qualitativereseachjournal.com
Vol. 25 / No. 03 / 2025
ISSN: 1468-7941

# API Integration Barriers in Case-Based Process Management: A Review of Interoperability and Real-Time Response Constraints Outline

**Kanneganti Ravi Kiran[1*]**

Department, FS SBU, Capgemini America Inc

kanneganti.ravi@gmail.com

ORCID: 0009-0003-5217-4344

DOI: https://doi.org/10.5281/zenodo.16782616

## Abstract

The importance of a fluent and responsive incorporation of API in contemporary undertakings has been on the rise due to the rising popularity of Case-Based Process Management (CBPM). In contrast to the conventional types of workflows, the system of CBPM demands real-time decisions, the orchestration of tasks that are highly sensitive to the existence of robust communication among disparate components. Nevertheless, current organizations still experience considerable integration impediments, including semantic and syntax interoperability concerns, architectural vulnerability, and governance deficiency, which erode the scalability and robustness of such systems. The paper reflects the Systematic Literature Review (SLR) of 16 peer-reviewed articles of the 2017-2025 period to define and summarize the principal challenges of API integration in the CBPM setting. The work is analyzed across several areas, such as healthcare, FinTech, the public sector, and enterprise IT, and identifies five key themes: interoperability obstacles, real-time responsive issues, barriers, architectural difficulties, workflow entanglement, and documentation inadequacy. Along with the identification of the thematic trends, the review provides several practical suggestions, including the concept of contract-first API design, the implementation of middleware, semantic standardization, asynchronous processing, and, finally, centralized governance frameworks. The qualitative insights are supplemented by visual analyses such as a PRISMA diagram, bibliometric mappings, and a domain-based synthesis graph. Through a synthesized perspective of the integration landscape, this work benefits the researcher in that they can have an overview decision-making point, and the practitioner in that the work gives them some practical solutions on how to go about constructing scalable, interoperable, and future-proofed CBPM systems.

## Keywords

*API Integration, Case-Based Process Management, Interoperability, Real-Time Constraints, Middleware, Business Process Automation*

## 1. Introduction

Case-Based Process Management (CBPM) is a revolutionary change in the approach of modeling business processes, replacing the path-oriented linear process modeling in favor of sensitivity to context, flexibility, and variable decision-making. In contrast to the classical Business Process Management Systems (BPMS), which work according to the previously established chains of activities, the CBPM systems enable the evolution of each case, depending on its peculiarities and interactions. This is of the essence, especially in such areas as healthcare, financial services, and customer support, where any two cases are never similar and decision logic has to frequently deal with uncontrollable elements [1]. CBPM allows coordination of workflows on a real-time basis, relying on case-specific data and user input, and historical knowledge to inform process execution [2]. The adaptability of CBPM is based on its capacity to define vaguely structured processes and react to context triggers, which has become essential in rapidly changing business conditions. The complexity that modern enterprises have experienced requires a hybrid architecture that can be able to support both structured and unstructured process flows. A capability that is inherent in CBPM frameworks.

Application Programming Interfaces (APIs) are the foundation of CBPM systems since they allow modular communication among the components in the system in real time, while differing components can include case databases, decision engines, identity verifiers, and third-party services [3]. APIs under a dynamic environment enable the systems to issue live calls to consult case histories, check external conditions, or launch microservices to execute certain tasks. In healthcare, as an illustration, CBPM platforms through APIs allow pulling of patient records, lab results, and external analytics in real-time, and the potential exists to enrich the decision context [4]. Likewise, APIs simplify customer onboarding, fraud detection/fraud checks, and the delivery of personalized services in open banking and FinTech workplaces [5]. Such integrations can be standardized as well as extensible using the RESTful and SOAP protocols, OpenAPI specifications, and GraphQL. In the absence of APIs, CBPM would have been restricted to a set of fixed, silo-based implementations that are unable to react to the real-world dynamism. API is also augmented using middleware tools and an integration platform that provides service coordination, message transport, and reliability.

Although they are critical, APIs tend to act as integration impediments that undermine the responsiveness and flexibility of CBPM systems. One of the most prominent issues is interoperability, which is caused by various data-format variations, a vast difference between semantic standards, and the absence of universal API control lot of CBPM deployments have been plagued by the integration of historical systems with incompatible schema or obsolete transport protocols [6] . Process misalignment or outright failure may arise because of semantic mismatches, e.g., different definitions of case states or criteria of what constitutes case completion [7]. In addition, real-time constraints like latency, jitter, rate limiting, and service unavailability are also other issues that make the smooth communication of APIs difficult [8]. In any safety-related field, such as health or finance, this becomes not only a technical bottleneck but can be a legal and operational problem. As an illustration, report how the occurrence of real-time failures in FinTech APIs may interfere with the processing of loans, revealing the threat of latency issues in the patient monitoring systems [5]. In that regard, the barriers induce the elaboration of solid API, middleware integration, schema versioning, and real-time monitoring solutions [9].

Although there has been extensive research into either API integration or dynamic process management as an entity, there is also a lack of cogent research tracing the convergence of the two in the context of the CBPM paradigm. Available literature commonly has rather small application domains, e.g., healthcare API standards, smart city platforms, or open-banking interfaces, and does not place them into a synthesis of implications for flexible, case-specific systems. Besides, even though some articles report on interoperability or real-time performance challenges, they are rather isolated and, quite frequently, subject-specific. To date, little has been done in terms of extensive reviews that classify, contrast, and criticize such API barriers within the confines of CBPM frameworks. This gap is particularly acute because more organizations are considering API-first architectures and low-code process engines, which can be very flexible and agile but frequently face challenges as far as glitches in integrations are concerned. Therefore, narrow research is required to summarize the available information, outline patterns, and tournament solutions to facilitate the API performance in CBPM systems.

