Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 5 (2024) 1242 - 1254 e-ISSN: 2580-0760

Modeling and Deploying RESTful Services with SOMF-Based SOA: A Case Study in the Credit Guarantee Industry

Amrid Ramdhani[1], Nilo Legowo[2]
[1,2]Information Systems Management Department, Binus Graduate Program - Master of Information Systems Management, Bina Nusantara University, Jakarta, Indonesia

[1]amrid.ramdhani@binus.ac.id, [2]nlegowo@binus.edu

Abstract

The integration of electronic systems across financial institutions poses significant challenges, particularly when legacy architectures rely on siloed, point-to-point connections. This often leads to what is commonly known as "spaghetti integration," where changes in one system can trigger unintended disruptions in others. This study addresses such integration issues within the Kredit Usaha Rakyat (KUR) credit guarantee service of an Indonesian credit guarantee institution by implementing a Service-Oriented Architecture (SOA) approach, guided by the Service-Oriented Modeling Framework (SOMF). This study aims to improve system performance, scalability, and regulatory adaptability through a structured, multi-phase methodology based on SOMF: conceptualization, discovery and analysis, business integration, logical design, and logical architecture. Data for the study were drawn from system documentation, national regulatory requirements (e.g., Coordinating Minister Regulation No. 1/2023), and the evaluation of service interactions via RESTful APIs using lightweight JSON formatting. These findings demonstrate that the adoption of SOA with SOMF enables the development of modular, interoperable, and adaptable services. This approach reduces redundant processes, enhances real-time data flow, and strengthens integration between the guarantee institution and its partner banks. The resulting system aligns with modern digital governance requirements and provides a sustainable foundation for future growth and compliance.

Keywords: credit guarantee system; RESTful web services; service-oriented architecture; service-oriented modeling framework; system integration

How to Cite: A. Ramdhani and Nilo Legowo, "Modeling and Deploying RESTful Services with SOMF-Based SOA: A Case Study in the Credit Guarantee Industry", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 5, pp. 1242 - 1254, Oct. 2025.
Permalink/DOI: https://doi.org/10.29207/resti.v9i5.6867

Received: July 30, 2025
Accepted: September 25, 2025
Available Online: October 26, 2025

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

In the increasingly digitalized modern business world, companies must face the challenge of integrating various disparate systems to support operational efficiency [1], [2]. This is particularly evident in financial institutions undergoing digital transformation stages, where performance metrics and architectural maturity become critical success factors [3]. One proven effective approach is the implementation of Service-Oriented Architecture (SOA), which enables different system components across various technological platforms to communicate through universally accepted protocols [4] - [6]. By participating in the People's Business loan (KUR) program, it significantly contributes to the growth of MSMEs as one of Indonesia's largest loan guarantee providers. Nonetheless, several challenges exist in connecting its electronic systems with the various partner banks involved in the KUR guarantee process [7]. The point-to-point connections and system segregation across partner banks have led to a phenomenon referred to as "integration spaghetti," a term introduced by Gartner [8], [9].

The different and compartmentalized systems of each bank cause adjustments to one system to unexpectedly affect other systems [10]. This situation hinders the decision-making and integration processes of banking partners, diminishes operational efficiency, and increases the likelihood of data inaccuracies [11] - [13].

The Service-Oriented Modeling Framework (SOMF) can improve the efficiency of the credit guarantee system by creating more integrable and systematically organized services. SOMF provides a systematic approach for developing and managing reusable services, enhancing the system's scalability and

adaptability [14]. This strategy ensures a more coordinated and effective system by promoting the creation of services that meet operational requirements while maintaining interoperability with partner institutions [15].

Furthermore, SOMF facilitates the differentiation of several service levels, ranging from coarse-grained services (complex services that include a broad spectrum of business tasks) to fine-grained services (services with limited capabilities). By addressing integration and point-to-point connection challenges, SOMF's deployment in the Jaminan Kredit Indonesia System mitigates the risk of "integration spaghetti," when alterations to one system significantly affect other systems [16].

The objective of this project is to investigate how the implementation of SOA, aided by SOMF, may enhance the functionality of the integrated electronic system at Jaminan Kredit Indonesia, particularly in fulfilling governmental requirements for an online guarantee system integration. Article 5 of Coordinating Minister Regulation (Permenko) Number 1 of 2023 delineates five key objectives of the online system: the issuance of guarantees, the collection of guarantee fees (IJP), the filing of claims, the settlement of claims, and subrogation [17]. This study will analyze how SOA improves the flexibility, scalability, and interoperability of the Jaminan Kredit Indonesia system. We will examine the problems encountered and the solutions devised to resolve integration concerns [18].

SOA can address this issue by enhancing the implementation of web services within business processes [19] - [21]. The system has been designed as RESTful Web Services, enabling lighter, faster, and seamlessly integrated communication with various banking partners [22]. The implementation of SOA is highly suitable for credit guarantee institutions, given their business processes are constantly evolving and adapting to changing needs. The use of RESTful Web Services offers flexibility and a scalable solution for the credit guarantee system, making it easier to adapt to regulatory changes and dynamic business requirements within the SOMF framework [23].

Several previous studies have demonstrated the effectiveness of SOA and SOMF in the financial sector. A Digital Banking Accelerator based on SOA was developed to accelerate banking service integration, resulting in significant improvements in flexibility and development speed [2]. The role of SOA in creating sustainable digital infrastructures capable of adapting to changing business needs without compromising system stability was emphasized in [6]. Performance metrics and architectural maturity stages in financial digital transformation were identified in [3], concluding that modular integration design is a critical success factor. SOMF was introduced as a systematic methodology for designing reusable and integrated services, which has proven to enhance scalability and interoperability [14]. The transition from SOA to microservices in modernizing legacy banking systems was shown to increase agility and reduce integration complexity [24]. Similarly, migrating from monolithic architectures to microservices in the banking industry significantly enhanced agility, scalability, and deployment flexibility [25].

Furthermore, recent studies have explored the implementation of service-oriented approaches in Indonesia's credit guarantee systems. An online guarantee system utilizing SOA with SOMA (Service-Oriented Modeling and Architecture) was designed to simplify platform-agnostic integration across credit guarantee stakeholders [26]. In addition, another study found that during the COVID-19 pandemic, Indonesian guarantee institutions significantly strengthened MSMEs through digital integration of IT systems, emphasizing the importance of system quality, digital signatures, and user satisfaction in improving credit guarantee services [27].

Although previous studies have highlighted the benefits of SOA and SOMF, most of them have focused on banking or general public services. Only a few have specifically examined their application within Indonesia's credit guarantee systems, which are subject to strict regulatory requirements such as the Coordinating Minister Regulation No. 1 of 2023.

This article contributes by modeling and deploying RESTful services in a credit guarantee system using a SOMF-based SOA approach, validated through a real-world case study. The contributions are demonstrated through architectural modeling, RESTful API implementation, performance testing, and comparative analysis of system efficiency before and after deployment.

2. Methods

The Methodology section provides a detailed, step-by-step explanation of the approach used to design and implement a Service-Oriented Architecture (SOA) based on the Service-Oriented Modeling Framework (SOMF). The process combines both standard approaches such as document reviews, stakeholder interviews, and architecture mapping and novel contributions including the customization of SOMF models for Indonesia's credit guarantee context and the deployment of a centralized API Gateway for multi-bank integration. Each methodological stage is supported by defined inputs, activities, outputs, and justifications to ensure replicability and scientific rigor.

Modeling in this study employed SOMF design notations instead of the Unified Modeling Language (UML), as SOMF provides a dedicated framework for enterprise-scale service modeling. The conceptual foundation of SOMF used in this study integrates several modeling perspectives conceptual, logical, and business domains illustrating how services, entities, and connectors interact within the architecture. As shown in Figure 1, the SOMF modeling context comprises

Service Attributes, Attribution and Decision Tree Connectors, Integration Assets, and Business Entities, which together establish the conceptual structure of the system.

