ISSN: 2755-0214

Journal of Economics & Management Research

Review Article

Open Access

Investigating the Potential of Enterprise Service Bus as a Fundamental Facilitator for Future Information Technology Infrastructure

Narayana Challa

Lead Integrations Developer

ABSTRACT
The Enterprise Service Bus (ESB) is a new integration infrastructure, and this article comprehensively explains the technology. It discusses the history of EAI and the technologies that form the backbone of an ESB—namely, Message-Oriented Middleware (MOM) and Service-Oriented Architecture (SOA). It details an ESB's basics and then presents Mule, an open-source ESB solution. Finally, an example app is provided to show the various benefits that an ESB can provide. Users can apply this framework with the help of ESB solutions, although these products have varying approaches and features. Integrating disparate applications by linking them over a communication bus is central to the ESB design; this bus ensures that each application can talk to it separately without needing to be aware of the presence of the others. Instead of the fragile and difficult-to-manage point-to-point integration, this decoupling of systems allows for more effective communication and eliminates the need for dependent systems.

***Corresponding author**
Narayana Challa, Lead Integrations Developer.

Received: June 07, 2021; **Accepted:** June 09, 2021; **Published:** June 16, 2021

Keywords: Enterprise Service Bus (ESB), Internet of Things (IoT), MOM, SOA, SOAP

Introduction
The Enterprise Service Bus (ESB) is a relatively new concept in the software industry, and because of the inherent uncertainty around it, it is a subject that holds a lot of interest for people. The question "What exactly is an Enterprise Service Bus?" is the most often asked. The answer is difficult to pinpoint because everyone acknowledges no single meaning of the term—several schools of thought regarding the fundamental components and technologies needed to build an ESB. As a consequence of this, a significant number of sellers assert that their wares qualify as ESBs or are compatible with the ESB principles. Organizations like Oracle, BEA Systems, IONA Technologies, Sun Microsystems, and IBM include ESB solutions in their integrated middleware product portfolios. Other firms, such as Cape Clear Software, Fiorano Software, and Sonic Software, build ESB products as their primary line of business [1].

Gartner analysts came up with the name "Enterprise Service Bus" for the first time in 2002, which is how this peculiar circumstance came about. They concluded that Service-Oriented Architecture (SOA) required a new infrastructure incorporating Message-Oriented Middleware (MOM), web services, transformation, and routing intelligence. This would act as the architecture's backbone. Many attempts have been made to put this idea into practice, and some products have evolved from web services infrastructure solutions or lightweight messaging products, while others have evolved from enterprise application integration suites that have included support for SOA. Despite the various techniques and definitions, the core ideas are comparable, and this article will explain and demonstrate those core ideas [2].

The Enterprise Service Bus (ESB) is a distributed integration infrastructure that operates on open standards and message-based communication. It aims to facilitate secure and reliable interactions between disparate distributed applications and services by providing routing, invocation, and mediation services. ESBs are implemented through service containers that are distributed across a networked environment. These containers host integration services such as routers, transformers, application adapters, and MOM bridges, providing a wide range [3].

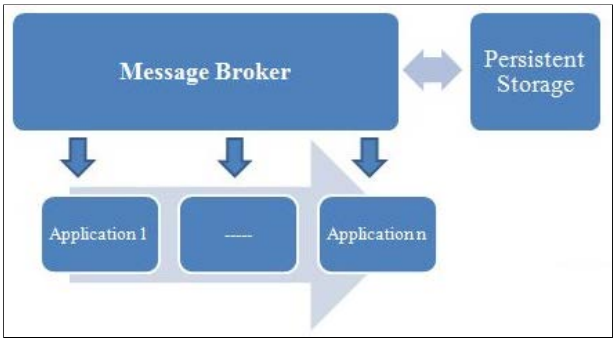

Figure 1: Architecture of a Message-Oriented Middleware

of communication facilities. In modern ESB solutions, messaging infrastructure is often built on top of middleware systems based on JMS, which ensures message delivery. Applications are connected to the bus using application adapters or a supported messaging mechanism. To support service-oriented architecture (SOA), ESB service containers must include all essential web service technologies. The components of the ESB and the mechanisms used to connect resources must be based on open standards to ensure interoperability and protect investments [4]. ESBs can send requests and receive responses from integration services and integrated resources, referred to as "invocation." To achieve this, ESBs must adhere to web service communication standards such as SOAP, WSDL, UDDI, and the WS-* family of standards. Integration with MOM systems and application servers also requires implementing the JMS API and J2EE Connector Architecture. Additionally, ESBs must be able to handle underlying protocols such as TCP, UDP, HTTP, and SSL. Other communication mechanisms such as JBI, RMI, JDBC, SMTP, POP3, FTP, and XMPP may also be employed [5].