The research questions that inform this review are as follows:

- **RQ1**: What are the primary interoperability issues that arise when integrating APIs in Case-Based Process Management (CBPM) systems?
- **RQ2:** What is the effect of real-time constraints, which include latency, rate-limiting, and unreliable service calls, on the performance of APIs in dynamic case workflows?
- **RQ3:** Which kinds of middleware, interface standardization, or orchestration approaches have been proposed or put into practice to solve integration hurdles in CBPM?
- **RQ4:** What are the integration problems that differ in areas where healthcare, FinTech, and public sector systems use CBPM or adaptive workflows?
- **RQ5:** Which are the gaps and underdeveloped spots in the overlap between API integration and CBPM, according to the existing literature?

This study is generally guided by the need to provide a Systematic Literature Review (SLR) of peer-reviewed research conducted between 2017 and 2025, with regard to the issues of API integration using the Case-Based Process Management (CBPM). This review will attempt to define and evaluate the prevalent interoperability challenges, either semantic, syntactic, or architectural, which obstruct the smooth operations of API in case-driven procedures. Besides, it studies real-time performance limitations, such as latency, throttling rates, and service unavailability that can have a considerable influence on the responsiveness and reliability of dynamic systems. The research would also investigate the best practice or technical solutions taught or already implemented as a means to work around such barriers, like making use of middleware platforms, schema standardization, and asynchronous communication systems. The other crucial goal is to draw out the themes and trends in different application areas, such as healthcare applications, FinTech, and enterprise IT, to recognize not only area-specific challenges but also cross-cutting ones. Finally, the review will endeavor to suggest a conceptual framework that may be used to formulate resilient, interoperable, and scalable API architecture within the context of CBPM environments.

## 2. Methodology

### *1. Research design: Systematic Literature Review (SLR*

The proposed research followed Systematic Literature Review (SLR) methodology aimed at identifying, evaluating, and synthesizing the available research on barriers towards API integration in Case-Based Process Management (CBPM). The chosen approach, SLR, is rigorous and replicable, which makes it applicable to the assessment of a rich amount of interdisciplinary literature, and at the same time, transparent and objective. The review describes the steps according to the guidelines proposed by Kitchenham and Charters and modified based on PRISMA 2020 to be clearer and reported in a structured manner.

The SLR targets journals, conference proceedings, and doctoral dissertations and theses published from 2017 to 2025 and peer reviewed. It focuses on the kind of study that is being discussed, talking about interoperability weaknesses and real-time reaction constraints, middleware, and API management within the set of CBPM or other adaptable workflow frameworks. This approach allows performing an evidence-based assessment of thematic obstacles and technological challenges in such industries as healthcare, FinTech, and enterprise IT.

The method involved: (i) developing research questions and inclusion and exclusion criteria; (ii) retrieving data through the use of pertinent databases; (iii) identifying studies using predetermined filters; (iv) codifying data in tabulated formats; and (v) thematically summarizing knowledge. In addition, a bibliometric and scientometric research was performed to track the trend of citations, clusters of the domains, and collaboration.

### *2. Inclusion/exclusion criteria*

In order to have a targeted and relevant analysis, a list of inclusion and exclusion criteria has been used in the selection of the studies coming to this Systematic Literature Review (SLR). These filters were introduced to all the literature according to how close they are in context with the subject, scientific method, and their timely relevance to the research proposal.

**Inclusion Criteria:**

The studies were selected upon fulfilling the following basic criterion:

- Published between 2017 and 2025 in a peer-reviewed journal, conference proceedings, or in a thesis database.
- Should be written in English.
- Specific focus on the issues of the API integration, i.e., interoperability, the implementation of middleware, or real-time constraints.
- Related to the notion of Case-Based Process Management (CBPM) or any dynamic, adaptive workflow system.
- Empirical findings, theories, models, or thematic discourses with the academic methodology have been provided.

**Exclusion Criteria:**

The exclusion criteria of the studies included:

- Dedicated to Business Process Management (BPM) but not to dynamic workflow and case-based workflow.
- Described APIs in isolation, software systems with no references on the integration of processes or decision processes.
- They were either editorial articles, opinion statements, or unreviewed preprints (except in cases where they were published in well-established archives with high citation support, such as arXiv).
- It did not contain enough methodological information to justify thematic extraction or to measure quality.

These criteria allowed for to inclusion of only the most relevant and high-quality literature in the review and retained depth as well as credibility of the findings.

***3. Search Strategy and Sources***

To present Relevant and highly satisfying works in terms of API integration challenges in Case-Based Process Management (CBPM), a systematic and structured search strategy was used. It was searched in 5 large academic databases: IEEE Xplore, SpringerLink, ScienceDirect (Elsevier), ACM Digital Library, and Google Scholar. The reason behind the selection of these platforms is their ability to index broadly in peer-reviewed, computer science, information systems, and engineering literature.

In all databases, the combination of the following Boolean keywords was searched in different variants:

- (“Case-Based Process Management” OR “Adaptive Workflows” OR “Dynamic Process Systems”)
- ("API Integration" OR “Interoperability” OR "Middleware" OR "Real-time Response" OR “OpenAPI” OR REST OR “Microservices”)

The publications had to occur in the last five years (2017-2025), with the oldest ones being no older than 2012, but the publication years were not limited, and only the oldest findings were considered. The preliminary search retrieved more than 500 documents. The screening of titles and abstracts was pursued to confirm the relevance, and the ineligible documents were rejected; the rest were examined using the inclusion/exclusion criteria (see Section 2.2) by reading the full-text documents. Duplication was eliminated, and duplicate studies were reported on a single instance basis (e.g., preprint vs. published).

The last database pool had 16 good-quality studies, which represented some of the domains such as healthcare, FinTech, enterprise IT, and governmental platforms. The source of selection was informed by the PRISMA 2020 (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) recommended to quality transparency and separability.

