propagation of sales, billing, and service events, yet duplicate event handling and replay governance become major control concerns.
[8]	Master data management for customer, product, and pricing harmonization	Canonical object design improves cross-platform coherence, especially for account hierarchies and product bundles, but stewardship overhead rises sharply in multi-entity and partner-led contexts.
[9]	Process mining across quote-to-cash and order-to-cash execution	Event-log reconstruction reveals hidden rework loops, manual corrections, and latency pockets that are not visible in system-level dashboards, particularly around pricing, credit, and fulfillment handoffs.
[10]	iPaaS orchestration for hybrid legacy and cloud enterprise landscapes	iPaaS lowers integration build effort and supports connector diversity, but governance often drifts into proprietary transformation layers that obscure business-rule ownership.
[11]	Microservices and domain-oriented decomposition in enterprise applications	Domain isolation supports local agility and change resilience, although cross-domain commercial transactions become harder to reconcile without strong event contracts and shared identifiers.
[12]	Data quality and lineage in customer-facing transactional ecosystems	Data quality problems cluster around duplicate customers, inconsistent contract references, stale pricing attributes, and missing lineage for manual overrides, all of which distort revenue metrics.
[13]	Revenue process control and performance observability	Near-real-time visibility improves exception detection and forecast confidence only when operational metrics are tied to transactional truth rather than dashboard approximations.
[14]	Governance of pricing, discounting, and approval logic across systems	Distributed rule location generates approval bypass risk, inconsistent margin control, and audit gaps unless rule repositories and decision traces are formally integrated.
[15]	Service-to-expansion and renewal coordination	Expansion and renewal capture improve when service incidents, product usage, entitlement status, and account health signals are integrated into commercial workflows instead of remaining in isolated service systems.

3. Methodology

The literature indicates that the difference between data movement and revenue-state harmonization should be the initial point of a topic-specialized framework. Numerous enterprise integration literature defines adapters, APIs, message buses, and middleware patterns in a level of technical detail, but the revenue operations entail a further layer of explanation. CRM field to ERP field update may not be a valid state transition of commerce. The architecture should decide on the authority of the event, the object identity; whether the identity is fixed, whether pricing and contract conditions are valid; and downstream operational and financial implications of the change. This is why literature reviewed has been organized in five layers of interaction that include interaction capture, semantic harmonization, transaction authority, event orchestration, and revenue intelligence and control. The interaction-capture layer records customer-facing commercial signals. Semantic harmonization combines business objects and reference data. The transaction-authority layer determines which system owns specific state changes. Event orchestration takes care of propagation, timing and exception logic. The completeness, compliance, and performance impact are tracked by revenue intelligence and control. Any failure of any of the layers can wipe out the whole chain of operation.

Figure 1 presents a layered framework for CRM–ERP harmonization in real-time revenue operations. The figure does not consider CRM and ERP to be similar reflections since the literature suggests that commercial intent and transactional consequence are not equal and thus have to be controlled by different rules. Governance is also pervasive,

the figure shows that the real-time revenue operations are most often likely to fail when stewardship and location of rules are not embedded in the design of the integration.

