Practicing ESB Technology for Internet, Mobile Banking and Third Party Services Integration in case of Tsehay Bank

Baheredin Surur
HiLCoE School of Computer Science & Technology
babasurr@gmail.com

Abstract

This paper discusses the importance of enterprise service bus (ESB) in facilitating service integration and meeting the changing demands of businesses, particularly in financial institutions. With many banks in Ethiopia adopting ESB, it is crucial for new players like Tsehay Bank to choose the right, efficient, and cost-effective ESB. The paper presents a comparative analysis of four widely used ESB technologies - WS02, Red Hat Fuse, Fiorano, and Apache ServiceMix and recommends the best ESB option for Tsehay Bank. Overall, this paper highlights the benefits of employing ESB in service integration and recommend one of the ESB tool mentioned above for Tsehay bank digital banking division.

Keywords: Integration, API, Gateway, tools, Enterprise service bus (ESB), applications, classification, service oriented architecture (SOA)

1. Introduction

Tsehay Bank is one of the banks that has recently joined the banking industry in Ethiopia. [12] Tsehay Bank is a share company established and started giving services for its customers on July 23, 2022 with a business license issued as per the banking business proclamation No 592/2008. It commenced operations with an authorized capital of Birr 2.8 billion and paid-up capital of Birr 734 million. The Bank's paid-up capital has reached Birr 1.1 billion (as of August 7, 2023) and raised its subscribed capital to Birr 5 billion.

The bank currently has over 750 employees. Tsehay Bank celebrated its one-year anniversary a few months ago and during this short period of time, the bank was able to open more than 80 branches for conventional banking service and started offering digital banking services such as mobile banking, ATMs, telegram bot services and more by utilizing Temenos core banking solution and Oracle database. Overall, Tsehay Bank has made significant progress in its first year of operation, but there is still room for improvement. By investing in infrastructure, talent, and customer experience, the bank can establish itself as a strong player in the Ethiopian banking industry by aspiring to deliver sustainable values for stakeholders with best customer experience and through state-of-the-art technology.

Tsehay Bank, despite being new to the banking industry, has already started offering

digital banking services such as payment cards, mobile and online platforms. The bank aims to achieve accessibility, process efficiency, and improved competitiveness by managing its existing system development process and infrastructure using an agile iterative approach. Additionally, the bank is waiting for the main digital and infrastructure project to commence to make its services robust enough to meet the ever increasing demand of customers with changing technology and business model and regulation.

Tsehay Bank also completed a lot of third party integration with Ethiotelcom, Ethswitch, Safarcom, MPESA, DARs, ECX, NBE, Guzo Go and others many more in the pipe line. Being new in the industry has not limited the bank to provide most of the services provided by other pioneers in the business. To expand the currently active services and incorporate the emergent services the bank still need to do a lot in infrastructure development and acquiring right tool for integration of different in house and third party applications. Used in House developed middleware (using Java spring boot framework) which lacks micro-service capability which make change and scalability difficult and any change made in one services will affect all system integrated in the middleware used. With current middleware used third party integration is not with required time frame and difficulty in deployment, protocol conversion, Micro services, plug and play features and scalability, flexibility with interface and automated test

Due to globalization and digitization most business operations are continuously changing and integration with different companies are becoming a necessity, and when it comes to a bank the stake is beyond that. Because of its emergent nature it is crucial to have well regard open application integration tools that can facilitate third party integration with different connection and format modalities. The concept of an Enterprise Service Bus (ESB) emerged in the early 2000s as a response to the challenges posed by traditional point-to-point integration methods. During that time, businesses faced significant costs and maintenance issues with their point-to-point connections for system integration.

This practice can be very helpful for Tsehay Bank to bring new reliable services quickly for customers. Enterprises all over the world have to face a fierce competition. In order to stay in business, they constantly have to automate their business processes, integrate with their business partners and provide new services to their customers.

Thus, the objective of this paper is to recommend the best Enterprise Service Bus (ESB) tools to Tsehay Bank and support in choosing the one that is good enough in facilitating communication and integration of applications and services.