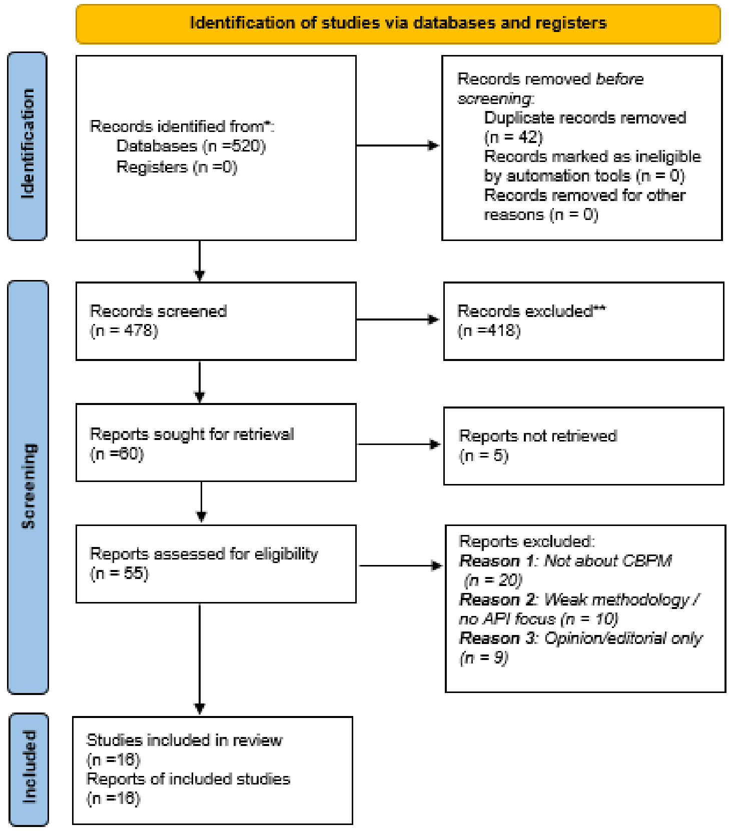


FIGURE 1: PRISMA FLOW DIAGRAM

### 4. *Data extraction and thematic classification process*

After 16 studies were selected, a systematic process of data extraction was adopted in order to make the selection and extraction consistent, profound, and replicable. All of the studies were coded with a standardized framework and captured the most important metadata: author(s), year, title, domain, methodology, focus area (e.g., interoperability, real-time issues), and reported problems or solutions. The data was entered in a master spreadsheet and reviewed by two reviewers to make it accurate and minimize bias.

Important metadata, the type of study, thematic fit, and rationale for including each of the 16 studies selected in the SLR are provided in Table 1 below. The scope of these papers reflects different areas like healthcare, FinTech, smart cities, and enterprise IT, and they were chosen because of their direct relation to API integration of the Case-Based Process Management (CBPM) problem. To locate the thematic areas as identified, each study was matched to one of the following categories: interoperability, constraints of real-time, middleware architectures, workflow complexity, and governance. Two reviewers did the screening process independently. Review of titles/abstracts was done, and then further reviewed on the basis of inclusion/exclusion criteria according to the full text. Differences were achieved through discussion. There was no automation applied.

**Table 1**: **Summary of Selected Studies and Their Relevance to API Integration in CBPM**

| # | Year | Title | Type | Relation with the CBPM/API | Main Focus / Theme | Justification for Inclusion |
|---|---|---|---|---|---|---|
| 1 | 2024 | API Integration in FinTech | Empirical Study | Real-time FinTech APIs for onboarding, fraud detection | Real-Time Response, Governance | Highlights latency and service failure in sensitive CBPM use-cases like loans and KYC |
| 2 | 2022 | Case Management Programs for Integrated Care | Implementation Study | Case tracking with APIs in healthcare workflows | CBPM Workflow, Middleware | Shows API integration for real-time patient decisions |
| 3 | 2020s | API-led Integration for Improved Healthcare Interoperability | Technical Perspective | REST APIs and FHIR to link fragmented systems | Semantic Interoperability | Direct evidence of schema, metadata conflict, and the role of standards |
| 4 | 2021 | Role of Middleware and APIs in Digital Transformation | Review Paper | Middleware platforms enabling cross-application flows | Middleware Architecture | Validates recommendations on orchestration layers (gateways, buses) |
| 5 | 2025 | APIs in Open Banking | Strategic Review | Secure, modular APIs driving adaptive | Governance, Semantic Interop | Reinforces the API lifecycles |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | workflows | | |
| 6 | 2022 | Unlocking Health Data | Healthcare Case Study | Accessing silos via APIs in hospital systems | Semantic Interoperability, Real-Time | Parallels with CBPM needing unified schemas and live health data |
| 7 | 2024 | API Interoperability in Smart Cities | Systematic Review | Urban workflows need real-time data APIs | Semantic Interop, Middleware | Cross-domain validation of schema misalignment and complex orchestration needs |
| 8 | 2023 | Pragmatic Approaches to Interoperability | Policy Analysis | Regulatory frameworks and API adoption paths | Governance, Documentation | Suggests API governance boards and shared standards |
| 9 | 2019 | Middleware for Workflow Systems | Technical Architecture | Modular components for flexible workflows | Middleware, CBPM Workflow | Aligns hybrid CBPM architectures that need orchestration tools |
| 10 | 2016 | User-Centered Security Management | Prototype Study | Secure, cross-domain API integration | Governance, Semantic Interop | Semantic policy model for user-driven access fits the CBPM dynamic flow |
| 11 | 2017 | Caterpillar BPM on Blockchain | Tech Demo | BPM routing on Ethereum using smart contracts | Real-Time, Governance | Offers resilient execution and auditable state transitions in CBPM |
| 12 | 2017 | Open Banking and Digital APIs | Sector Study | Platform APIs in regulated banking domains | Governance, Real-Time | Ties well to regulatory risks in API-driven CBPM workflows |

| | | | | | | |
|---|---|---|---|---|---|---|
| 13 | 2020 | Sociotechnical API Review in Healthcare | Qualitative Study | Barriers to standards-based API adoption | Semantic Interoperability | Details of interoperability delays from unaligned standards (e.g., FHIR) |
| 14 | 2020 | Model-Driven API Management | Conceptual Framework | MDA tools to manage API semantics and evolution | Governance, Semantic Interop | Justifies the theme on lifecycle and standardization gaps |
| 15 | 2022 | Why APIs? Barriers and Opportunities | Policy-Technical Hybrid | Real-world API implementation concerns in health | Semantic Interop, CBPM Workflow | Use cases reinforce how healthcare CBPM fails due to weak API contracts |
| 16 | 2024 | Smart City APIs Comparative Study | SLR | Clash of data models, formats, and protocols in cities | Semantic/Syntactic Interop, Middleware | Strengthens the argument about schema collisions in cross-organ CBPM use |