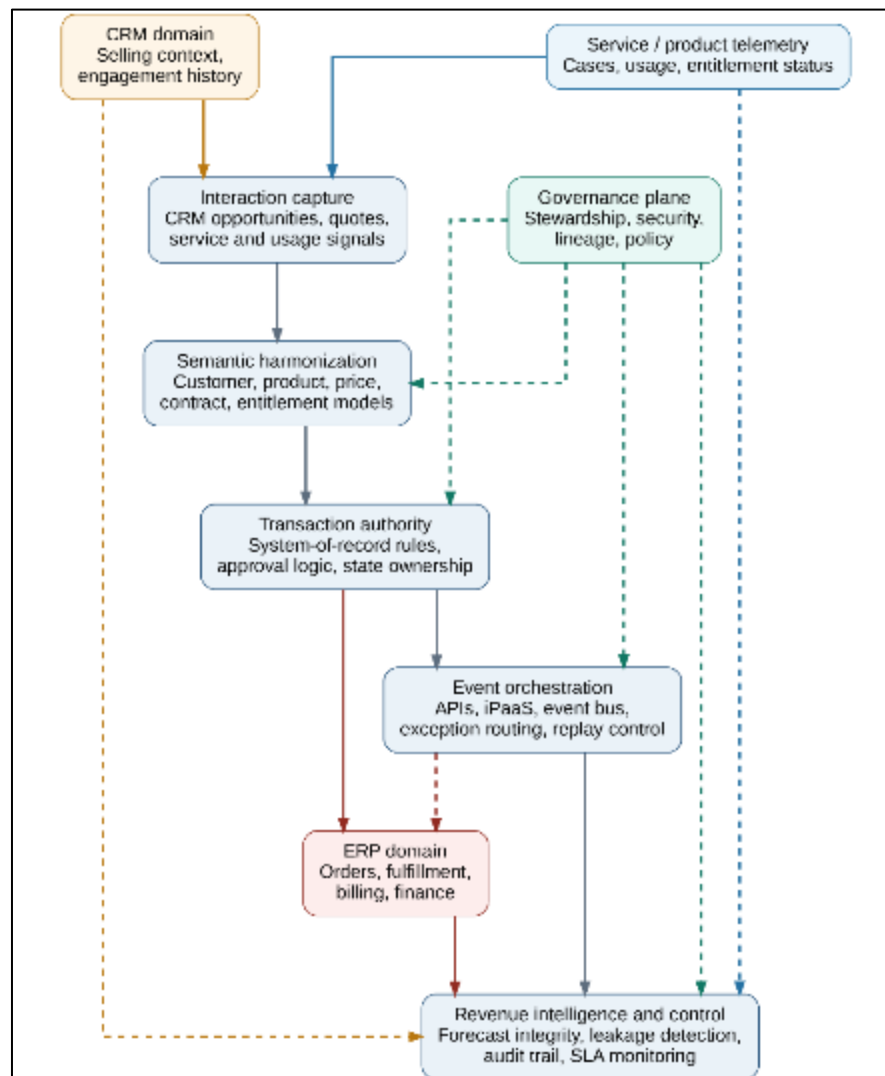


Figure 1 Layered framework for harmonizing CRM and ERP in real-time revenue operations

In general, the literature is qualitative and case-study and design-oriented architecture modeling and implementation analysis. The fact that CRM and ERP harmonization is highly contextual can be used to explain the domination. The structure of the industry, complexity of products, channel design, revenue model, legacy landscape and regulatory exposure are all factors that influence integration needs. Case studies capture this richness and are also especially helpful in discovering previously unknown dependencies such as the manual override paths, shadow pricing logic or various hierarchies of customers. Design-based research is also desirable since the discipline does not only entail passive observation, but rather, active configuration decisions. However, these approaches have recurring weaknesses due to the tendency of focusing methodologically in these approaches leads to repetitiveness. The validity of these studies is weak externally and there is a low level of cross-study comparability and the achieved results are normally anchored on locally based concepts of success. One design that works well in one application, e.g. a subscription-software context, may not work well in a different application, e.g. in an environment where manufacturing is asset-intensive or in a highly-channel-complex distribution environment.

Event-log analytics and process mining have become increasingly significant as they provide a transitional point between the architectural discourse and evidence of operations. These techniques allow the examination of the processes of cross-system flows directly after deployment. This ability is valuable to revenue operations since commercial slippage frequently manifests itself as waiting time, bounce-back loops, missing linkage keys or untraceable manual interventions. Nevertheless, process mining cannot be considered a universal solution. The quality of event

grammar and cross-system identifiers is as informative as the event logs. CRM-ERP orchestration may generate a lot of telemetry without a consistent analysis. In addition, the process mining is more diagnostic compared to ex ante design choice not accompanied by architecture modeling, or simulation.