2. Related Work

2.1 Application Integration Progress and ESB Technologies

Harmonizing to [3] IT infrastructure can have a wide range of diverse technologies, platforms and applications that need to interoperate to serve business goals. Furthermore [3]

explained Enterprise Application integration (EAI) which can be defined as "the enablement of data sharing between heterogeneous information systems within one enterprise" is becoming more and more essential to enterprises as it streamlines business processes, ensures the integrity between different systems, reduces labor cost and minimizes human errors. It is mentioned on [3] IT infrastructure runs a variety of diverse technologies, platforms and systems that may involve legacy systems, ERP solutions, and cloud-based systems, makes such goal hard to achieve, especially with the rapid growth of information technologies and non-stop business growth that rely heavily on Information Technology. Organizations like banking industry (Fintech) that relay on IT for their day today services are required to merge the different IT components and solutions together to form a solid infrastructure that enables easy sharing of data and information across different systems in a seamless fashion.

According to [2] the first generation of EAI involves the use of point-to-point middleware which provides APIs or program interfaces to link one application to another using message queuing technology such as Common Object Request Broker Architecture (CORBA) or remote procedure calls (RPCs). Here the middleware feature is embedded into applications and need extra work or modification at both end to complete the integration process and also it is kind of one to one relationship and it is difficult to manage as the number of integration increase and correspondingly increase development cost, maintenance and change.

The second generation of EAI is the centralized integration platforms which is known as message broker. In this integration approach, the message broker is a central hub that is responsible for data transformation and routing. Here the process of connecting various systems and transferring messages between them can be a complex task. However, with the use of adapters, this process is made simpler and more efficient. Adapters are used to connect all systems to a hub, allowing for seamless communication between them. In addition to facilitating communication, adapters also play a crucial role in converting data formats into a format that the hub can understand. Once the sender application publishes messages to a message broker, the message broker takes over and processes the message by applying data transformation. This ensures that the message is delivered in the correct format and can be understood by all systems involved in the communication process. Overall, the use of adapters and message brokers is essential for ensuring smooth and efficient communication between different systems.

In recent years, EAI moved toward loosely coupled fashion and standard exchange semantics. This can be seen in Enterprise Service Bus (ESB) integration technology which is standard-based integration platform that emerged Service Oriented Architecture (SOA) technologies and Message Oriented Middleware (MOM) to standardize information flow and message processing using a central hub [4]. ESB is a highly distributed messaging backbone which combines traditional middleware technology and web services to towards

more loose coupling and standard semantics. ESB provides flexibility, reusability and neutrality of SOA technology and web services to the integration world. The fact that SOA technology is platform- and programming language- independent, makes it ideal solution to unify the integration access diverse systems [5].

At stated on [11] architecturally in the absence of ESB products, heterogeneous applications could be integrated through alternative architecture as shown in figure 1. However, this application integration architecture is difficult to scale and maintain. Moreover, adding an application would mean interface customization of all other networking applications.

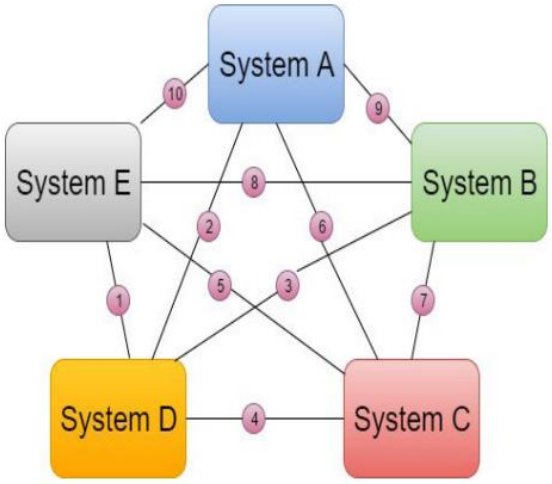

FIGURE 1. Point to point integration architecture.

Enterprise service bus architecture: As shown in figure 2, applications are indirectly linked through ESB instead of being linked directly to each other. Enterprise service bus architecture. Responsible for all the built-in logic necessary to make the structures interact/integrate.