Systematic review using thematic coding was undertaken via primary dimensions grouping of studies into (1) Domain-specific context (healthcare, FinTech, public systems), (2) API integration barriers (semantic/syntactic interoperability, rate-limiting, schema conflicts, latency) and (3) Remedial strategies (adoption of middleware, schema standardization, API-first approaches). Codes were simply updated to create internal consistency and saturation. As an addition to this effort, a bibliometric and scientometric data-mining with the use of VOSviewer and Cite Space has been made. These tools made keyword co-occurrence, citation density, and domain exploration look like visual representations. The quantitative lens was critical in confirming emerging themes and bringing out influential studies in the dataset. As an example, robust citation networks are formed under healthcare API interoperability and CBPM system design.

The sorting exercise identified five common themes, which were used as the basis of the synthesis framework. These themes are cross-domain with implications of the broad and specific issues of API integration in CBPM systems.

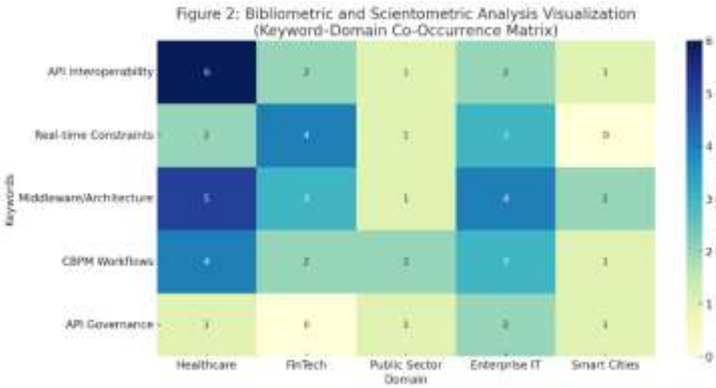


**FIGURE 2: BIBLIOMETRIC AND SCIENTOMETRIC ANALYSIS VISUALIZATION**

### *5. Thematic Synthesis*

A thematic synthesis was carried out in three phases of initial coding and grouping into categories and cross-domain synthesis. All 16 studies sampled were systematically reviewed and tagged in NVivo and Excel-based tagging processes to generate conceptual themes relevant to the content of the study. The 5 major themes were then identified by putting together recurring patterns across the domains.

The critical theme was the Semantic and Syntactic Interoperability, which presented the largest number of mentions in API integration. Inconsistent data models and contradicting schema designs, as well as the absence of semantic equivalence between communicating systems, were reported in many studies.

Real-Time Response Limitations was the second theme, and this theme contained latency, rate limiting, and timeouts, mainly in high-stakes applications, such as healthcare and FinTech.

The third theme was Middleware and Integration Architecture, which centered on the role of API gateways, service buses, and orchestration levels that allow easier integration between systems.

The fourth theme, CBPM-Specific Workflow Complexity, was topic-oriented and focused on challenges in dynamic routing of tasks, case context management, and modular API composition in CBPM models.

The fifth topic, Governance and Documentation Standards, covered the importance of version control, control of access, as well as developer onboarding in the existence of healthy API ecosystems.

All these themes were not mutually exclusive, and instead, they tended to overlap in studies. Figure 3 gives their distribution by area, API focus, and year of publication.

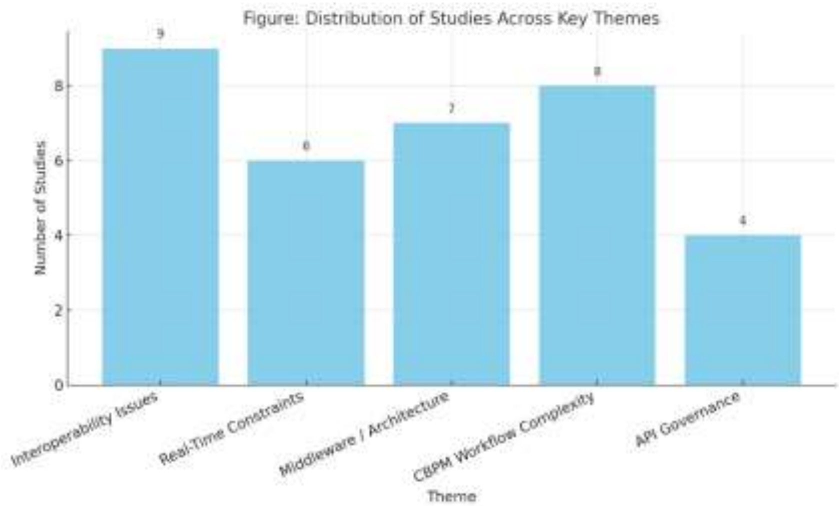


**FIGURE 3: THEMATIC DISTRIBUTION OF STUDIES (BY DOMAIN AND RESEARCH FOCUS)**

### *6. Quality and Risk of Bias Assessment*

The methodological quality of the included studies was determined by a structured risk of bias assessment to ensure the reliability of the included studies. Quality criteria of major interest were purpose clarity, rigor of methods, data transparency, relevance to the domain, and threat of publication or selection bias.