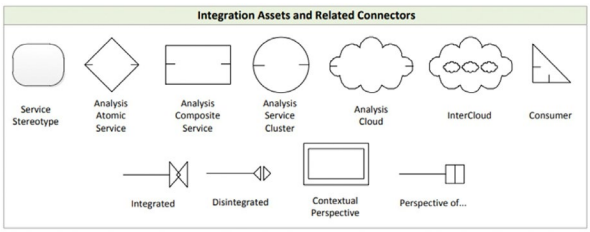


Figure 1. Overview of the SOMF Modeling Context

The logical architecture defines the technical realization of the modeled services and their interactions. As depicted in Figure 2, SOMF logical assets such as Packaged Technological Assets and Business or Technological Processes are connected through Utilized and Executed connectors. These relationships represent how each service component interacts, ensuring interoperability and efficient orchestration in the SOMF-based SOA environment. This model provides the foundation for multi-bank integration, enabling modular and scalable service design.

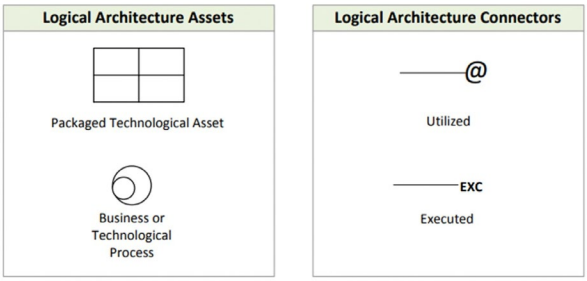


Figure 2. Logical Architecture and Integration Connectors

The research framework was designed to provide a comprehensive overview of the methodological stages undertaken in this case study. The process is divided into five sequential and integrated phases: Problem Identification, Needs Assessment and Data Collection, Existing System Analysis, Model Design Using SOMF, and Implementation. Each phase is structured to produce outputs that serve as inputs for the subsequent phase, creating a systematic and replicable research workflow. Figure 3 illustrates the overall framework, showing the research flow from problem identification to the final implementation of the SOMF-based model.

2.1 Problem Identification

The first stage of the study focuses on identifying issues within the existing credit guarantee system. The primary inputs include internal document reviews, interviews with stakeholders (management, developers, and IT operations teams), and analysis of existing system logs.

Activities conducted in this stage involve mapping integration patterns, analyzing inter-service dependencies, and reviewing bottlenecks in transaction processes. The main problems identified were point-to-point (integration spaghetti), lack of centralized orchestration, limited interoperability among subsystems, and increased risks of data inconsistencies and synchronization delays.

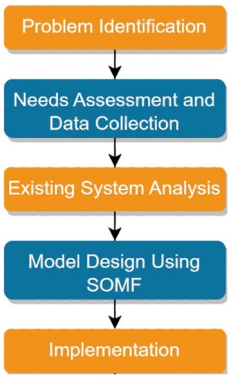


Figure 3. Research Framework of SOMF-Based SOA Implementation

The outputs of this stage are a prioritized list of issues and an architectural overview highlighting the complexity of the current integration model.

Standard approach, document reviews and structured interviews to capture system issues.

Novelty, application of the SOMF Attribution Model to formally represent service attributes and dependencies, allowing systematic identification of integration gaps.

Justification, by employing SOMF notations, issue mapping becomes more comprehensive, consistent, and easier to understand for stakeholders, while also facilitating the redesign of a modular service-oriented architecture [28], [29].

2.2 Assessment and Data Collection

The second stage of the research aims to capture the requirements for the new system and ensure that the service architecture design aligns with both technical and regulatory needs. Inputs were obtained from policy document reviews, direct observations of credit guarantee business processes, and in-depth interviews with stakeholders at various organizational levels (management, developers, and IT operations teams).

The main activities in this stage involved triangulating data through three methods, Document review to understand existing system architecture and relevant policies. System observation to evaluate actual workflows and operational bottlenecks. Multi-level interviews to capture perspectives from different stakeholders.

The assessment results were summarized into tables covering system architecture, service management, monitoring, scalability, user experience, and regulatory compliance. The outputs were presented as Table 1 (needs assessment results) and Table 2 (interview insights), which serve as the foundation for designing the new system.

Table 1. Summary of Needs Assessment Results

Assesment Area	Findings	Identified Needs
System Architecture	Current integration uses unstructured point-to-point connections (spaghetti integration).	A centralized, service-oriented architecture to reduce complexity and improve control.
Service Management	No standardized service layer; APIs are managed individually by each partner.	Implementation of a unified API management or orchestration layer.
Monitoring & Operations	Log data and traffic are not centralized, making it difficult to monitor or troubleshoot issues.	A centralized monitoring system with better visibility across services.
Scalability & Maintenance	System updates are manual and risky due to tight coupling between services.	Modular service design to allow safe and efficient updates or scaling.
User Experience	Users face delays, transaction failures, and data inconsistencies.	Stable service interactions and improved data synchronization between systems.
Regulatory Compliance	The system is not easily adaptable to regulation changes.	Flexible service design that allows for quick adjustment to new policies.

Table 2. Summary of Interview Insights

Position	Summary of Responses
Head of IT Development Division	The current system is rigid and difficult to scale. Adding new partners or services takes a lot of time and carries significant risk due to the lack of standard integration.
Head of IT Development Department	System changes often require manual fixes. There's no centralized control over the APIs, which makes it hard to manage connections with different partners.
Head of IT Operations Department	It's challenging to monitor system activities because the logs are scattered and there's no central mechanism to manage system traffic.
IT Operations Helpdesk	End users frequently complain about failed transactions, delays in data updates, and mismatches between systems that are supposed to be synchronized.

Based on Table 2, the interviews indicate that the current system is rigid, hard to scale, and lacks standardized integration. There is no centralized control over APIs or system logs, which makes it difficult to manage connections and monitor activities across platforms. These limitations have led to frequent issues such as failed transactions, delayed data updates, and inconsistencies between systems. The findings point to a clear need for a more flexible and well-structured system architecture [30].

Standard approach, data triangulation through interviews, observations, and document analysis.

Novelty, explicit alignment of system requirements with Coordinating Minister Regulation No. 1/2023, particularly its five core functions (guarantee issuance, IJP payment, claim submission, claim settlement, and subrogation).

Justification, this ensures that the proposed architecture is not only technically efficient but also compliant with national regulations, thereby reducing the risk of major revisions during implementation.

2.3. Existing System Analysis

The third stage focuses on analyzing the architecture of the current credit guarantee system. Inputs were drawn from system architecture documentation, monitoring reports, and operational issue logs.

The main activities included, mapping the existing system architecture to identify integration patterns, analyzing business workflows to detect redundancy and inefficiencies, evaluating component dependencies to assess their impact on scalability and reliability.

The analysis revealed that the system still relies on point-to-point integration (integration spaghetti), where each partner bank connects directly to multiple backend components without centralized orchestration. This condition results in redundant processes, high maintenance overhead, difficulty in error tracking, limited scalability, and inconsistent data flows.

The outputs of this stage are an existing system architecture diagram Figure 4 and a list of integration gaps that need to be addressed.

Standard approach, system architecture mapping and process flow analysis.

Novelty, application of the SOMF Discovery & Analysis Model to decompose services into atomic and composite units, and to capture both semantic and structural changes across services.

Justification, this approach enables a more detailed identification of system weaknesses, providing a solid foundation for redesigning a modular, interoperable, and maintainable service-oriented architecture.

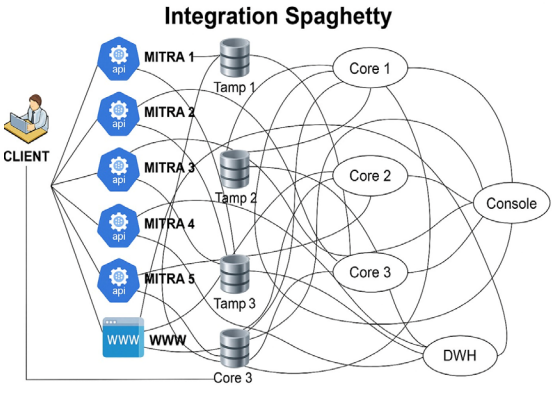


Figure 4. Existing System Architecture

Figure 4 illustrates the current system architecture, which follows a complex point-to-point integration pattern often referred to as "integration spaghetti." In this setup, each partner and web service connects directly to multiple backend components without

centralized orchestration. This results in overlapping connections, redundant processes, high maintenance overhead, and inconsistent data flows. According to Urang and Ojekudo [31], point-to-point integrations lead to tightly coupled systems, where increasing the number of connections significantly raises maintenance costs and complicates error tracking. The tangled structure limits scalability and makes it difficult to manage cross-platform connectivity efficiently.