Routing is a crucial feature of ESBs that enables the decision of a message's destination during transport. To facilitate routing and other communication features, disparate messaging endpoints must be referenced using Uniform Resource Identifiers (URIs). WS-Addressing may also be implemented in ESB solutions to describe web service endpoints in a transport-neutral manner. The decision of a message's destination can be based on several conditions, leading to different types of routers [6].

A content-based router inspects a message's content and forwards it to different channels depending on that content, allowing the sender to send messages without specifying an exact destination. Content-based routing for XML messages can be implemented using XPath to address parts of a message needed for the routing decision. A message filter is a form of a content-based router that only forwards a message if its content matches a specific criterion; otherwise, the message is deleted [7].

When a message must be sent to multiple recipients, a recipient list router can compute a list of destinations or have a static configuration for multiple receivers. A splitter can break down a single message into individual messages if messages have various parts. An XML message splitter can utilize XPath to address parts and transformations based on XSL for generating separate messages [8].

An aggregator collects and stores individual messages, and when it receives a complete set of related messages, it sends a single message distilled from the individual messages to the configured destination. A sequencer is a router that collects related but out-of-sequence messages and forwards them in the correct order. For the sequencer to function, each message must have a unique sequence number [9].

The various routers can be combined to create complex message flows, and all routers can be implemented as dynamic routers, which means that the router can reconfigure its routing rules based on particular configuration messages that participating destinations can send.

Mediation converts or transfers dissimilar resources such as transportation protocols, message formats, and message contents. This conversion is crucial for integration because applications typically do not conform to a uniform data format. XSL and XPath are powerful tools used to manipulate XML messages. These standards enable an ESB to offer generic XML transformer components that can be configured through an XSL stylesheet. In some cases, more complex transformers may require additional resources connected to the bus, such as a database, to supplement the message with additional information. Particular types of transformation services include normalizers, content enrichers, content filters, and envelope wrappers [10].

ESB solutions often offer a variety of application adapters, including adapters for widely used application packages such as Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and Customer Relationship Management (CRM). These adapters connect to the native transaction interfaces, APIs, and data structures of the respective business applications and provide a standardized interface that facilitates the reuse of business logic and data. Typically, most adapters of a specific vendor operate similarly, which reduces the skills required to work with each connected system. Using prebuilt adapters minimizes the effort needed to integrate applications into a Service-Oriented Architecture.

Role of ESB as a Game-Changer in the IT Industry

The Enterprise Service Bus (ESB) has already significantly impacted the present state of IT infrastructure, but its true potential lies in its ability to shape the future of digital transformation. As organizations adopt cloud-based and hybrid infrastructures, ESBs will become increasingly essential in ensuring seamless connectivity and interoperability across the IT landscape.

One of the key benefits of using an ESB is its ability to simplify the integration process, reducing the time and effort required to connect disparate systems and applications. This is particularly important as organizations adopt new technologies and services increasingly rapidly. ESBs provide a standardized approach to integration, enabling organizations to integrate new applications and services as needed more quickly. This can lead to increased agility, reduced costs, and improved time-to-market for new products and services.

Another way ESBs will impact the future is by enabling organizations to build more modular and flexible IT infrastructures. ESBs break down silos between applications and systems, facilitating better collaboration and communication across the enterprise. This allows organizations to quickly adapt to changing business needs and take advantage of new opportunities.

ESBs also provide a powerful platform for implementing Service-Oriented Architectures (SOA), allowing organizations to build more resilient and responsive IT systems. SOA enables the development of loosely coupled applications that can be easily updated and modified without disrupting the entire system. This can lead to increased innovation and greater agility in responding to customer needs.

Moreover, ESBs will play an increasingly important role in supporting the Internet of Things (IoT), poised to become one of the most significant technological trends of the coming years. As more devices and sensors connect to the internet, there will be a growing need for platforms to manage data flow between these devices and backend systems. ESBs provide a centralized control point for managing data flows across IoT devices, enabling organizations to harness the full potential of this emerging technology.