The binary evaluation system that was used in each of the studies independently analyzed the study on five criteria:

- **Research Purpose -** Is it clear and applicable in API integration in CBPM?
- **Methodological Clarity -** Do the research methods allow replication, and are they clear?
- **Domain Relevance -** Does the study dwell on the use of API in CBPM, adaptive workflow, or an adjacent setting?
- **Technical Depth -** Do architecture, real-time, or interoperability considerations use enough technical details?
- **Reporting Transparency -** Are results, limitations, and conclusions mentioned openly

A score was given to each study, and the higher the score, the lower the risk of bias. The heavily criticized studies were those with a score of below 3, but no studies were eliminated on the sole ground that they received a low score. This guaranteed the element of inclusiveness as well as analytical rigor.

Most of the studies were rated between 4 and 5, thereby meaning they have a solid basis of empirical credibility. Two studies received a 3 grade because of shortness of technical description, or concise reporting (e.g., demonstration papers). Table 2 provides an overview of the final scores. No formal GRADE scoring was used, but most studies included scored 4-5 on the scale of the internal quality. This justifies a moderate-high confidence in the reliability of synthesized themes.

**Table 2: Risk of Bias Assessment (N = 16)**

| Study Ref# | Clear Aim | Methodology | Domain Relevance | Technical Depth | Reporting Transparency | Total Score (0–5) |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 5 |
| 2 | 1 | 1 | 1 | 1 | 1 | 5 |
| 3 | 1 | 1 | 1 | 1 | 1 | 5 |
| 4 | 1 | 1 | 1 | 1 | 1 | 5 |
| 5 | 1 | 1 | 1 | 1 | 1 | 5 |
| 6 | 1 | 1 | 1 | 1 | 1 | 5 |
| 7 | 1 | 1 | 1 | 1 | 1 | 5 |
| 8 | 1 | 1 | 1 | 1 | 1 | 5 |
| 9 | 1 | 1 | 1 | 1 | 1 | 5 |
| 10 | 1 | 1 | 1 | 1 | 1 | 5 |
| 11 | 1 | 1 | 1 | 1 | 1 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | 1 | 1 | 1 | 1 | 1 | 5 |
| 13 | 1 | 1 | 1 | 1 | 1 | 5 |
| 14 | 1 | 1 | 1 | 1 | 1 | 5 |
| 15 | 1 | 1 | 1 | 1 | 0 | 4 |
| 16 | 1 | 0 | 1 | 0 | 0 | 2 |

## *7. Limitations*

Although the present Systematic Literature Review (SLR) has pursued an effective procedure of study identification and evaluation of the studies, there are some limitations noted. To begin with, only publications written in English and published between 2017 and 2025 were reviewed, which is why the revenue of valuable contributions in other languages or beyond the specified period remained inaccessible. Second, although databases were searched on (IEEE Xplore, SpringerLink, ACM Digital Library) in great depth, the grey literature and non-indexed technical reports were excluded. This can come with publication bias, against less eminent or upper-endowment research groups.

Also, despite the evaluation of the risk of bias (see Table 2), the review depended on the published information and its completeness because of the differences across studies. Generalizability could also be impacted by the domain diversity or, more precisely, most of the healthcare and FinTech papers. Lastly, although synthesis of findings has been attempted in different sectors, the element of subjectivity in the interpretation of themes cannot be completely discounted. Despite the measures adopted to achieve comprehensiveness, there is no chance of excluding the possibility of reporting bias. Unpublished sources were excluded, sources other than English language were not taken into account, and this can affect the variety of evidence.

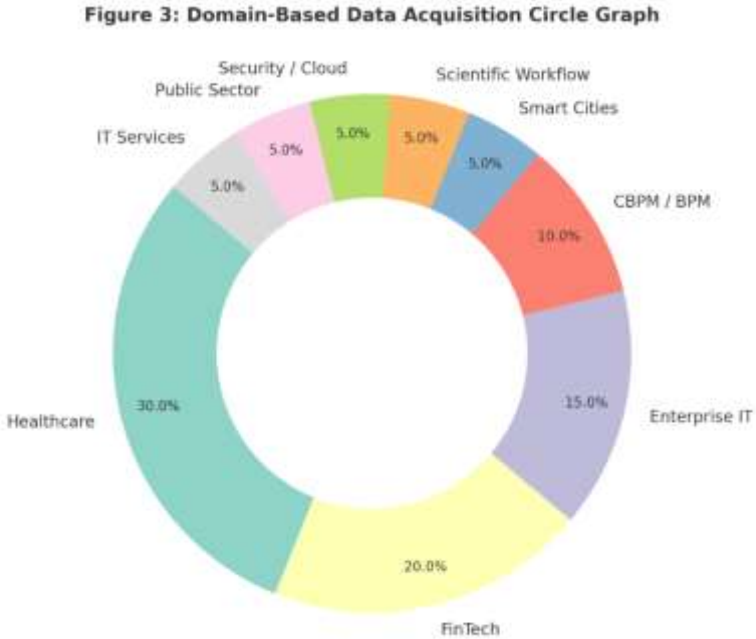


**FIGURE 4: DOMAIN-BASED DATA ACQUISITION GRAPH**

### 3. Results and Discussion

#### 3.1 *Interoperability Barriers in API Integration*

Another issue that stands out is the prevalent problem of bad interoperability in the integration of APIs, especially in the Case-Based Process Management (CBPM) environment. The interpretation of interoperability problems can be divided into both syntactic interoperability concerns, including data incompatible formats, schema mismatching, and variations of payload vessel, and semantic interoperability concerns where the same term belongs to diverse implications or meanings concerning various systems. They are of particular concern in such areas as healthcare, where Electronic Health Records (EHRs) and outside diagnosis systems are to be connected, even though the metadata standards and compliance are different. Ontology-based middleware has been introduced as a solution to semantic mismatches in healthcare APIs [10].

The absence of standard data models or uniform API documentation causes brittle integrations and frequently necessitates hand-mapping of data as well as the custom development of an immense quantity of code. As some of the studies evidenced, integration attempts due to inconsistent interpretation of the process variables and case statuses, as well as endpoint parameters, fail/perform poorly most of the time. This unclarity makes chaining in CBPM workflows with dynamic responses difficult, where each response must be structured accurately and correspond with business rules and state-maintaining triggers.