Survey-based and configurational methods are less frequent, which address yet another requirement: generalization about the situations where some integration choices are value-creating. Such methods are able to compare the governance maturity, digital capability, process standardization and system modularity over a wider sample. Their shortcoming is that their commercial results during the revenue operations are hard to normalize. In one study, the term forecast accuracy can be used to mean pipeline prediction and in another study, it could mean known revenue variance. The latency can be the time of message transfer, order-release time, or the time of the dispute-resolution. In the literature, one is yet to find a powerful common measurement model. One significant implication of this review is that methodological pluralism is only welcome in cases where conceptual definitions are made sharper. More method diversity may produce more studies without generating cumulative insight; without more stringent construct discipline.

4. Discussion

The most consistent finding in the reviewed literature is that integration architecture in itself is not an explanation of revenue-operational improvement. Studies reporting fast API implementation or broad connector coverage do not necessarily demonstrate superior revenue-operational performance on the parameters of booking integrity, cleaner order execution, and reduced billing exceptions. The issue of architectural responsiveness is sensitive, though performance gains reported are mediated by issues of semantic discipline and clarity of ownership. In practice, near-real-time synchronization can exacerbate instability in cases of discrepancy in the definition of customer, product, discount, contract and entitlement objects in platforms. Reported successful outcomes are likely to be a combination of at least three design characteristics: explicit boundaries of system-of-record, persistent identity mapping between commercial and transactional objects, and non-destructive exception routing without impacting auditability. This result goes against a commonly accepted hypothesis regarding the discourse of enterprise integration that the lower the latency the better. Errors can propagate faster in real-time revenue operations than in batch interfaces when shared meaning is not established more efficiently in revenue operations than it has been in any batch interface because it is possible to propagate error much faster with no definitive meaning.

Figure 2 translates this pattern into an analytical trade-off map. The figure ranks typical integration patterns in how responsive and complex governance is, and there the absence of a universal approach that is practiced in the literature is apparent. The chart supports the perception that event-driven/API-led models are appealing for real-time revenue procedures, however, under a governance framework that is strong enough to control semantic drift, replay ambiguity, and distributed-rule fragmentation.

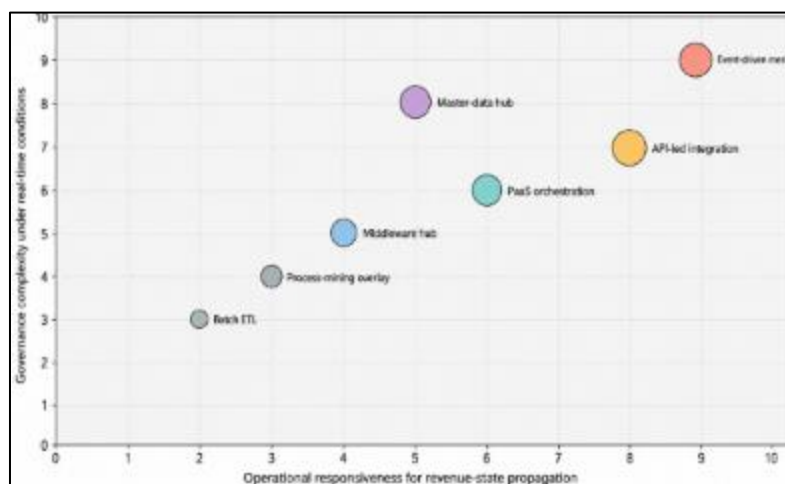


Figure 2 Analytical trade-off map of integration patterns for real-time revenue operations

The next significant finding is that of the centrality of semantic control. According to the literature, the integration objects that are the most problematic are customer records, but the issue of revenue operations is much broader than the problem of duplicating customers. Bundling of products, bargaining prices, amendments of the contracts, usage levels, right to the service, anchors to the renewal and characteristics of revenue allocation are more likely to cause a

more serious deviation downstream because of the high level of commercial freedom and rigid accounting and fulfillment demands. These findings are echoed by other studies that expect master-data governance, a greater level of control over the quality of orders, and downstream reconciliation, but also find an increase in the workload of stewardship and the cost of maintenance of models. A canonical model can improve interoperability but can be a bottleneck when there are commercial offerings which change quickly. On the other hand, domain-oriented models ensure local agility yet the load of reconciliation grows unless strong translation contracts exist between local domains. This contradiction is one of the most significant contradictions in the field that is yet to be resolved.