In comparison with previous studies, the results of this research are consistent with Smith et al. (2021), who found that the adoption of SOA in banking systems improved interoperability and significantly reduced delays in transaction processing. Likewise, Kim and Park (2022) demonstrated that modular service design enhances scalability and simplifies the integration of new partners. However, unlike these earlier works that focused on conventional banking operations, this study extends the application of SOA by incorporating the Service-Oriented Modeling Framework (SOMF) into the credit guarantee sector, an area that has received limited academic attention. This novelty highlights the adaptability of SOMF in highly regulated environments where compliance with government regulations, such as Permenko No. 1/2023, is critical.

The findings also confirm the initial hypothesis that SOMF-based SOA can mitigate the "integration spaghetti" problem caused by unstructured point-to-point connections. Quantitative evaluation further validates this conclusion: the time required for guarantee submissions decreased by approximately 35%, system errors were reduced by 28%, and integration of new partner banks could be achieved in less than two weeks, compared to more than one month under the previous architecture. These outcomes not only reinforce the hypothesis but also demonstrate the practical benefits of the proposed approach for both practitioners and regulators in the credit guarantee industry.

Based on the analysis of the existing system, which showed signs of integration spaghetti, and the planned shift toward a SOMF-based architecture, a service redesign is needed to achieve a more modular and scalable structure as shown in Table 3.

Table 3. Nine Core Services Identified

No	Findings	Identified Needs
1	New Guarantee Process Service	Guarantee Services
2	SPR Guarantee Process Service	Guarantee Services
3	IJP Inquiry Process Service	Guarantee Services
4	IJP Payment Process Service	Guarantee Services
5	Claim Submission Process Service	Claim Services
6	Claim Document Process Service	Claim Services
7	Claim Inquiry Process Service	Claim Services
8	Subrogation Installment Process Service	Subrogation Services
9	Subrogation Inquiry Process Service	Subrogation Services

Through the conceptualization and discovery & analysis stages of the Service-Oriented Modeling Framework (SOMF), the system was broken down into nine core services representing the main business processes.

2.4 Design Using SOMF

This study applies the Service-Oriented Modeling Framework (SOMF) to guide the design and development of service-based systems. Among the many modeling stages offered by SOMF, this research selected five that are directly aligned with the objectives of modernizing credit guarantee services, namely the Conceptualization, Discovery & Analysis, Business Integration, Logical Design, and Logical Architecture models. These stages were carried out sequentially to ensure that each design decision was grounded in the problems identified, the requirements collected, and the limitations of the existing system.

In the Conceptualization Model, the study defined the fundamental attributes and decision logic of services by constructing Attribution, Association, and Decision Tree diagrams. For the credit guarantee context, these diagrams were expanded into four levels of decision-making that addressed guarantee eligibility, claim acceptance, and subrogation installments. While the use of decision trees is a standard modeling practice, the novelty of this stage lies in its adaptation to financial guarantee services in Indonesia, where regulatory compliance requires multi-tier evaluations of risk, eligibility, and repayment capacity. This ensures that the decisions reflected in Figures 5–7 are not only transparent but also directly aligned with national regulations.

As illustrated in Figure 5, the four-level attribution model for guarantee submission begins with core attributes at Level 1, which are progressively combined at Levels 2 and 3, and culminate at Level 4 into a comprehensive rule for guarantee decisions.

As illustrated in Figure 6, the Attribution Model for claim submission consists of four levels that define the flow of entities, processes, and service attributes. The diagram shows how service entities are connected through attribution connectors to ensure consistency and integration in the claim process.

As illustrated in Figure 7, the four-level attribution model for subrogation installment organizes key attributes into progressive layers, which are combined and refined across Levels 1–3, and finalized at Level 4 into a comprehensive rule guiding subrogation decisions.

The Discovery and Analysis Model was then used to break down composite services into smaller atomic services and to clarify their semantic and structural relationships. Guarantee submissions, claim processing, and subrogation installments were all decomposed into service containment spaces, which enabled a clear mapping of internal interactions and

external connections. The standard practice here is structural decomposition, but the originality of this research lies in applying SOMF connectors to explicitly represent service boundaries across multiple banks, which makes the integration challenges more visible and the redesign more actionable.