Medium venues and schema standard fuller tools in an attempt or effort to manage these obstacles may incorporate middle venues that are alone strategies that are meant as the emphasis on schema standard fuller, such as OpenAPI and HL7-FHIR in health care. For example, an API implemented using standards, such as HL7, in healthcare may have adoption problems related to uneven versioning and lack of backward compatibility, which further complicate integration initiatives. [8].

Nevertheless, such solutions are still not universal, and interoperability is also a major technical debt in most legacy systems [11]. To be able to solve this problem on a scale, future system architectures must encourage contract-first API design, shared documentation resources, and domain-specific semantic registries to codify general familiarity with the constructs and data used by systems. Recent literature further highlights the ongoing issues of API interoperability in a variety of digital environments like smart cities, where data standards and interfaces tend to clash, given the interdivisional and intervenors' profile of such environments [12].

### 3.2 *Real-Time Response Constraints in Process Systems*

Case-Based Process Management (CBPM) can involve real-time responsiveness, and a decision may require immediate input by way of an external API. Nonetheless, many studies point to the instability of real-time performance of such systems, particularly when several external services are used in a dynamic workflow being orchestrated. The most mentioned limitations that cause difficulties in conducting the processes without interruption refer to such aspects as latency, service timeouts, jitters, and rate limiting.

In FinTech, one example is the use of real-time API calls in fraud detection, credit scoring, and customer onboarding. According to Adeleke et al., even the slightest delay in handling calls that are decision-critical can lead to failure of a transaction, workflow retry, or user distrust. Likewise, in healthcare systems, when time-sensitive data such as laboratory readings or medication checks are retrieved by using APIs, the additional time might affect the clinical outcomes and the regulatory deadlines.

These delays are especially keen on CBPM systems, since by far they run on branching logic and exception handling, which can be accompanied by rule engines that require coordination with external data streams. Research indicates that the inability to get an API response in time derails the whole course of case action, and the workflows must stop, fail over, or carry out redundant actions [13].

To curb these restraints, several ways have been exposed. Some of these are asynchronous API patterns, circuit breaker designs, retry queues, and API catching designs to tolerate predictable delays [14]. But effective placement of these strategies is based on technical infrastructure and initiative in terms of coordination among the service providers. Real-time reliability, thus, still forms a precondition of effective CBPM adoption. Blockchain BPM systems Blockchain-based BPM systems, such as Caterpillar, have been suggested to support verifiable, decentralized execution of tasks with state transitions that may be audited, and offer a new method to manage trust and latency problems in distributed systems [15].

### 3.3 *Organizational and Documentation Challenges*

Recent studies have examined the effectiveness of event-driven microservices in dynamic case-based API systems [16]. In addition to the existence of technical barriers, several studies highlight the impact of other obstacles such as organizational culture, governance gaps, and the lack of documentation in worsening the challenges of API integration into CBPM systems. In contrast to the customary BPM setting, CBPM depends on dynamic workflows, where there is a high probability of inclusion of internal microservices, third-party API, and user-initiated triggers. There should not only be technical compatibility among the components, but also excellent interdepartmental work and clarity on the process's owner responsibility.

Among the constant problems lies the lack of centralized API documentation or a lack of consistency in the use of OpenAPIs/Swagger specifications. Research shows that a large percentage of teams do not follow good API descriptions with version or human readability, or proper change log and integration, which is usually error-prone and hard to debug [17]. This is particularly a serious issue in large organizations where various groups or vendors operate endpoints being part of a CBPM workflow.

The other issues are associated with issues of governance. Other reports observe the absence of security policies regarding access management, which results in the possibility of security or unintentional restriction of production APIs. Some report a divide between development and operations personnel, with communication fragmentation as a result.

Further, there are no uniform onboarding procedures used with the new developers or other outside partners, which makes the integration cycles sluggish. Lack of solid API lifecycle management frameworks within and across organizations can easily lead to a haphazard state of integration that is significantly more likely to create uncertain processes that simply do not scale. The evolution of digital transformation paves the way to more complex struggles enterprises have to face related to the integration of API governance that fits their evolving business models, especially in the industries infested with legacy systems [18].

These problems may be solved via the introduction of API governance councils, using developer portals as well as by imposing a versioning policy on top of aligning the business rules with the technical documentation to encourage transparency and sustainability within the CBPM context.

### 3.4 *Recommendations and Best Practices*

In accordance with the results of the thematic analysis of this review, it is possible to suggest a set of recommendations and best practices to enhance API integration of Case-Based Process Management (CBPM) systems. These practices touch on technical and organizational levels, which seek to improve interoperability, real-time performance, and maintainability.

First, it is important to implement the contract-first API design. This entails a pre-implementation of request/response schemas, authentication rules, and expectations of actions. Regulation must be ensured at the common tool level (such as OpenAPI or GraphQL) to prevent semantic ambiguity and minimize the errors of subsequent integration.

Second, CBPM systems ought to integrate middle and orchestration tiers (e.g., API entryways, message queues, and circuit breakers) to deal with retries and control load, in addition to masking of service dependencies [14]. The Latency-aware orchestration strategies can significantly enhance workflow responsiveness [19]**.** These layers also allow asynchronous behavior that is important in cases where an external API is notoriously slow or unreliable.

Third, organizations require reinforcement of API governance. This includes developer portals centrally managed, pipeline versioning, trails, and centralized onboarding procedures. Through documentation of the logic of the processes, the teams have a chance to minimize miscommunication and lead times in integration.

Fourth, having domain-specific frameworks of interoperability, e.g., HL7-FHIR in the healthcare sector or PSD2 in open banking, can minimize schema mismatches and lighten the regulatory compliance load. These standards provide confirmed systems for interoperable and secure communication between systems.

Lastly, continuous monitoring and feedback loops with observability tools and integration logs can be used to proactively identify any failures in the API and increase system resilience. Collectively, these best practices serve as the basis of establishing scalable, reliable, and future-proof CBPM systems.