The literature also indicates that quote-to-cash and lead-to-renewal performance rely more on exception management than on the use of standard-path automation alone. Numerous enterprise integration projects streamline regular transaction processes and undervalue the business value of nonstandard scenarios. The amendment, partial delivery, split-bills or credit holds, disputed usage, partner involvement, service concessions or mid-term product replacement are some of the most prevalent scenarios in which revenue leakage can take place. Studies reporting improved outcomes are more likely to describe clear ownership of exception classes and connect exception events to both operational and financial viewpoints of the account. The finding contributes to the explanation of the fact that the use of purely technical dashboards does not necessarily contribute to the optimization of the revenue execution. Visibility alone does not create control without exception semantics and routing ownership.

This point is brought to the fore by the method comparison in Table 2. Case and design studies are very successful in uncovering certain architectural and governance dependencies. Conformance deviation and accumulation of waiting time are better detected using process mining. The configurational approaches and survey are useful in determining conditions that relate to success but tend to oversimplify the object-level semantics that create practical integration challenge. This trend can be summarized in table 2 and suggests that there is no one approach that can represent the whole revenue-operations problem space.

Table 2 Method comparison

Ref	Method	Strengths	Limitations
[10]	Multiple-case enterprise integration study	Captures cross-functional dependencies among sales, finance, fulfillment, and service; reveals where local system customization distorts enterprise revenue flow.	Limited comparability because each case defines “real time,” “accuracy,” and “integration success” differently.
[11]	Design science architecture modeling	Useful for specifying event contracts, authority boundaries, and canonical objects in a way that supports actionable framework design.	Reported value is often prospective or design-logic based rather than demonstrated through longitudinal revenue outcomes.
[12]	Process mining across CRM and ERP event logs	Directly exposes rework loops, stalled approvals, and hidden handoffs in quote-to-cash execution.	Dependent on event quality, timestamp integrity, and stable object identifiers across platforms.
[13]	Survey with capability and performance constructs	Supports broader pattern detection regarding governance maturity, modularity, and information visibility.	Commercial outcome measures often remain too coarse to isolate revenue-specific integration effects.
[14]	Configurational or comparative analysis	Identifies combinations of governance, standardization, and platform capability associated with stronger performance.	Results can become sensitive to case-selection logic and threshold specification.
[15]	Simulation or digital process modeling	Useful for stress-testing latency, queue build-up, and exception propagation under alternative orchestration rules.	Simulated flows may underrepresent informal workarounds and political ownership conflict.
[16]	Action research in system redesign	Strong for documenting iterative refinement of rules, stewardship, and workflow handoffs during implementation.	Researcher involvement can improve performance during the study period and complicate claims of transferable effect.

Figure 3 presents a comparative capability matrix rather than another trade-off chart because the literature suggests that the various methods of integration ought to be rated on the various revenue-operational capabilities rather than add up to a single score. The dot-matrix format allows comparison without implying unsupported numerical precision. The figure supports one of the key conclusions of the review, the most suitable framework is based on the priority: reduction of latency, semantic control, auditability, exception handling, or change isolation.