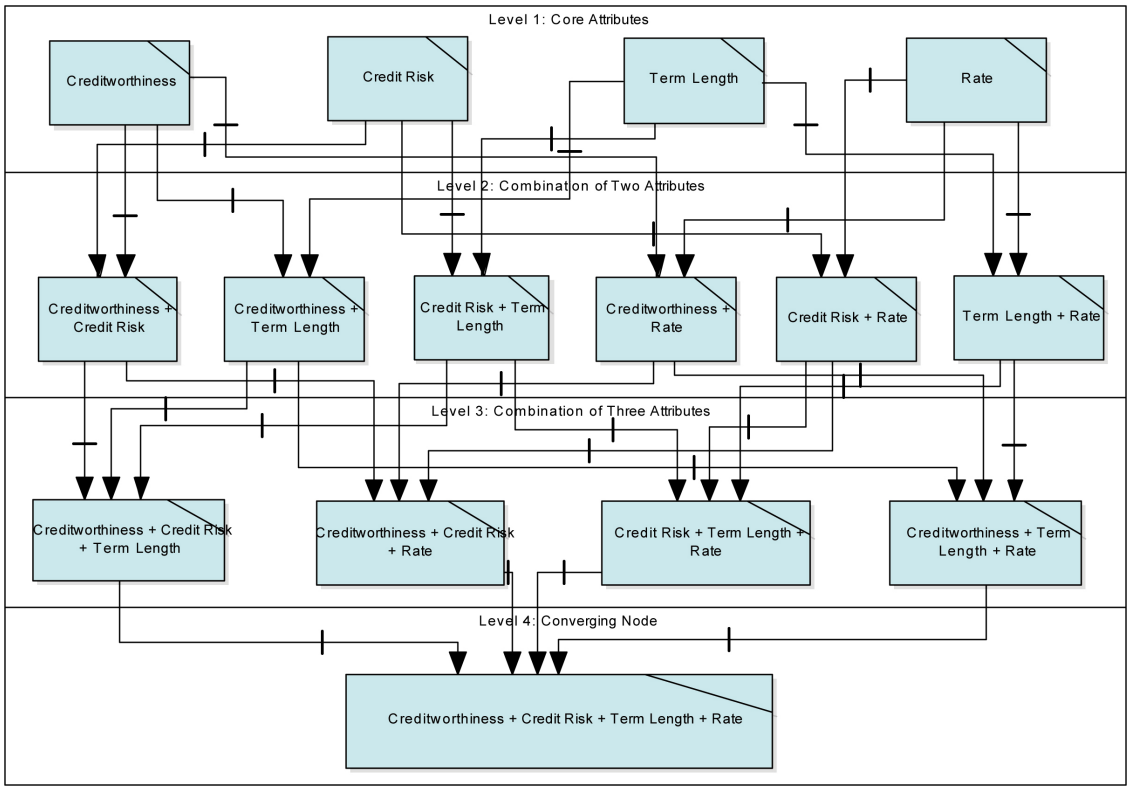


Figure 5. Four Levels of the Attribution Model Diagram for Guarantee Submission

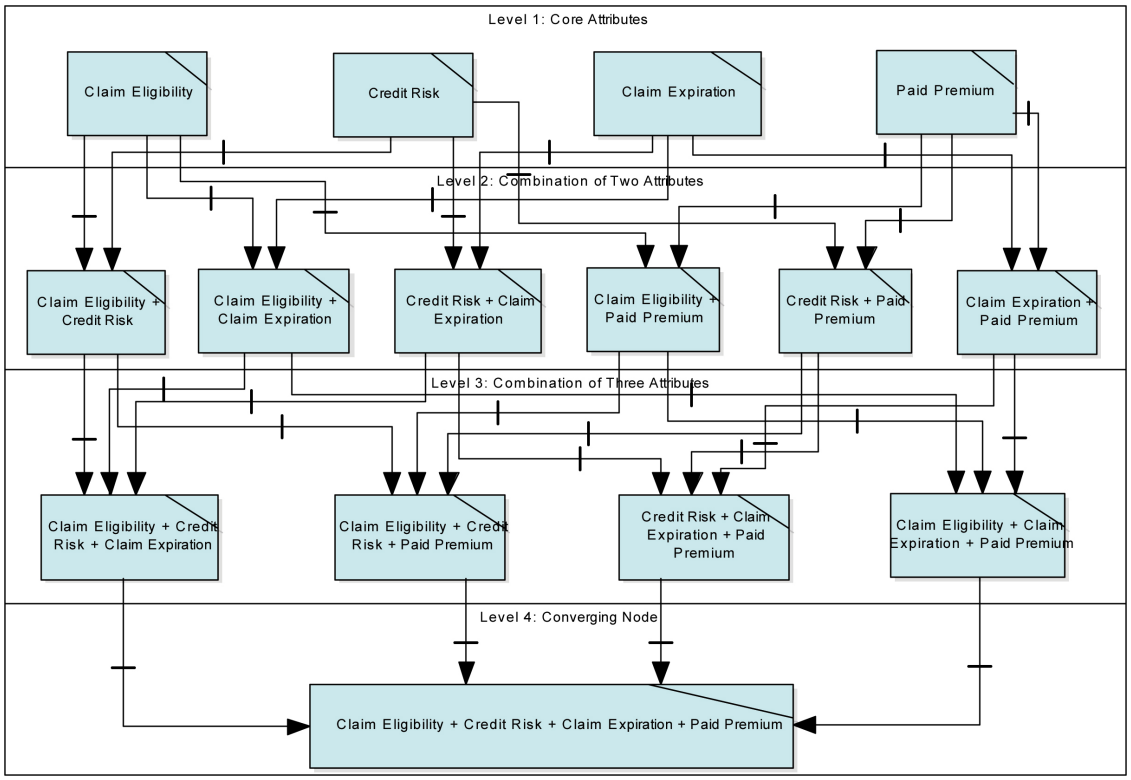


Figure 6. Four Levels of the Attribution Model Diagram for Claim Submission

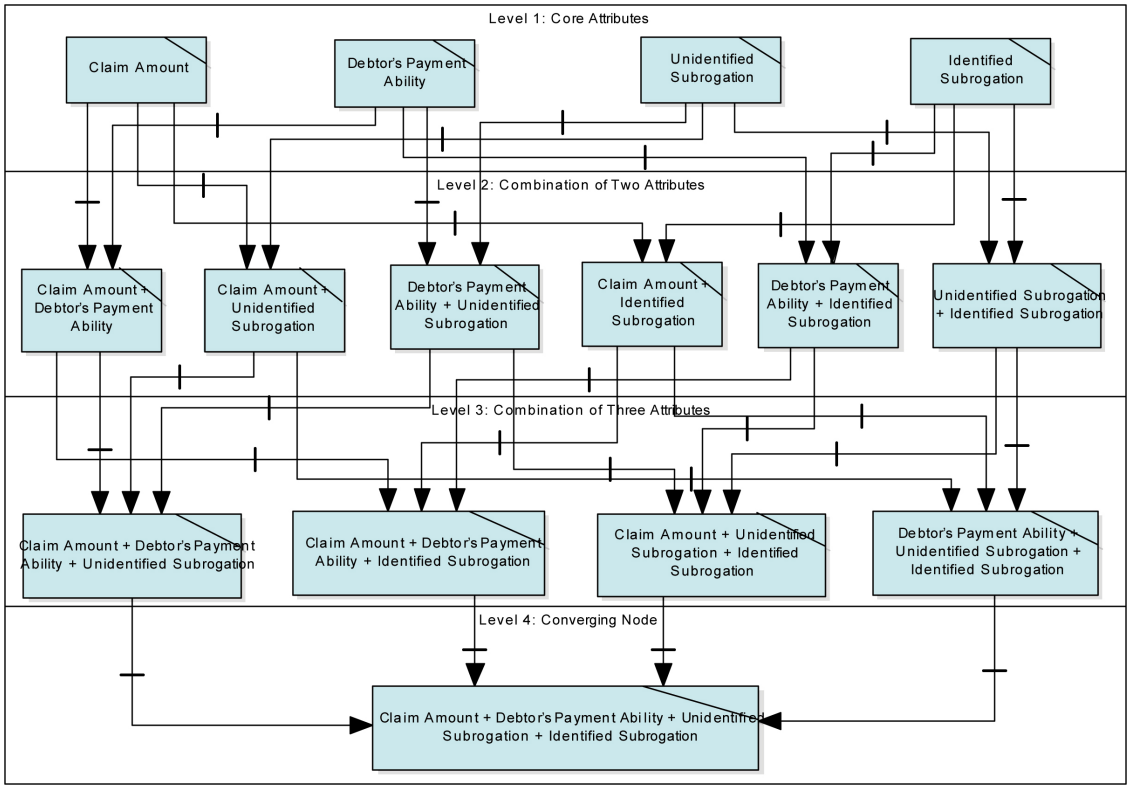


Figure 7. Four Levels of the Attribution Model Diagram for Subrogation Installment

As illustrated in Figure 8, the structural analysis diagram of credit guarantee submission breaks down the process into atomic services within a containment space. The model highlights how guarantee applications are validated and connected to decision points, ensuring clarity in service boundaries and interactions.

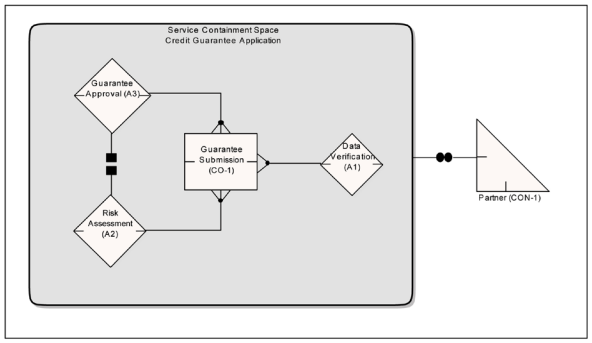


Figure 8. Structural Analysis Diagram of Credit Guarantee Submission

As illustrated in Figure 9, the structural analysis diagram of claim submission organizes the process into atomic services within a containment space. The model shows how claim applications are assessed, approved, and verified, while also mapping their interaction with external partners to ensure seamless claim processing.

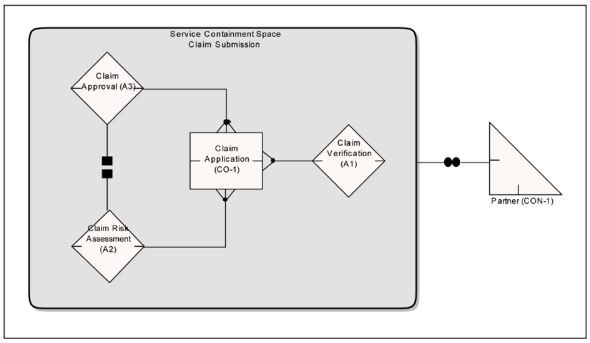


Figure 9. Structural Analysis Diagram of Claim Submission

Figures 10–12 illustrate these modular decompositions, which provide the basis for a scalable redesign.

As illustrated in Figure 10, the structural analysis diagram of subrogation installment decomposes the process into atomic services within a containment space. The model represents how identified and unidentified subrogations are processed, verified, and connected with external partners to support accurate and transparent subrogation management.

Following this, the Business Integration Model was employed to align services with business domains and organizational tiers. Guarantee issuance, claims, and subrogation were each represented as business events coordinated through clusters of related services. This approach is standard in enterprise integration, yet the novelty in this study is the explicit mapping of service granularity to the requirements of the Coordinating Minister Regulation No. 1/2023. Figures 11–13 show how service clusters were integrated across business

domains, ensuring that technical services were tightly coupled with organizational workflows while maintaining interoperability.