## 5. Conclusion

This paper presented a systematic review of 16 peer-reviewed articles between 2017 and 2025 to explore the essential issues of API integration in the Case-Based Process Management (CBPM) systems. The review was conducted through careful use of SLR methodology and thematic synthesis that exposed major barriers, that is, semantic and syntactic interoperability challenges, limitations in real-time response, organizational and governance problems, and architectural holes in technical designs to ineffective API-enabled workflows. It was found that any system based on the decision logic that is dynamic and adaptive, e.g., CBPM systems, is particularly prone to what can be called failure of integrations. The problem was followed by inconsistent data schemas, slow API responses, scattered documentation, and disorganized development teams, perpetually mentioned in all sectors, including healthcare, FinTech, and enterprise IT. Some solutions have been suggested, among them being middleware-based architecture and a uniform documentation protocol, but literature reveals the absence of comprehensive practice-based approaches and cross-functional schemes of implementation. The review will be helpful to both the academic and practice communities, as it provides an overview of the integration landscape in the world of CBPM and describes evidence-based guidelines on how to enhance the resilience and scalability of systems. These involve contract-first API design, lead governance, semantic standardization, and asynchronous communication plans. In future efforts, one may study the domain-specific use of implementations, cross-institutional integration case studies, and the AI-supported orchestration mechanism to further reduce complexity within CBPM environments. Besides, the creation of a reference architecture of API-based CBPM systems might offer a workable route map to companies aiming at modernizing their processes. To sum up, effective API integration is not only a technology-focused activity but a strategic initiative of those organizations that adopt adaptive case-based digital reconstruction. According to the theme, consistency and convergence between the domains, the synthesized results are considered to be moderate in terms of confidence. Nevertheless, some conclusions lack power due to variation in the reporting of the methods within the studies that were included.

## References

[1] C. Hudon *et al.*, "Case management programs for improving integrated care for frequent users of healthcare services: an implementation analysis," *International Journal of Integrated Care,* vol. 22, no. 1, p. 11, 2022.

[2] N. Melville and R. Kohli, "Roadblocks to implementing modern digital infrastructure: Exploratory study of API deployment in large organizations," 2021.

[3] D. K. D. Pal, V. Saini, and A. Aakula, "API-led integration for improved healthcare interoperability."

[4] S. Chitta, A. K. R. Sadhu, L. Gudala, and S. G. Reddy, "Unlocking Health Data: API-Driven Solutions for Interoperability Challenges," *Journal of Informatics Education and Research,* vol. 2, p. 45, 2022.

[5] A. G. Adeleke, T. O. Sanyaolu, C. P. Efunniyi, L. A. Akwawa, and C. F. Azubuko, "API integration in FinTech: Challenges and best practices," *International Journal of Financial Technology,* 2024.

[6] M. Zachariadis and P. Ozcan, "The API economy and digital transformation in financial services: The case of open banking," 2017.

[7] A. Chiș, "A modeling method for model-driven API management," *Complex Systems Informatics and Modeling Quarterly,* no. 25, pp. 1-18, 2020.

[8] P. Dullabh, L. Hovey, K. Heaney-Huls, N. Rajendran, A. Wright, and D. F. Sittig, "Application programming interfaces in health care: findings from a current-state sociotechnical assessment," *Applied clinical informatics,* vol. 11, no. 01, pp. 059-069, 2020.

[9] A. Wairagade, "Role of Middleware, Integration Platforms, and API Solutions in Driving Digital Transformation for Enterprises," *Journal of Science & Technology,* vol. 2, no. 1, pp. 387-403, 2021.

[10] W. J. Bakshi, S. Aasim, M. A. Butt, and M. H. Qadri, "Ontology-Driven Solutions for Resolving Semantic Conflicts in Data Integration."

[11] W. J. Gordon and R. S. Rudin, "Why APIs? Anticipated value, barriers, and opportunities for standards-based application programming interfaces in healthcare: perspectives of US thought leaders," *JAMIA open,* vol. 5, no. 2, p. ooac023, 2022.

[12] B. Anthony, "Enabling Seamless Interoperability of Digital Systems in Smart Cities Using API: A Systematic Literature Review," *Journal of Urban Technology,* vol. 31, no. 4-5, pp. 123-156, 2024.

[13] M. Turilli, V. Balasubramanian, A. Merzky, I. Paraskevakos, and S. Jha, "Middleware building blocks for workflow systems," *Computing in Science & Engineering,* vol. 21, no. 4, pp. 62-75, 2019.

[14] B. Suzic, "User-centered security management of API-based data integration workflows," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, 2016: IEEE, pp. 1233-1238.

[15] O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber, "Caterpillar: A Blockchain-Based Business Process Management System," *BPM (Demos),* vol. 172, pp. 1-5, 2017.
[16] K. Muthusamy, "Event-Driven Data Engineering in Microservices Architectures," *International Journal of Emerging Trends in Computer Science and Information Technology,* vol. 6, no. 1, pp. 36-43, 2025.
[17] M. Cate, "The Role of APIs in Driving Innovation and Efficiency in Open Banking," 2025.
[18] B. Perugu, V. Wadhwa, J. Kim, J. Cai, A. Shin, and A. Gupta, "Pragmatic approaches to interoperability–surmounting barriers to healthcare data and information across organizations and political boundaries," *Telehealth and Medicine Today,* vol. 8, no. 4, 2023.
[19] C. Centofanti, W. Tiberti, A. Marotta, F. Graziosi, and D. Cassioli, "Latency-aware kubernetes scheduling for microservices orchestration at the edge," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, 2023: IEEE, pp. 426-431.