The ESB was introduced as a more efficient and cost-effective solution for integrating systems and applications. Because an ESB controls the way work moves, it makes it easy to change components or add additional components to an application. It also makes for a convenient place to enforce security and compliance requirements, log normal or exception conditions and even handle transaction performance monitoring. Finally, an ESB also provides load balancing, in which multiple copies of a component can be instantiated to improve performance, as well as failover support should a component or resource fail.

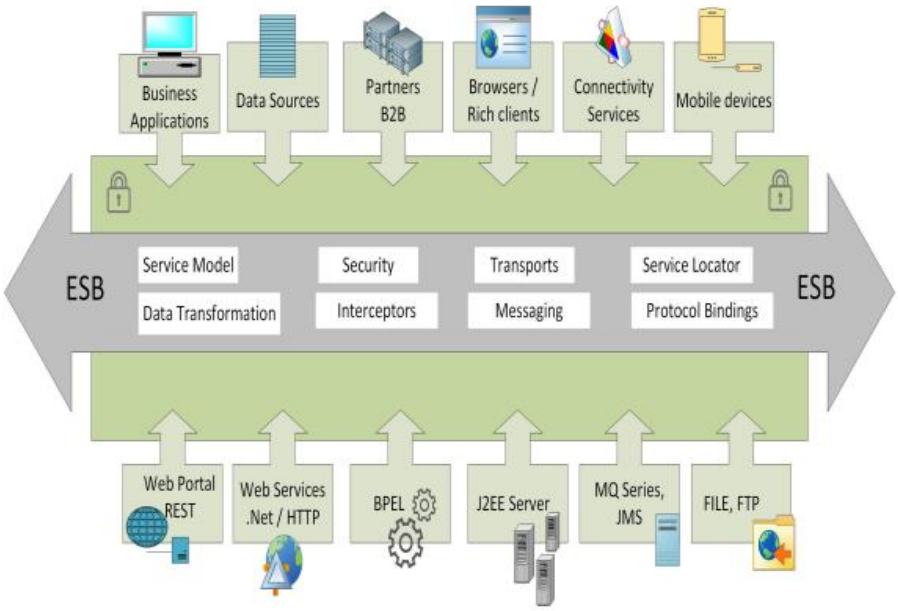

FIGURE 2. Enterprise service bus architecture.

2.2 Benefits of ESB

There are many advantages to us to get by including ESB in the integration of different applications and services

ESB enables Standardized integration: It reduces proprietary vendor lock-in and increases the ability to integrate with business partners using standard interface and protocols. It also helps businesses in leveraging the existing IT staff instead of looking talent or specialized integration architects outside.

ESB is loosely-coupled: This is also essential aspect of ESB as the application logic is separated from the connection infrastructure and independent web services can be accessed as shared services by the distributed systems connected to the bus which grants more flexibility and scalability

ESB will improve services flexibility and reusability: A pattern which uses web services, intelligent routing and message transformation and services orchestration in order to make services more flexible and reusable. Thus, the company is not just prepared for the short run but also for the long run, for the future technological changes and evolution.

ESB enables Service substitution feature: This enable the substitution of one service implementation by another with no effect to the clients of that service. This requires both the service interfaces that are specified in a manner that is independent of the service location and communication protocol that is involved.

ESBs can be easily scaled horizontally by adding more instances to meet growing

demands. This enables organizations to handle large volumes of data and transactions efficiently.

ESB offers flexibility in terms of message routing and transformation. It supports many protocols and message formats and can transform messages between different formats. Organizations can then integrate diverse messaging applications with ease.

ESB supports various security mechanisms, such as authentication, authorization, and encryption, which ensure the confidentiality and integrity of the data.

ESBs provide a library of reusable components that can be leveraged across different applications and systems. This reduces development time and effort, giving organizations greater code reusability.

ESB enables support of versatility: is a versatile integration platform that offers support for multiple integration paradigms. This is made possible by its ability to accommodate a range of interaction patterns and to provide a unified infrastructure for three major styles of Enterprise Integration - Service-oriented architectures, Message-driven architectures, and Event-driven architectures. In addition to this, ESB also provides advanced capabilities for mediating and transforming service messages and interactions, thereby enhancing the overall efficiency and effectiveness of the integration process.