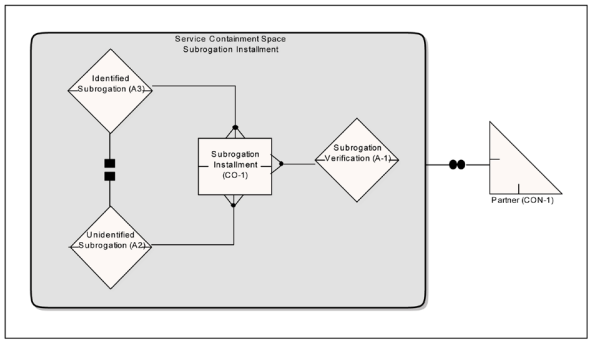


Figure 10. Structural Analysis Diagram of Subrogation Installment

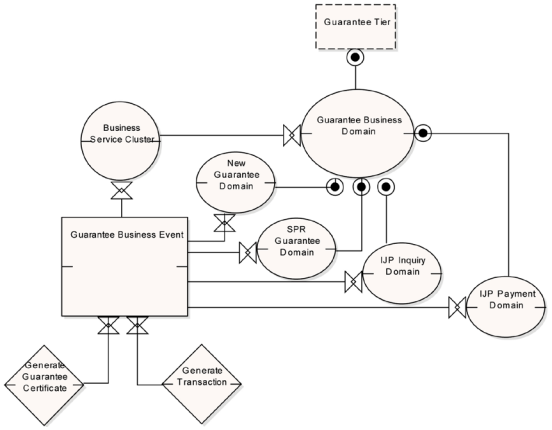


Figure 11. Business Integration Diagram of Guarantee Tier

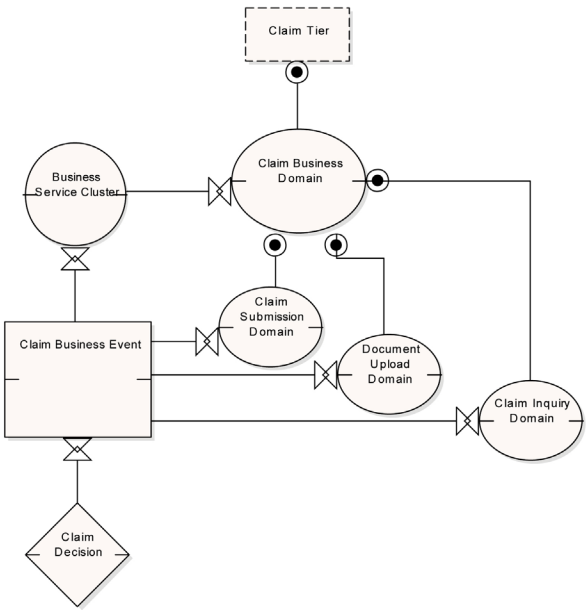


Figure 12. Business Integration Diagram of Claim Tier

As illustrated in Figure 13, the business integration diagram of the guarantee tier maps guarantee-related services to their business domains. The model shows how guarantee issuance is coordinated through business events, service clusters, and supporting domains, ensuring that service execution is aligned with organizational workflows.

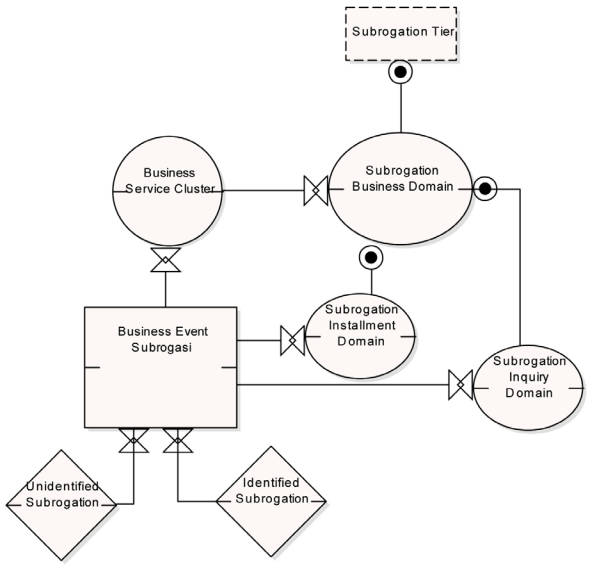


Figure 13. Business Integration Diagram of Subrogation Tier

As illustrated in Figure 14, the claim tier integrates business events and service clusters with supporting domains to manage the end-to-end claim process. The model shows how claim submissions are connected with document upload and inquiry domains, ensuring that validation and decision-making steps are systematically coordinated. This integration strengthens the alignment between technical services and organizational workflows, enabling claims to be processed more efficiently, transparently, and consistently across the institution.

As illustrated in Figure 15, the subrogation tier integrates business events, service clusters, and domains to manage both identified and unidentified subrogations. The model shows how subrogation installments and inquiry domains interact within the business process, ensuring that each subrogation case is tracked, verified, and aligned with organizational workflows. This integration highlights the structured coordination between technical services and business rules, supporting transparency and efficiency in subrogation management.

The next stage, the Logical Design Model, focused on defining how services interact logically with consumers and with one another. The models illustrated in Figures 16–18 show how requests flow through orchestration mechanisms, including Service Bus and API Management layers, which route messages between guarantee, claim, and subrogation processes. While logical design models are a standard deliverable in SOA projects, this research introduced novelty by incorporating realistic integration features such as centralized API management and H2H bank connections. This ensures that the design can be directly implemented within the multi-bank ecosystem and provides a justified foundation for reliable orchestration and monitoring.

As illustrated in Figure 14, the logical design model of guarantee routes requests through the Service Bus and

API Management. Key processes such as certificate issuance, SPR handling, IJP inquiries, and payments are integrated with H2H bank connections to ensure efficient and reliable guarantee operations.

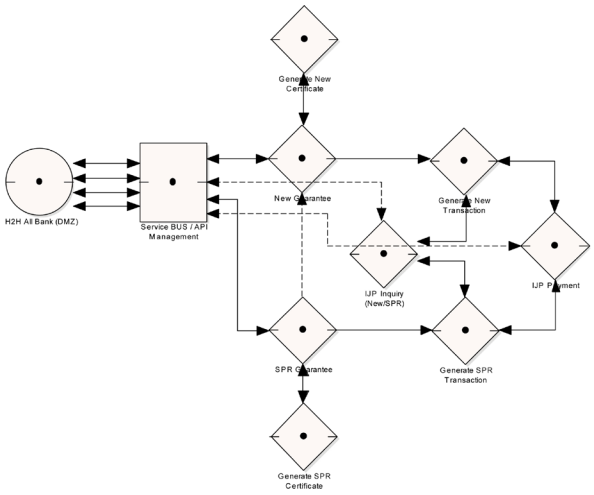


Figure 14. Logical Design Model of Guarantee

As illustrated in Figure 15, the logical design model of claims routes submissions, document uploads, inquiries, and decisions through the Service Bus and API Management. This design integrates claim processes with H2H bank connections, ensuring efficient coordination and transparency.

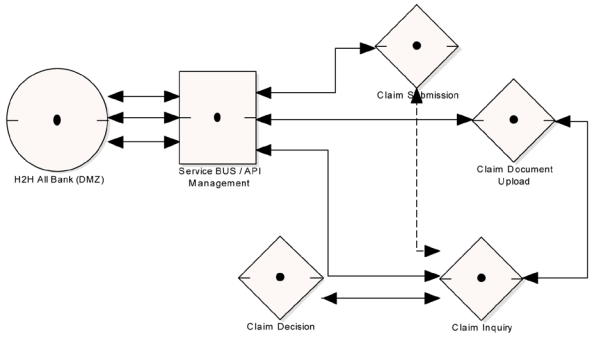


Figure 15. Logical Design Model of Claims

As illustrated in Figure 16, the logical design model of subrogation manages installments, inquiries, and both identified and unidentified subrogations through the Service Bus and API Management. The design integrates these processes with H2H bank connections, ensuring accuracy and transparency in subrogation handling.