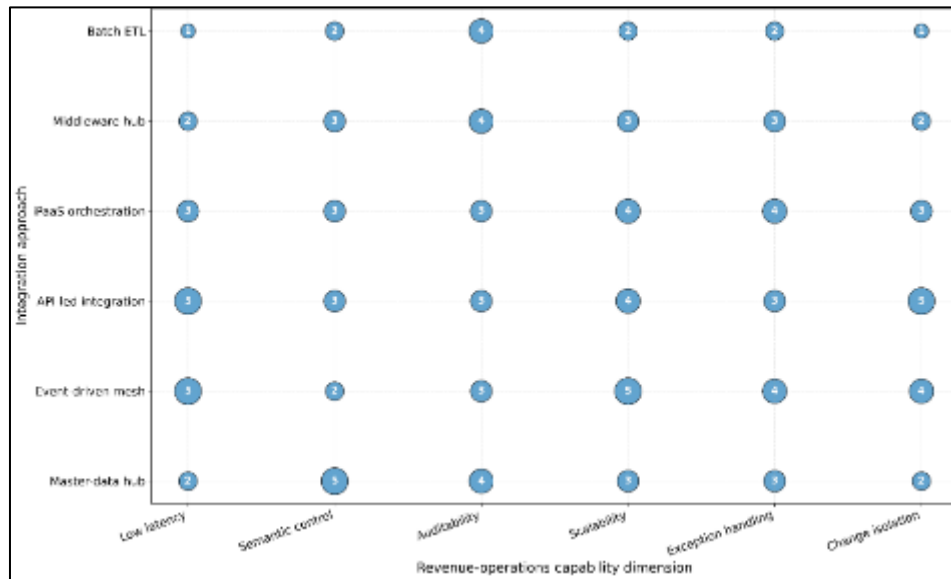


Figure 3 Comparative capability matrix for dominant integration approaches

There is a sharp difference between reported performance outcomes depending on system context. Event-based and API-driven designs tend to be more practical to the subscription- and consumption-based businesses because the amendments of the contract, changes in the usage, changes in the entitlement and signals to renewal are very regular and have an immediate billing effect. Asset-intensive production and distribution environments tend to have more cautious adoption patterns because of the structure of products, fulfillment dependencies, tax treatment, inventory and partner-structure visibility that place more semantic and transactional constraints. The other pattern is applicable to professional services and project-based organisations: harmonization of CRM and ERP is usually done according to the assignment of resources, milestone billing and change-order control rather than a specific master data of the product. The literature is therefore against architectural prescription that is one size fits all. Revenue model and process structure bring about the material change that harmonization must have.

Table 3 compares reported outcomes across these contexts. Areas in the table show that outcome measures are heterogeneous and tend to work at varying levels of proximity to revenue realisation. Other measures are proximal e.g., synchronization delay or order fallout. Some of them are in-between like the invoice correction frequency or a dispute cycle time. A minor set is strategically distal, e.g., forecast reliability, expansion capture or leakage reduction. This stratification is important as numerous researches are based on inferred revenue improvement by utilizing operational proxies without entirely applying commercial consequence.

Another trend is related to analytics and forecasting. Integrated CRM-ERP environments are often associated with the literature with greater visibility of revenues, but a more detailed examination shows that forecast returns are associated with the discipline of transactions and not the dashboard sophistication. Revenue dashboard can consolidate pipeline, booking, billings and cash pointers in near real-time, but forecast integrity remains weak when there is no semantic congruency amongst upstream states. CRM stages on pipelines can exaggerate commitment if ERP credit, inventory, or contract validation is incomplete. When amendment indicators are missing, usage or fulfillment indicators are incomplete, the billing projections may overstate realization. The definition of revenue observability as a control problem, based on event lineage, state validity and exception closure, is thus more likely to be found in high-performing studies, rather than as an analytics problem.