2.3 WS02

WS02 ESB is an open-source enterprise service bus solution that allows for the integration of various applications and services. It supports multiple protocols and provides features such as message routing, transformation, mediation, security, and monitoring. It also has a graphical user interface for easy configuration and management of integration flows. WSO2's API management offering is available in two versions: on-premises (WSO2 API Manager) and public cloud (WSO2 API Cloud) [6].

2.4 Red Hat Fuse

Red Hat Fuse ESB is an open-source enterprise service bus solution that enables the integration of various applications and services. It provides a lightweight, modular, and flexible platform for building and deploying integration solutions. Red Hat has a strong presence in the application infrastructure market. It has now integrated a powerful API management platform with several application infrastructure offerings, including Red Hat Fuse (an integration platform), AMQ (a messaging platform), OpenShift (a container offering) and Red Hat Single Sign-On [6].

2.5 Fiorano

Fiorano ESB is a high-performance, scalable, and flexible enterprise service bus solution that enables the integration of various applications and services. Fiorano ESB supports multiple protocols and standards, including JMS, HTTP, SOAP, REST, and FTP. It also has a visual drag-and-drop interface for easy configuration and management. The Fiorano ESB implementation supports development in multiple programming languages. This, coupled with the inherent portability of the underlying infrastructure, makes the Fiorano ESB a true multi-language, multi-platform enterprise backbone.

2.6 Apache ServiceMix

Apache ServiceMix is an open-source enterprise service bus (ESB) and integration platform that provides a flexible and powerful infrastructure for integrating various systems, applications, and services. It is based on the Java Business Integration (JBI) standard and supports a wide range of protocols and technologies, including web services, messaging, and more. ServiceMix offers features such as routing, transformation, orchestration, and monitoring, making it suitable for building scalable and reliable integration solutions. It is widely used in enterprise environments for building and managing complex integration architectures. Apache ServiceMix is a flexible, open-source integration container that unifies the features and functionality of Apache ActiveMQ, Camel, CXF, and Karaf into a powerful runtime platform you can use to build your own integrations solutions. The software is distributed under the Apache License. ServiceMix supports the OSGi framework. ServiceMix integrated Spring Framework support and can be run at the edge of the network (inside a client or server), as a standalone ESB provider or as a service within another ESB. ServiceMix is compatible with Java Platform, Standard Edition or a Java Platform, Enterprise Edition application server. ServiceMix uses ActiveMQ to provide remoting, clustering, reliability and distributed failover. The basic frameworks used by ServiceMix are spring and XBean.

3. Comparative Analysis

3.1 Selection Criteria

The primary goal of this paper is to establish a framework for evaluating proposed ESB architectures or designs, in order to determine if they are capable of delivering the anticipated benefits within our organization and accordingly to compare and select the best possible ESB, to achieve this objective we will rely on the most widely accepted elements of the ESB definition as outlined in [8]. It is mentioned there are lot of criteria’s that can be used but here defined some minimum or mandatory characteristics that should be exhibited by an Enterprise Service Bus.

3.2 Methodology

Out there many Enterprise service bus technologies (ESB) to choose from but out of all I have selected four ESB technologies to be compared. Based on their capabilities to be deployed in in-premise since our country legislation is not allowing cloud based services mainly in the financial sectors and also there is no ground to use cloud based ESB while all core services of the bank are on premise beside security related concerns when using open source technologies.

After short listing the ESB technologies I have selected four parameters for comparison among them and these are Ease of Learning, Features and Protocols, free resource & community support, and License & Cost. Parameters were explored through literature in the area of study, the contents found on web especially on particular technologies under our focus, looking on customer reviews who already deployed those technologies and by analyzing those customers' feedbacks and testing done on the technologies by setting up on small scale lab environments. Here it set parameters value both quantitative and qualitative means of measurement. Ease of Learning rated qualitatively as easy, moderate and steep considering the amount of time it takes to adopt the technology and the complexity nature of the system. Features and Protocols rated using high, few and average variable based on the number of features and protocols they can support with. Free resource and community support rated as low and high variables based on the availability of documents in the web and community pages. All parameters are combined together and presented in a tabular form.