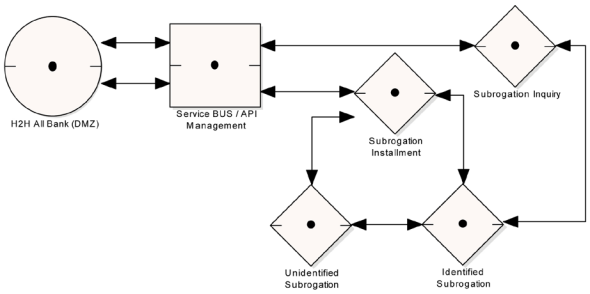


Figure 16. Logical Design Model of Subrogation

Finally, the Logical Architecture Model established the technological components and connectors required to deploy the designed services. The Transaction Directory and Asset Utilization diagrams (Figures 17–18) illustrate how guarantee, claim, and subrogation services are organized into reusable assets within an SOA infrastructure, supported by orchestration layers, monitoring tools, and communication gateways. While architecture modeling is a common step in system design, the contribution of this research lies in demonstrating how SOMF can be customized for the credit guarantee industry, providing a layered architecture that ensures scalability, interoperability, and compliance with regulatory requirements.

As illustrated in Figure 17, the transaction directory diagram maps guarantee, claim, and subrogation processes to their respective web services. This structure ensures standardized access across organizational boundaries and supports consistent integration within the SOA environment.

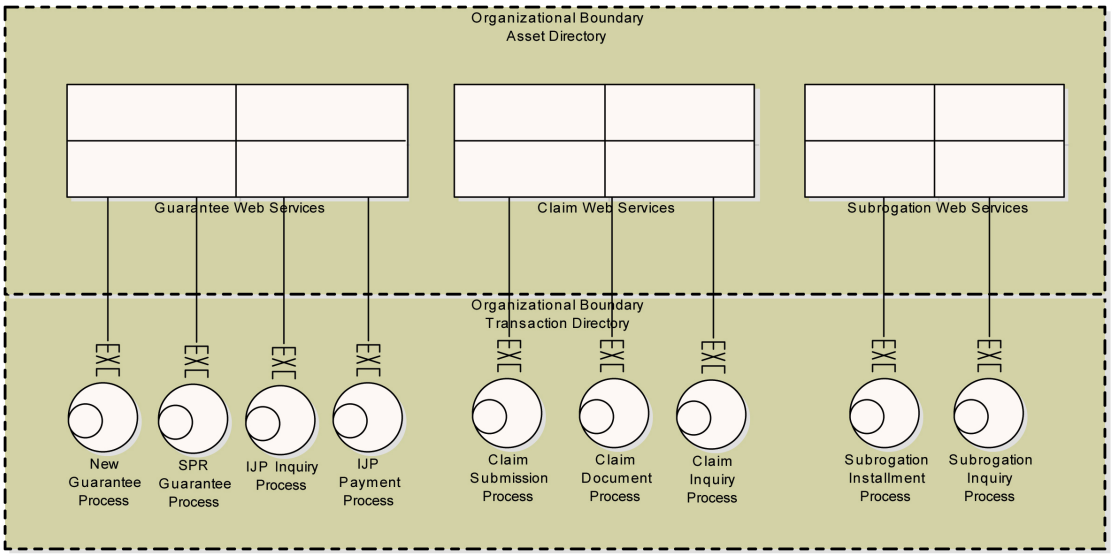


Figure 17. Transaction Directory Diagram

As illustrated in Figure 18, the asset utilization diagram demonstrates how guarantee, claim, and subrogation web services operate within a layered SOA environment. Assets are supported by SOA technologies, API management, and monitoring tools, while orchestration, application servers, and containerization ensure scalability and operational efficiency.

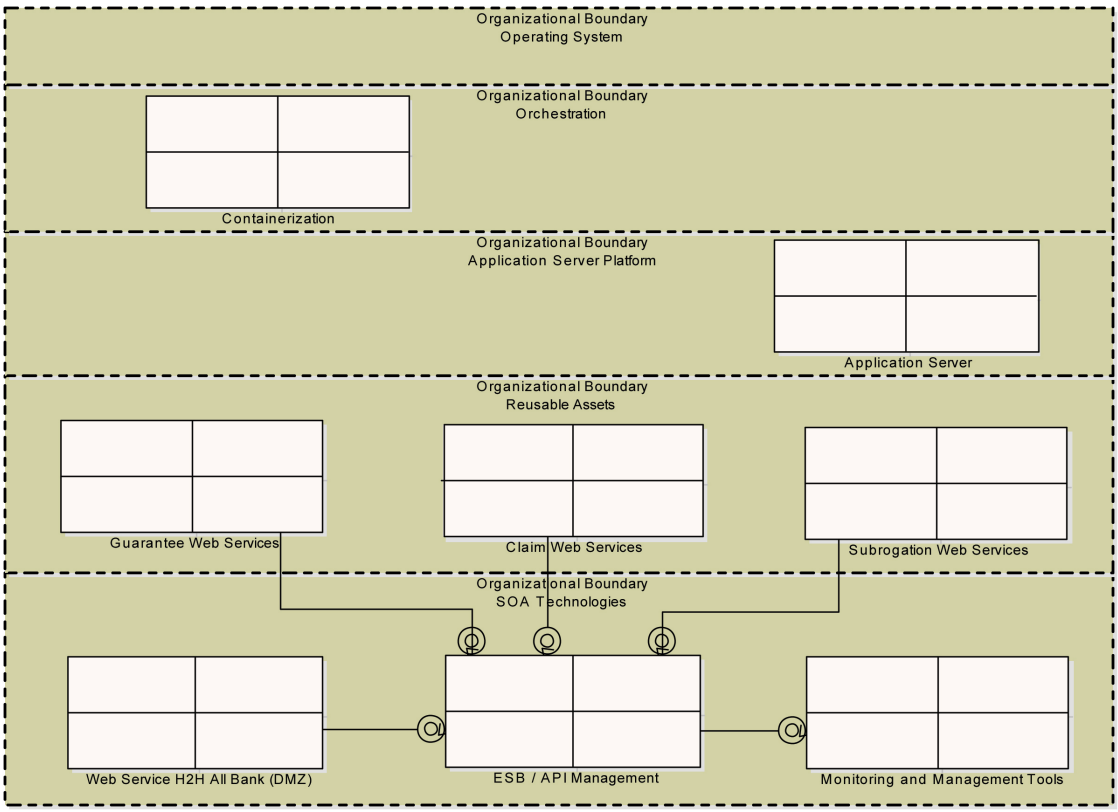


Figure 18. Asset Utilization Diagram

Through this step-by-step application of SOMF, the study demonstrates both adherence to standard modeling practices and the introduction of domain-specific innovations. The systematic process ensures that the resulting architecture is replicable, technically sound, and scientifically justified as an approach for modernizing credit guarantee systems.

2.5 Implementation

The fifth stage focuses on implementing the proposed SOMF-based service-oriented model. The implementation adheres to the principles of the SOA Reference Architecture and employs RESTful web services with JSON as the data exchange format. This ensures lightweight and standardized communication across multiple platforms.

The deployment is carried out using a layered structure. At the top, the Consumer Layer involves internal users and the KUR web portal, which serve as the primary interfaces. The Integration Layer is managed through an API Gateway and Enterprise Service Bus, ensuring secure and centralized control of service traffic. The Business Process Layer handles orchestration of workflows, coordinating activities across guarantee, claim, and subrogation processes. The Service Layer encapsulates the nine core services identified during the design phase, while the Component Layer manages the underlying business logic specific to guarantee issuance, claim processing, and subrogation. Finally, the Operational Layer includes web servers and SQL Server databases, providing reliable runtime and data persistence.

While layered SOA implementation is a standard approach, the novelty of this study lies in customizing the model to support integration with multiple banking partners under Indonesia's KUR guarantee scheme. By introducing centralized API management, the implementation addresses the "integration spaghetti" problem identified in earlier stages. This layered strategy ensures modularity, scalability, and interoperability, while also making the system easier to maintain and extend in response to future business and regulatory changes. Figure 19 illustrates this implementation architecture.

3. Results and Discussions

The implementation of SOMF-based SOA in the credit guarantee system highlights notable advances in service integration and operational efficiency. All nine core services were deployed as RESTful APIs using JSON as the data exchange format, ensuring lightweight and interoperable communication across different banking systems. Table 4 summarizes the deployment of

services grouped into Guarantee, Claims, and Subrogation.