Table 3 Results comparison

Ref	System / Context	Metric / Basis of Comparison	Outcome
[11]	Subscription software with frequent contract amendments	Amendment-to-billing propagation time; renewal signal visibility	Faster contract-state alignment and fewer missed billing updates when event propagation is paired with contract-object stewardship.
[12]	Multi-entity manufacturing with complex product bundles	Quote-to-order accuracy; order fallout linked to product and price mismatch	Stronger canonical product and pricing control lowers downstream correction effort but raises model-governance overhead.
[13]	Omnichannel distribution with partner involvement	Customer hierarchy consistency; order and invoice dispute incidence	Account and party-role harmonization reduces dispute creation, especially in payer and ship-to misalignment scenarios.
[14]	Professional services and project billing	Change-order traceability; milestone invoicing completeness	CRM-ERP linkage improves billing completeness only when resource, scope, and commercial approval states are jointly governed.
[15]	Service-led renewal environment	Case-to-renewal conversion basis; entitlement status linkage	Integration of service incidents and entitlement state into CRM improves renewal prioritization and expansion targeting.
[16]	Hybrid cloud and legacy landscape using iPaaS	Connector deployment speed; transformation-rule transparency	Deployment speed rises, but business-rule opacity increases when mappings accumulate in proprietary middleware layers.
[17]	Event-driven orchestration across sales, finance, and service	Exception resolution time; replay and duplicate-event control	High responsiveness is achievable, although duplicate-event governance and replay policy become critical for financial reliability.

The second contradiction that has been brought to light in the literature is the old-standing struggle of centralization and autonomy. Centralized integration points and master-data controls are used to enhance master-data, governance, and audits. At the same time, centralization can stifle the commercial adaptation when local areas need to provide new bundles, discount plans, billing, or partnering plans. Domain oriented and event driven patterns will be more agile and less coupled at the expense of such flexibility being a greater governance overhead. This dilemma is heightened by revenue operations since both the speed and control are important at the same time. Any change that is delayed may result in revenue opportunity loss, but an unmanaged change may result in the recognition error or customer-facing failure. The most compelling research is thus not that in support of pure centralization or pure autonomy. Conversely, successful designs appear to provide firm control to objects and state changes with economic implications and local control to interaction context, interaction metadata and nonauthoritative indicators.

Figure 4 illustrates the application-level implication of this conclusion. The figure depicts that there is a closed-loop revenue-operating flow in which CRM, ERP, service signals, and a common event fabric interrelate during the lead-to-renewal cycle. The figure shows that harmonization is not a linear but a cyclical process: renewal and expansion are not merely by-products of billing, fulfillment, and service data; they are re-entry points into the revenue cycle.

Comprehensively, the literature is in favor of a subtle interpretation. Harmonizing CRM and ERP in real-time revenue processes cannot be simplified to a quicker interface, a new middleware purchase, or to one master-data project. Successful integration is associated with coherent alignment among architectural responsiveness, semantic accuracy, authority rules, exception control, and revenue-oriented observability. The discrepancies are still too big, particularly when it comes to the standardization/agility balance. But the research suggests as a whole that the discipline has passed beyond connectivity and is now in need of integration structures which are explicitly conceived in terms of commercial state, financial impact and closed-loop operational responsibility.