Core Functionalities of ESB

The core functionalities provided by an enterprise service bus (ESB) include uncoupling, transport protocol conversion, high availability and scalability, message transformation, routing, and security. However, no universal standards for implementing ESBs or principles governing their use exist. While some message-oriented middle-layer (MOM) technology suppliers have adopted the ESB model as a de facto standard for SOA architecture, other IT companies have applied the ESB label to existing middle-layer and integration approaches without adhering to the fundamental characteristics of bus theory. ESBs as a technology category originated in 2002 with the introduction of Sonic ESB. Since then, ESB development has taken on a life of its own, with software platform suite vendors (e.g., IBM and BEA), EAI suppliers (e.g., TIBCO and cloud technologies), and providers of web services toolsets all incorporating the ESB model into their offerings. However, ESB vendors and suppliers still compete on various factors, such as fundamental designs, connectivity options, ease of access, and quality of service (QoS) issues.

To date, no systematic mapping research has been published on ESBs. Therefore, this study aims to identify and present existing research on ESB applications, obtain valuable real-world results, and identify gaps in the current research that require further exploration. The authors conducted a systematic mapping study to achieve this goal, an emerging technique used to construct a classification pattern and plan an area of interest. Using search strings in digital libraries, the authors identified 214 articles from an initial 16,400 articles published between 2011 and 2019. After categorizing the studies based on their contribution type, ESB applications, research type, and approach, the authors selected 22 studies for inclusion in the mapping study. The results showed that ESB has gained significant support since 2011, with most studies presented at conferences rather than published in journals. Solution proposal and implementation & evaluation research were the most common research types, and a method was the most frequently used approach in the selected studies.

Conclusion

In conclusion, the ESB is crucial in a Service-Oriented Architecture (SOA), a dependable and adaptable framework for integrating diverse systems and applications. Organizations' adoption of cloud-based and hybrid infrastructures has increased the importance of Enterprise Service Buses (ESBs) in facilitating smooth communication and interoperability within the IT environment. In this study article, an analysis has been conducted on the fundamental characteristics of an Enterprise Service Bus (ESB), encompassing message routing, mediation, transformation, and application adapters. Additionally, an examination has been conducted on the advantages of using an ESB, including the mitigation of intricacy, the enhancement of scalability, and the fortification of security measures. Furthermore, we have identified obstacles and potential disadvantages of implementing an ESB, including heightened latency, vendor dependency, and supplementary operational costs.

Even with these problems, the benefits of employing an ESB significantly surpass the disadvantages, particularly in the context of extensive and intricate information technology (IT) landscapes. ESBs offer a standardized methodology for integration, facilitating the seamless incorporation of new applications and services into organizations' existing systems. This phenomenon could enhance operational flexibility, decrease expenditures, and expedite the introduction of novel offerings to the market. In addition, ESBs provide a robust framework for implementing Service-Oriented Architectures (SOAs), enabling organizations to construct an IT infrastructure characterized by modularity and flexibility, hence facilitating prompt adaptation to evolving business requirements. ESBs play a crucial role in enhancing cooperation and communication within organizations by dismantling barriers between various applications and systems.

References

1. Kharade KG, Kharade SK, Katkar SV, Kamat RK (2020) Performance improvement using covering index to reduce row identifier lookups. Recent Advances in Science and Technology Research 3: 114-119.
2. Aziz O, Farooq MS, Abid A, Saher R, Aslam N (2020) Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study. IEEE Access 8: 31180-31197.
3. Menge F (2007) Enterprise Service Bus, presented at the Free and Open Source Software for Geospatial 1-6.
4. Schmidt MT, Hutchison B, Lambros P, Phippen R (2005) The enterprise service bus: making service-oriented architecture real. IBM Systems Journal 44: 781-797.
5. Luo M, Zhang LJ (2008) Practical SOA: Service Modeling, Enterprise Service Bus, and Governance. IEEE Congress on Services Part II (services-2 2008) Beijing, China 13-14.
6. Qiu G, Song R, He S, Song Z (2018) Diversified Bus Services and Enterprise Information System: An Example of Beijing. Sixth International Conference on Enterprise Systems (ES), Limassol, Cyprus 172-179.
7. Dharrao D, Juke N (2018) Accurate face tagging with efficient regularizer and distance metric algorithm. International Conference On Advances in Communication and Computing Technology (ICACCT) 468-472.
8. Jingde Cheng (2005) Connecting components with soft system buses: a new methodology for design, development, and maintenance of reconfigurable, ubiquitous, and persistent reactive systems. 19th International Conference on Advanced Information Networking and Applications (AINA'05) 1: 667-672.
9. Cheng J (2006) Persistent computing systems as continuously available, reliable and secure systems, First International Conference on Availability, Reliability and Security (ARES'06), Vienna, Austria 8-638.
10. Li G, Xiao J, Li C, Li S, Cheng J (2012) A Comparative Study between Soft System Bus and Enterprise Service Bus. 2012 International Conference on Computer Science and Service System, Nanjing, China 557-561.