License & Cost is crucial part for selecting final technologies and has been given the highest priority and technologies with highest priority will be chosen and recommended. The final ESB technology nominated shall improve the service integration and service orchestration at Tsehay Bank S.C

3.3 Learning Curve

When implementing new technology, it is important to study the learning curve of the artifact we are going to adopt. At present, teams working in core systems and Digital banking departments at Tsehay Bank has not yet applied features of ESB in full scale during application customization and integration. Sometimes even if the teams are conscious on the technologies and understand the rewards that can be exploited; it is worthless unless it is easily customizable, can be integrated without any hassle & complexity with minimum effort and time. It should be adopted within a very short period, with minimum and self-learning based approach. Otherwise, it impacts greatly the adoptability of the tool and discourages team's effort from quickly learning and applying the technology in their integration work. Hence, this need to be one of the main criteria when we select any artifact and familiarize ourselves with it.

The artifact selected here in this paper have comparable GUI and they look like to be

fairly easy to use. But, ESB, like all new technologies we encounter in our day to day operations, are always new unless we have prior knowledge and exposure.

In WS02 the learning curve can be considered relatively good even if it requires understanding of concepts such as API management, message brokering, and service orchestration. However, WSO2 API Manager offers an intuitive and user-friendly interface that makes it easy to navigate and perform tasks. The layout and organization of features are well-designed, ensuring a smooth and efficient user experience. Whether it's creating APIs, managing subscriptions, or analyzing data, the interface provides a seamless experience for users. Apache ServiceMix is an open-source ESB that is based on Apache Karaf and provides a lightweight, flexible, and powerful integration platform. Its learning curve can be steep for those who are new to ESBs and related technologies, as it requires a good understanding of OSGi (Open Service Gateway initiative) and the Apache Karaf container. Overall, the learning curve for Apache ServiceMix can be significant for beginners.

The learning curve for Red Hat Fuse can also vary depending on the individual's background and experience with ESB technologies. For developers and IT professionals with prior experience in ESBs and integration platforms, the learning curve for Red Hat Fuse ESB may be relatively moderate. However, for those who are new to ESBs and integration technologies, the learning curve for Red Hat Fuse ESB may be steeper. And it demands them to familiarize themselves with the core concepts of ESB, such as message routing, transformation, mediation, and service orchestration, as well as learn how to configure and manage Red Hat Fuse ESB instances and integrate various systems and applications.

Fiorano has a learning curve that ranges from moderate to steeper depending on the individual, but its visual development tool makes it easier for non-technical users to understand to certain extent.

3.4 Features and Protocols

Each ESB tool discussed in the paper offers a rich set of features and supports various protocols with slight differences. WSO2 provides a wide range of features, such as API management, data integration, and support for microservices architecture. It also supports protocols like SOAP, REST, and WebSockets. Apache ServiceMix supports a wide range of protocols for communication and integration. Some of the key protocols supported by Apache ServiceMix include: HTTPS, JMS, SOAP SOAP, REST, FTP, SMTP, JDBC, AMQP and WebSockets. Fiorano stands out for its visual development tools, real-time messaging, and event processing capabilities, and supports protocols like JMS, AMQP, MQTT, and REST. Red Hat Fuse offers integration with other Red Hat products, containerization support, and cloud-native development features, while also supporting protocols like HTTP, JMS, AMQP, and MQTT. Each ESB tool has its own strengths and capabilities to cater to different integration needs.

3.5 Resources and Support Community

WSO2 offers comprehensive resource support with a dedicated team of experts who provide technical assistance, documentation, and training materials. They also have a large knowledge base, forums, and blogs for developers to find solutions to their queries. WSO2 also has a professional services team that offers consulting, implementation, and support services for enterprises.

WSO2 has a larger library of integrations compared to, which may be due to its longer presence in the market. The Apache ServiceMix community provides extensive support and documentation to help users understand, implement, and troubleshoot the platform through, mailing list, forums, issue tracker, documentation and community Events. Fiorano has a smaller community of users and contributors, but offers dedicated support services for enterprise customers. Red Hat Fuse benefits from the larger Red Hat community and support resources.

3.6 License & Cost

The tools discussed in this paper offer both open-source and enterprise options. For example, WSO2 has both open-source and enterprise versions, with pricing based on the number of production cores and support level. The Free License for Non-commercial, Educational, or Trial use is also available. Apache ServiceMix also free but there will be associated cost when professional support is need. The monthly average fee ranges from 200 to 500 USD for Red Hat Fuse and Fiorano ESB, which have relatively higher prices and depend on the number of cores and support level agreement.