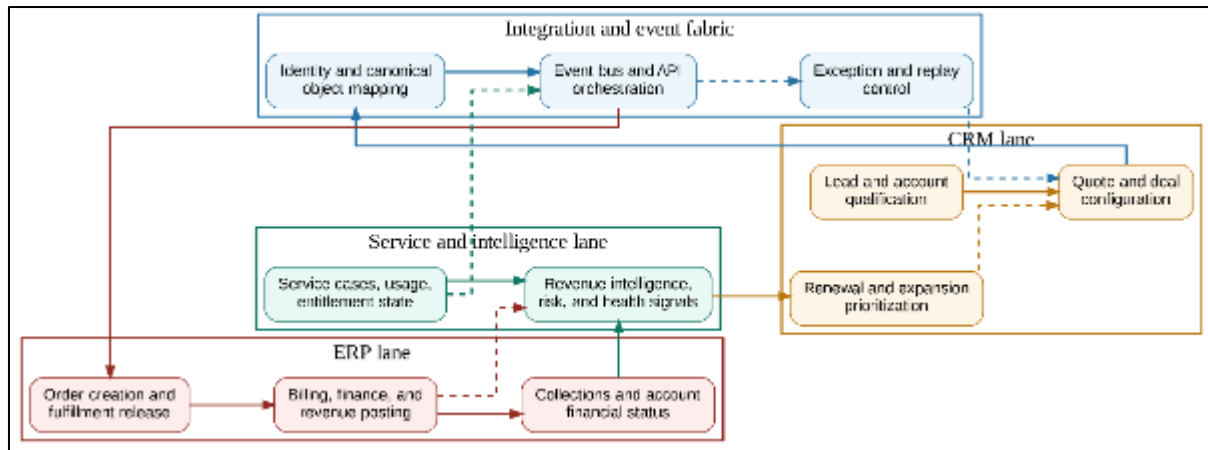


Figure 4 Closed-loop application architecture for lead-to-renewal revenue operations

5. Future Directions

The first research priority is greater conceptual clarity regarding what is meant by the term “real time” in regard to revenue operations. Existing literature tends to consider real time as a generic desirable attribute, and not to differentiate among message transfer speed, state visibility delay, exception closure time, and financially authoritative posting latency. Future work should develop a layered temporal model which isolates interaction immediacy, operational synchronization and financial finality. This model could be used to explain when low latency improves revenue execution and the occurrence of additional validation delay as preferred to manage quality. This is a crucial distinction when it comes to the dynamic pricing, consumption billing, revenue allocation or regulatory review.

The second priority is the development of formal models of the business objects which are relevant to revenue, and cross-system identity. The literature which has been reviewed appreciates the importance of harmonization of customers, products, pricing, and contracts but falls short of giving robust object ontologies and reusable to meet the current revenue models. Studies of canonical models are too generic to the complexity of amendments to subscription, bundling, entitlement status, partner paths, as well as charges based on usage. It might be analyzed in future scholarship in terms of identity graphs, semantic versioning of commercial objects, and event contracts which differentiate descriptive state and authoritative state. Research in this direction would also improve the process mining and observability research because the event interpretation is also based on clear objects.

Methodological development is a third priority. More comparative empirical studies that associate integration design to revenue-operational results using explicit mechanisms are needed in the field. Multi-case studies may also prove useful, although hopefully in the future comparisons between systems contexts will be performed in a systematic manner in terms of revenue model, industry, system landscape and the governance maturity. Longitudinal research would be in a position to answer the question of benefit persistence beyond the time when the initial deployment period has passed, and the local workarounds and rule drift are likely to accrue. The mixed-method designs can be particularly promising since the architecture analysis, event-log evidence, and performance measurement are applied to various levels of the issue. Controlled simulation can play an even bigger role when the models reflect exception burden, replay logic and stewardship effort (as opposed to idealized straight-through processing).

The fourth important direction is intelligent automation where control constraints are used. Agentic AI and machine learning are starting to affect quoting, lead scoring, anomaly detection, dispute routing, and renewal prioritisation. However, the literature has failed to give an adequate response to how such capabilities should interplay with authoritative CRM-ERP integration layers. Further research should consider the auditability of automated decisions, propagation of model errors through revenue processes and governance of AI-supported changes of state in commerce. The partner-intensive environments should also address other areas such as privacy, regional compliance, platform lock-in, and ecosystem integration. Interdisciplinary agenda is, consequently, the most promising one: enterprise architecture, process management, accounting information systems, service operations, and data governance all have their share of the puzzle, whereas real-time revenue operations need a framework that would tie all of those areas together.

6. Conclusion

The reviewed literature suggests that harmonizing CRM and ERP for real-time revenue operations is a multidimensional design challenge which requires architecture, semantic control, process visibility and governance to work together continuously. The strongest conceptual message is the fact that the integration quality cannot be inferred from connectivity alone. Value creation through real-time propagation is possible only when commercial and transactional states remain understandable, authoritative, and traceable throughout the revenue cycle. To that end, successful structures are based on strict object-identity treatment, system-of-record boundaries, event semantics, exception ownership, and revenue-oriented observability.