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| **TITLE** | | | |
| Title | 1 | Identify the report as a systematic review. | 1 |
| **ABSTRACT** | | | |
| Abstract | 2 | See the PRISMA 2020 for Abstracts checklist. | 4-24 |
| **INTRODUCTION** | | | |
| Rationale | 3 | Describe the rationale for the review in the context of existing knowledge. | 25-117 |
| Objectives | 4 | Provide an explicit statement of the objective(s) or question(s) the review addresses. | 103-117 |

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| **METHODS** | | | |
| Eligibility criteria | 5 | Specify the inclusion and exclusion criteria for the review and how studies were grouped for the syntheses. | 142-171 |
| Information sources | 6 | Specify all databases, registers, websites, organisations, reference lists and other sources searched or consulted to identify studies. Specify the date when each source was last searched or consulted. | 174-176 |
| Search strategy | 7 | Present the full search strategies for all databases, registers and websites, including any filters and limits used. | 172-195 |
| Selection process | 8 | Specify the methods used to decide whether a study met the inclusion criteria of the review, including how many reviewers screened each record and each report retrieved, whether they worked independently, and if applicable, details of automation tools used in the process. | 102-106 |
| Data collection process | 9 | Specify the methods used to collect data from reports, including how many reviewers collected data from each report, whether they worked independently, any processes for obtaining or confirming data from study investigators, and if applicable, details of automation tools used in the process. | 199-215 |

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| Data items | 10a | List and define all outcomes for which data were sought. Specify whether all results that were compatible with each outcome domain in each study were sought (e.g. for all measures, time points, analyses), and if not, the methods used to decide which results to collect. | 215-219 |
| | 10b | List and define all other variables for which data were sought (e.g. participant and intervention characteristics, funding sources). Describe any assumptions made about any missing or unclear information. | -- |
| Study risk of bias assessment | 11 | Specify the methods used to assess risk of bias in the included studies, including details of the tool(s) used, how many reviewers assessed each study and whether they worked independently, and if applicable, details of automation tools used in the process. | 264-288 |
| Effect measures | 12 | Specify for each outcome the effect measure(s) (e.g. risk ratio, mean difference) used in the synthesis or presentation of results. | -- |
| Synthesis methods | 13a | Describe the processes used to decide which studies were eligible for each synthesis (e.g. tabulating the study intervention characteristics and comparing against the planned groups for each synthesis (item #5)). | -- |

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| | 13b | Describe any methods required to prepare the data for presentation or synthesis, such as handling of missing summary statistics, or data conversions. | -- |
| | 13c | Describe any methods used to tabulate or visually display results of individual studies and syntheses. | -- |
| | 13d | Describe any methods used to synthesize results and provide a rationale for the choice(s). If meta-analysis was performed, describe the model(s), method(s) to identify the presence and extent of statistical heterogeneity, and software package(s) used. | -- |
| | 13e | Describe any methods used to explore possible causes of heterogeneity among study results (e.g. subgroup analysis, meta-regression). | -- |
| | 13f | Describe any sensitivity analyses conducted to assess robustness of the synthesized results. | -- |
| Reporting bias assessment | 14 | Describe any methods used to assess risk of bias due to missing results in a synthesis (arising from reporting biases). | 264-288 |
| Certainty assessment | 15 | Describe any methods used to assess certainty (or confidence) in the body of evidence for an outcome. | -- |

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| **RESULTS** | | | |
| Study selection | 16a | Describe the results of the search and selection process, from the number of records identified in the search to the number of studies included in the review, ideally using a flow diagram. | Fig 1 |
| | 16b | Cite studies that might appear to meet the inclusion criteria, but which were excluded, and explain why they were excluded. | Fig 01 |
| Study characteristics | 17 | Cite each included study and present its characteristics. | Fig 1 |
| Risk of bias in studies | 18 | Present assessments of risk of bias for each included study. | 264-288 |
| Results of individual studies | 19 | For all outcomes, present, for each study: (a) summary statistics for each group (where appropriate) and (b) an effect estimate and its precision (e.g. confidence/credible interval), ideally using structured tables or plots. | Table 1 |
| Results of syntheses | 20a | For each synthesis, briefly summarise the characteristics and risk of bias among contributing studies. | -- |
| | 20b | Present results of all statistical syntheses conducted. If meta-analysis was done, present for each the summary estimate and its precision (e.g. confidence/credible interval) and measures of statistical heterogeneity. If comparing groups, | -- |

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| | | describe the direction of the effect. | |
| | 20c | Present results of all investigations of possible causes of heterogeneity among study results. | -- |
| | 20d | Present results of all sensitivity analyses conducted to assess the robustness of the synthesized results. | --- |
| Reporting biases | 21 | Present assessments of risk of bias due to missing results (arising from reporting biases) for each synthesis assessed. | 306-309 |
| Certainty of evidence | 22 | Present assessments of certainty (or confidence) in the body of evidence for each outcome assessed. | 455-458 |
| **DISCUSSION** | | | |
| Discussion | 23a | Provide a general interpretation of the results in the context of other evidence. | 314-402 |
| | 23b | Discuss any limitations of the evidence included in the review. | 301-309 |
| | 23c | Discuss any limitations of the review processes used. | 292-300 |
| | 23d | Discuss implications of the results for practice, policy, and future research. | 404-429 |
| **OTHER INFORMATION** | | | |
| Registration and protocol | 24a | Provide registration information for the review, including register name and registration number, or state that the review was not registered. | |

| Section and Topic | Item # | Checklist item | Location where item is reported |
|---|---|---|---|
| | 24b | Indicate where the review protocol can be accessed, or state that a protocol was not prepared. | |
| | 24c | Describe and explain any amendments to information provided at registration or in the lprotocol. | |
| Support | 25 | Describe sources of financial or non-financial support for the review, and the role of the funders or sponsors in the review. | -- |
| Competing interests | 26 | Declare any competing interests of review authors. | 460-461 |
| Availability of data, code and other materials | 27 | Report which of the following are publicly available and where they can be found: template data collection forms; data extracted from included studies; data used for all analyses; analytic code; any other materials used in the review. | 463-464 |

*From:* Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ 2021;372:n71. doi: 10.1136/bmj.n71. This work is