Finally, all parameters are combined together in the table below.

Table 1: All Comparison parameters combined together

	WS02	**Apache ServiceMix**	**Red Hat Fuse**	**Fiorano**
Learning Curve	Easy	Steep	Moderate	Steep
Features and Protocols	High	High	Average	Average
Resources and Support Community	High	High	Average	Low
License & Cost	Free	Free	$500 per month	Free

4. Recommendation and conclusion

4.1 Recommendation

The digital banking division of Tsehay Bank S.C is dedicated to delivering cutting-edge services in the digital banking sector. In order to achieve this goal, the bank recognizes the need to reevaluate its use of ESB (Enterprise Service Bus) technologies or tools. As a new

bank, this presents an opportune moment to carefully select the right tool, minimizing the risk of future rework. Once the appropriate tool has been chosen and the setup process completed, subsequent integration efforts can be carried out without disrupting the established framework, thereby reducing costs and minimizing efforts. However, selecting the right tool is not a straightforward task, especially for a division with limited experience and exposure in the domain. To address this challenge, four prominent ESB options have been identified as potential candidates. These options have been compared based on criteria such as learning curve, features and protocols, resources and support community, as well as license and cost. Upon analysis, it is evident that WS02 stands out as the most favorable choice. With its user-friendly learning curve, extensive range of features and protocols, robust resource and support community, and cost-effectiveness due to its free licensing, WS02 emerges as the optimal choice to be selection based on the parameters defined.

4.2 Conclusion

The ESB tools discussed in this paper offer substantial benefits to Tsehay Bank's digital banking division, particularly in managing the increasing volume of service and application integration tasks. By leveraging ESB tools, the bank can save time, enhance operational efficiency, reduce service delivery timelines, and ultimately elevate customer satisfaction. Furthermore, utilizing ESB for integration work simplifies troubleshooting processes and facilitates adjustments to business rules and logic. Among the ESB options considered, I recommend WS02 due to its free licensing and robust ESB technology. While open-source products like WS02 may have limitations in terms of source code accessibility and potential vulnerabilities, especially in a corporate environment like Fintech, the availability of enterprise solutions provides a safety net when needed. It is prudent to initially familiarize the team with free options before considering enterprise solutions. However, it is important to note that this recommendation is based on theoretical analysis, as practical testing of the tool's performance was constrained by inadequate lab resources Future work should include performance-related measurements as a significant parameter for the selection process.

References

[1] Razvan Daniel , Radu Stefan , Laurenhu Ciovica “Enterprise Service Bus-A Backbone for SOA”

[2] Omer Aziz, Muhammad Shoaib, Adnan Abid , Naeem Asleam “ Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study”

[3] Afnan Alawami “Evolution of Enterprise Application Integration Technologies” (May 2022)

[4] Periorellis, P., and P. Watson. "Enterprise Service Bus: An overview." School of Computing Science Technical Report Series (2007).

[5] Wei, Ning. "Research on an information application integration platform based on SOA and web service." International Journal of Multimedia and Ubiquitous Engineering 10.12 (2015): 165-174.

[6] Paolo Malinverno, Mark O'Neill, Aashish Gupta, Kimihiko Iijima. " Magic Quadrant for Full Life Cycle API Management" (October 2019)

[7] Fiorano Software Inc "Fiorano ESB Architecture" (2006 -2009)

[8] Mark Endrei Jenny Ang Ali Arsanjani Sook Chua Philippe Comte Pål Krogdahl Min Luo Tony Newling " Patterns: ServiceOriented Architecture and Web Services" (2004)

[9] Adam Badura; Bartosz Sakowicz; Dariusz Makowski "Integration of management protocols based on Apache ServiceMix JBI platform" (2009 April)

[10] Apache Service Mix https://servicemix.apache.org/

[11] Omer Aziz, Adnan Abid , Naeem , Rubab Shaer "Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study" (2020 February)

[12] Tsehay Bank Website , https://tsehaybank.com.et/company-profile/