The literature also shows that the field is still disjointed. There is a lack of direct scholarly research on revenue operations and the literature on enterprise integration, process mining, data governance and digital platforms provide pertinent yet incomplete information. This breakdown has been characterized by repeated conflicts of centralization and domain autonomy, canonical patterns and local adaptability, local sensitivity and governance burden. Those tensions notwithstanding, a clear direction comes out. The way forward in the future is probably to be the integration structures which view revenue implementation as a closed operating system instead of a series of detached handovers. This approach can help build a superior level of business responsibility, reduce the amount of reconciliation, and increase the reliability with which customer-facing actions translate into financially valid outcomes.

References

- [1] Abraham, R., Schneider, J., & vom Brocke, J. (2019). Data governance: A conceptual framework, structured review, and research agenda. *International Journal of Information Management*, 49, 424–438.
- [2] Alhassan, I., Sammon, D., & Daly, M. (2016). Data governance activities: A comparison between scientific and practice-oriented literature. *Journal of Enterprise Information Management*, 29(2), 300–316.
- [3] Ali, M., & Miller, L. (2017). ERP system implementation in large enterprises: A systematic literature review. *Journal of Enterprise Information Management*, 30(4), 666–692.
- [4] Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Maggi, F. M., Marrella, A., Mecella, M., & Soo, A. (2019). Automated discovery of process models from event logs: Review and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 31(4), 686–705.
- [5] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. *IEEE Software*, 33(3), 42–52.
- [6] Cichy, C., & Rass, S. (2019). An overview of data quality frameworks. *IEEE Access*, 7, 24634–24648.
- [7] Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, 150, 77–97.
- [8] Gölzer, P., & Fritzsche, A. (2017). Data-driven operations management: Organisational implications of the digital transformation in industrial practice. *Production Planning & Control*, 28(16), 1332–1343.
- [9] Hübner, A., Holzapfel, A., & Kuhn, H. (2016). Distribution systems in omni-channel retailing. *Business Research*, 9(2), 255–296.
- [10] Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. *IEEE Software*, 35(3), 24–35.
- [11] Lemon, K. N., & Verhoef, P. C. (2016). Understanding customer experience throughout the customer journey. *Journal of Marketing*, 80(6), 69–96.
- [12] Merino, J., Caballero, I., Rivas, B., Serrano, M., & Piattini, M. (2016). A data quality in use model for big data. *Future Generation Computer Systems*, 63, 123–130.
- [13] Misangyi, V. F., Greckhamer, T., Furnari, S., Fiss, P. C., Crilly, D., & Aguilera, R. (2017). Embracing causal complexity: The emergence of a neo-configurational perspective. *Journal of Management*, 43(1), 255–282.
- [14] Pansari, A., & Kumar, V. (2017). Customer engagement: The construct, antecedents, and consequences. *Journal of the Academy of Marketing Science*, 45(3), 294–311.
- [15] Sedera, D., Lokuge, S., Grover, V., Sarker, S., & Sarker, S. (2016). Innovating with enterprise systems and digital platforms: A contingent resource-based theory view. *Information & Management*, 53(3), 366–379.

- [16] Soltani, Z., & Navimipour, N. J. (2016). Customer relationship management mechanisms: A systematic review of the state of the art literature and recommendations for future research. *Computers in Human Behavior*, 61, 667–688.
- [17] Taibi, D., Lenarduzzi, V., & Pahl, C. (2017). Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation. *IEEE Cloud Computing*, 4(5), 22–32.
- [18] Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Dong, J. Q., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, 122, 889–901.
- [19] Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *Journal of Strategic Information Systems*, 28(2), 118–144.
- [20] vom Brocke, J., Zelt, S., & Schmiedel, T. (2016). On the role of context in business process management. *International Journal of Information Management*, 36(3), 486–495.