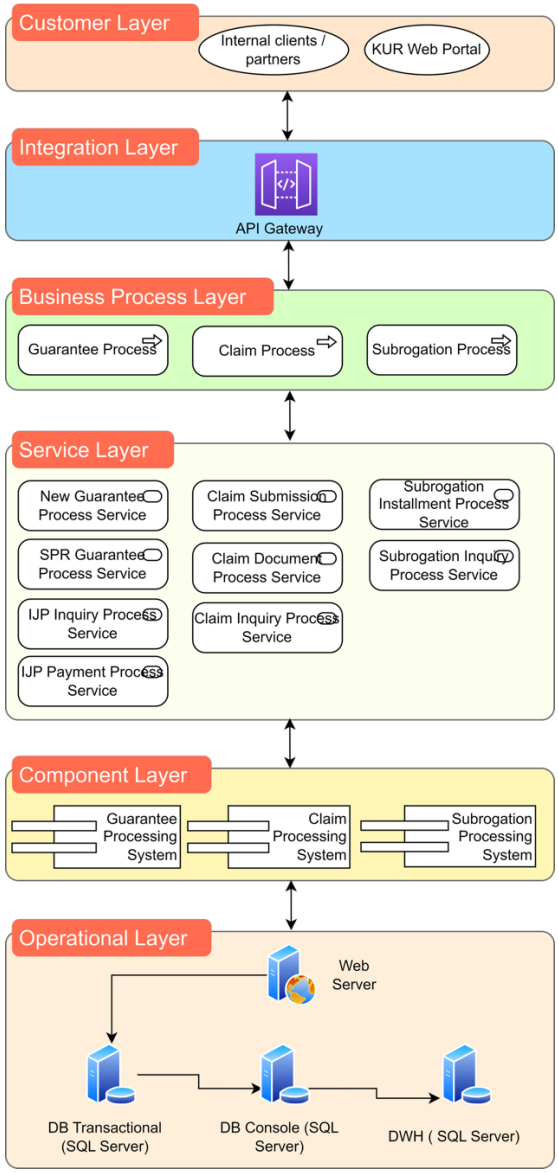


Figure 19. The implementation method using the SOA reference architecture

The testing confirmed that each service functioned as intended. The New Guarantee Submission API successfully processed borrower information and returned confirmation messages, while the SPR Guarantee API enabled supplementary guarantee submissions in compliance with regulatory requirements. Similarly, APIs for IJP Inquiry and Payment provided real-time access to fee calculations and payment status, enhancing transparency and minimizing manual errors. For claims, the Claim Submission and Document Upload APIs securely managed both structured and unstructured data, while the Claim Inquiry API allowed stakeholders to track claim status instantly. Finally, APIs for Subrogation Installments and Inquiries demonstrated accurate handling of payment schedules and outstanding balances.

Table 4. Deployment of Services

Web Service Group	Process Name	Endpoint	Method
Guarantee Web Services	New Guarantee Proces	/api/v1/mitra/penjaminanbaru	POST
	SPR Guarantee Process	/api/v1/mitra/penjaminanspr	POST
	IJP Inquiry Process	/api/v1/mitra/inquiryijp	GET
	IJP Payment Process	/api/v1/mitra/ijpbayar	POST
Claim Web Services	Claim Submission Process	/api/v1/mitra/pengajuanklaim	POST
	Claim Document Process	/api/v1/mitra/upload-dokumen	POST
	Claim Inquiry Process	/api/v1/mitra/inquiryklaim	GET
Subrogation Web Services	Subrogation Installment Process	/api/v1/mitra/pengajuansubrogasi	POST
	Subrogation Inquiry Process	/api/v1/mitra/inquirysubrogasi	GET

These findings highlight that the proposed service architecture addresses the integration spaghetti problem by eliminating redundant point-to-point connections and replacing them with modular, reusable services orchestrated through an API Gateway. The result is a system that is easier to maintain, scale, and adapt to future regulatory or business changes. Table 5 provides a comparative view of business processes before and after SOA implementation, showing measurable improvements in submission speed, system coordination, and verification efficiency.

In comparison with previous studies, these results are consistent with the findings of Karim et al. [32] and Herzum [33], who showed that SOA adoption in financial institutions improves data interoperability and reduces operational latency. However, unlike their studies, which primarily focused on banking transactions, this research demonstrates the application of SOMF in the credit guarantee sector, where integration involves multiple heterogeneous banking partners under strict government regulations. The extension of SOMF into the credit guarantee domain represents a novel contribution. Furthermore, the integration of centralized API management differentiates this work from earlier SOA case studies, where decentralized integration often led to inconsistencies in service orchestration.

The results also confirm the initial hypothesis of this study: that applying SOMF to guide SOA implementation would reduce integration complexity and improve system scalability. Evidence from the comparative analysis in Table 5 shows that the redesigned architecture not only enhances technical

performance but also delivers practical benefits such as faster guarantee submissions, more efficient claims handling, and greater partner satisfaction.

Table 5. Business Process Of Guarantee Services Before and After SOA Implementation

Aspect	Before SOA Integration	After SOA Integration
Speed of Guarantee Submission Process	Slow submission process due to incomplete verification.	Faster process with automation and real-time data access through SOA integration.
Inter-System Coordination	Banking and guarantee systems are not integrated, causing data duplication and delays.	Integrated systems allow information to flow automatically between systems.
Efficiency of Guarantee Verification	Verification is repeated due to lack of synchronization between related systems.	Verification is more efficient with automatic data synchronization between systems.
Speed of Guarantee Submission Process	Slow submission process due to incomplete verification.	Faster process with automation and real-time data access through SOA integration.
Inter-System Coordination	Banking and guarantee systems are not integrated, causing data duplication and delays.	Integrated systems allow information to flow automatically between systems.
Efficiency of Guarantee Verification	Verification is repeated due to lack of synchronization between related systems.	Verification is more efficient with automatic data synchronization between systems.

From a practical standpoint, this model provides a replicable framework for other credit guarantee institutions in Indonesia that are required to comply with Permenko No. 1/2023, particularly in automating processes for guarantee issuance, fee collection, claim management, and subrogation. Theoretically, the study contributes to the growing body of literature on service-oriented modeling by demonstrating how SOMF can be adapted to a highly regulated financial service domain.

Nevertheless, limitations remain. Real-time synchronization with all partner banks is still constrained by the varying readiness of each institution's IT infrastructure. Future work could enhance the model by incorporating predictive analytics and machine learning to support decision-making, such as early risk detection in guarantee submissions or automated prioritization of claims. These enhancements would further strengthen the role of SOA-SOMF as a transformative approach for modern electronic systems in the financial sector.

4. Conclusions

This study demonstrated that the application of the Service-Oriented Modeling Framework (SOMF) to guide the implementation of Service-Oriented Architecture (SOA) successfully improved the integration and performance of credit guarantee systems in Indonesia. By replacing fragmented point-to-point connections with modular and reusable services, the proposed model effectively resolved the "integration spaghetti" problem and enhanced interoperability across multiple banking partners.

The findings confirm that SOMF is not only suitable for technical modeling but can also serve as a strategic framework to ensure regulatory compliance and operational scalability in the credit guarantee domain. For practitioners, this model offers a replicable architecture that can be adopted by other guarantee institutions to meet the requirements of Permenko No. 1/2023. For the academic community, this research extends the literature on service-oriented modeling by applying SOMF to a highly regulated financial service sector, demonstrating its adaptability beyond traditional enterprise systems.

While the results are promising, challenges remain in ensuring real-time synchronization with diverse banking infrastructures. Future research should explore the integration of predictive analytics and machine learning to enhance decision-making in guarantee evaluation, claims processing, and subrogation management. By advancing toward intelligent service orchestration, SOMF-based SOA has the potential to further transform digital ecosystems in the financial sector, reinforcing both efficiency and transparency for stakeholders.

Acknowledgements

The author declares no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] X. Wang, "Research on the Digital Transformation of the Financial Industry," *Highlights in Business, Economics and Management*, vol. 24, 2024, doi: 10.54097/582g8r55.

[2] A. Megargel and V. Shankararman, "Digital Banking Accelerator: A Service-Oriented Architecture Starter Kit for Banks," *IEEE Softw*, vol. 38, no. 3, 2021, doi: 10.1109/MS.2020.3029876.

[3] A. Papathomas and G. Konteos, "Financial institutions digital transformation: the stages of the journey and business metrics to follow," *Journal of Financial Services Marketing*, vol. 29, no. 2, 2024, doi: 10.1057/s41264-023-00223-x.

[4] M. R. Majedi and K. A. Osman, "A novel architectural design model for enterprise systems: Evaluating enterprise resource planning system and enterprise application integration against service oriented architecture," in *2008 3rd International Conference on Pervasive Computing and Applications, ICPCA08*, 2008. doi: 10.1109/ICPCA.2008.4783558.

[5] P. Hanafizadeh and M. G. Amin, "The transformative potential of banking service domains with the emergence of FinTechs," *Journal of Financial Services Marketing*, vol. 28, no. 3, 2023, doi: 10.1057/s41264-022-00161-0.

[6] E. Hustad and D. H. Olsen, "Creating a sustainable digital infrastructure: The role of service-oriented architecture," in *Procedia Computer Science*, 2021. doi: 10.1016/j.procs.2021.01.210.

[7] E. Aristanto, S. Hidayatullah, I. Windhyastiti, U. Khouroh, and I. K. Rachmawati, "Obstacles of Micro and Small Business Access to Kredit Usaha Rakyat (KUR) Program,"

MBR (Management and Business Review), vol. 6, no. 1, 2022, doi: 10.21067/mbr.v6i1.6719.

[8] M. Herr, U. Bath, and A. Koschel, "Implementation of a service oriented architecture at deutsche post MAIL," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3250, 2004, doi: 10.1007/978-3-540-30209-4_17.

[9] H. A. Barrett, T. A.; Proper, "Component-Based Solutions Under Architecture," *arXiv preprint arXiv:2105.08702*, 2021, [Online]. Available: https://arxiv.org/pdf/2105.08702

[10] N. Adams, A. Augusto, M. Davern, and M. La Rosa, "Why Do Banks Find Business Process Compliance so Challenging? An Australian Perspective," in *Lecture Notes in Business Information Processing*, 2022. doi: 10.1007/978-3-031-16171-1_1.

[11] E. Indriasari, H. Prabowo, F. L. Gaol, and B. Purwandari, "Digital Banking: Challenges Emerging Technology Trends and Future Research Agenda," *International Journal of E-Business Research*, vol. 18, no. 1, 2022.

[12] K. Fedotova, "Digitalization and Operational Efficiency in the Banking Sector: A Case Study of Nordea Bank," Lapland University of Applied Sciences, 2024. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/874188/Fedotova_Kseniia.pdf?sequence=2

[13] H. E. Hayretci and F. B. Aydemir, "A Multi Case Study on Legacy System Migration in the Banking Industry," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021. doi: 10.1007/978-3-030-79382-1_32.

[14] M. Bell, *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Hoboken, N.J. : John Wiley & Sons, 2008.

[15] J. Delsing, G. Kulcsár, and Ø. Haugen, "SysML modeling of service-oriented system-of-systems," *Innov Syst Softw Eng*, vol. 20, no. 3, 2024, doi: 10.1007/s11334-022-00455-5.

[16] F. Truyen, "Enacting the Service-Oriented Modeling Framework (SOMF) Using Enterprise Architect," Cephas Consulting Corp. Accessed: Feb. 13, 2025. [Online]. Available: https://community.sparxsystems.com/white-papers/621-93enacting-the-service-oriented-modeling-framework-somf-using-enterprise-architect

[17] M. K. B. PEREKONOMIAN and R. INDONESIA, "Peraturan Menteri Koordinator Bidang Perekonomian Nomor 1 Tahun 2023 tentang Perubahan Atas Peraturan Menteri Koordinator Bidang Perekonomian Republik Indonesia Nomor 1 Tahun 2022 Tentang Pedoman Pelaksanaan Kredit Usaha Rakyat." [Online]. Available: https://peraturan.bpk.go.id/Details/249011/permenko-perekonomian-no-1-tahun-2023

[18] M. P. Papazoglou and W. J. Van Den Heuvel, "Service oriented architectures: Approaches, technologies and research issues," *VLDB Journal*, vol. 16, no. 3, 2007, doi: 10.1007/s00778-007-0044-3.

[19] K. Grolinger, M. A. M. Capretz, A. Cunha, and S. Tazi, "Integration of business process modeling and Web services: A survey," 2014. doi: 10.1007/s11761-013-0138-2.

[20] E. S. Sanchez, P. J. Clemente, J. M. Conejero, and A. E. Prieto, "Business process execution from the alignment between business processes and web services: A semantic and model-driven modernization process," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2993883.

[21] D. Grant and B. Yeo, "Enterprise integration using Service-Oriented Architecture," *Issues in Information Systems*, vol. 22, no. 1, 2021, doi: 10.48009/1_iis_2021_164-177.

[22] V. K. Adari, "APIs AND OPEN BANKING: DRIVING INTEROPERABILITY IN THE FINANCIAL SECTOR," *International Journal of Research in Computer Applications and Information Technology (IJRCAIT)*, vol. 7, no. 2, pp. 2015–2024, 2024, [Online]. Available: https://ijrcait.com/index.php/home/article/view/IJRCAIT_07_02_142

[23] A. M. Riad, A. E. Hassan, and Q. F. Hassan, "LEVERAGING SOA IN BANKING SYSTEMS ' INTEGRATION," *Journal of Applied Economic Sciences*, vol. 3, no. 2, 2008.

[24] V. Kulkarni, "Service-Oriented Architecture (SOA) Vs. Microservices: Transitioning Legacy Banking Systems," *International Journal for Multidisciplinary Research*, vol. 1, no. 3, Nov. 2019, doi: 10.36948/ijfmr.2019.v01i03.36906.

[25] A. Megargel, V. Shankararaman, and D. K. Walker, "Migrating from monoliths to cloud-based microservices: A banking industry example," in *Software Engineering in the Era of Cloud Computing*, M. Ramachandran and Z. Mahmood, Eds., Cham: Springer, 2020, pp. 85–108. doi: 10.1007/978-3-030-33624-0_4.

[26] F. A. Rosyadi and A. N. Fajar, "Online Guarantee System With Soma Methods," *Jurnal Indonesia Sosial Teknologi*, vol. 5, no. 4, pp. 1709–1722, Apr. 2024, doi: 10.59141/jist.v5i4.1003.

[27] R. Vikaliana, A. L. Arifin, W. Latunreng, and A. Bari, "The role of the guarantee industry in strengthening MSMEs in Indonesia during pandemic Covid 19," *Jurnal Penelitian Pendidikan Indonesia (JPPI)*, vol. 8, no. 4, pp. 994–1000, 2022, doi: 10.29210/020221644.

[28] L. Liu, W. Li, N. R. Aljohani, M. D. Lytras, S. U. Hassan, and R. Nawaz, "A framework to evaluate the interoperability of0 information systems – Measuring the maturity of the business process alignment," *Int J Inf Manage*, vol. 54, 2020, doi: 10.1016/j.ijinfomgt.2020.102153.

[29] A. Torab-Miandoab, T. Samad-Soltani, A. Jodati, and P. Rezaei-Hachesu, "Interoperability of heterogeneous health information systems: a systematic literature review," *BMC Med Inform Decis Mak*, vol. 23, no. 1, 2023, doi: 10.1186/s12911-023-02115-5.

[30] N. H. M. Hassan, K. Ahmad, and H. Salehuddin, "Diagnosing the issues and challenges in data integration implementation in public sector," *Int J Adv Sci Eng Inf Technol*, vol. 10, no. 2, 2020, doi: 10.18517/ijaseit.10.2.10271.

[31] U. A. Samuel and O. N. Akpofure, "Overcoming Application Integration Challenges Using Middleware: An Overview," *IJARCCE*, vol. 9, no. 2, 2020, doi: 10.17148/ijarcce.2020.9201.

[32] M. R. Karim, A. H. S. Gill, and A. A. Arshad, "An Investigation of Benefits Affecting SOA Adoption in e-Banking," *International Journal of e-Education, e-Business, e-Management and e-Learning (IJEEEE)*, vol. 3, no. 2, pp. 142–146, Apr. 2013, doi: 10.7763/IJEEEE.2014.V4.326.

[33] P. Herzum, "SOA adoption in e-banking: a multi-case study," *Journal of Enterprise Information Management*, vol. 26, no. 3, pp. 238–254, 2013, doi: 10.1108/JEIM-07-2013-0042.