

Evolution of Enterprise Application Integration Technologies

Afnan Alawami¹

¹²Saudi Aramco, Dhahran, Saudi Arabia

DOI: <https://doi.org/10.5281/zenodo.6536509>

Published Date: 10-May-2022

Abstract: IT infrastructure can have a wide range of diverse technologies, platforms and applications that need to interoperate to serve business goals. Enterprise Application integration (EAI) which can be defined as “the enablement of data sharing between heterogenous 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 labour cost and minimizes human errors. This paper presents the different generations of enterprise applications integration technologies that evolved over the past 30 years to link different IT enabled solutions together. The paper covers the architectural features and challenges associated with each generation of Enterprise Application Integration [EAI].

Keywords: Enterprise Application Integration, Middleware, Integration Platform as a Service.

I. INTRODUCTION

One of the most important objectives of IT departments is to streamline business processes and ensure that IT backbone is powerful enough to support these objectives. The fact that 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. IT organizations 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. The hidden layer allows interoperability between different technologies is Enterprise Application integration (EAI). EAI links all IT solutions and platforms in a seamless manner and enables systems that runs on different platforms and uses different programming languages to communicate with one another and syncs the information in real-time to improve productivity and ensures information integrity.

The EAI concept evolved from being about data integration, where the focus was on data format, to system integration, where heterogenous systems interface with each other and exchange data. The latter further evolved into business processes integration, when organization realized the need for business processes to be controlled, modelled and interoperate with each other [15]. EAI can automate business processes and allow interoperation between them. When an event happened in one system, EAI can automate the execution of consequent events in different systems which facilitate complex business processes [3].

EAI benefits to businesses can be summarized as follows: [10]

- It helps to streamline business processes that are carried out across different functions of common businesses such as supply chain, human resources and marketing.
- It helps in sharing data with vendors and business partners.

- It enhances customer experience by providing a single solution that gathers all sorts of information available inside the organization.
- It is an optimal solution to achieve standardization within an organization.
- It helps businesses to reduce production time and service delivery time.

Over the years, EAI has been always a challenge for organizations. However, it evolved over different generations since businesses adopted technologies to run and operate. EAI can be categorized into four generations which are traditional integration approach, centralized-based integration approach, standard-based integration approach and cloud-based integration approach. Each section of this paper provides an overview of one generation of EAI and highlights the benefits and challenges associated with it.

II. TRADITIONAL INTEGRATION APPROACH

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). The point-to-point middleware embeds the middleware layer into applications, and it requires excessive alteration on both sides to achieve the integration. As the number of point-to-point integrations increases in an organization, it becomes unmanageable and very difficult to track. Also, the development cost of maintenance and adoption of new changes on existing point-to-point integrations is very high as it requires excessive amount of changes to the existing designs [1][5]. Given these limitations of point-to-point middleware, it led EAI to evolve to a different generation that combined different set of technologies to enhance EAI infrastructure. Fig 1 illustrates the complexity of point-to-point integrations.

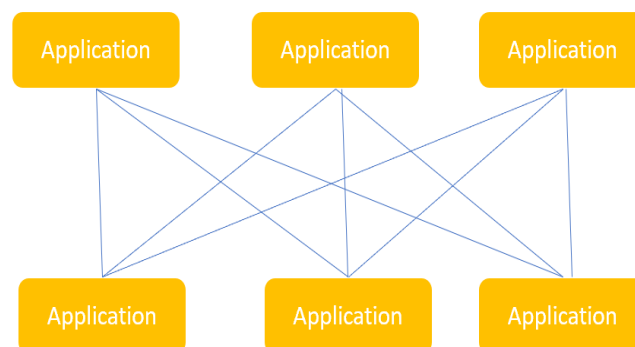


Fig. 1: Point-to-Point Integration Model.

III. CENTRALIZED INTEGRATION APPROACH

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. It uses adapters to connect all systems to a hub to transfer messages and convert data formats to a format that the hub can understand. Sender application publishes messages to a message broker, message broker in turn process the message by applying data transformation. Then, the broker routes the message to the receiving application. Both sender and receiver does not have to be presented during this process and they can stay anonymous to each other [1][2]. Message broker can support different transmission patterns like one to many, many to one, publish/ subscribe and request/ response [3]. Message brokers as shown in Fig. 2 consists of three components: *message transformation* which handles the transformation of message content from the sender format to the receiver format. *Intelligence routing* which involves identifying incoming messages and sending them to the right receiver after processing the message, and the *rule engine* which provides capabilities to control the distribution of messages and allow dynamic routing decision [1].

Message brokers provide more flexibility and less complexity than point-to-point integration approach. In message broker, one connection needs to be in between the application and the broker rather than multiple point-to-point connections [3]. On the other hand, the fact that message broker is central makes it a single point of failure. Furthermore, it is tightly coupled with the applications, and lacks standardization in communication protocols and data format [4].

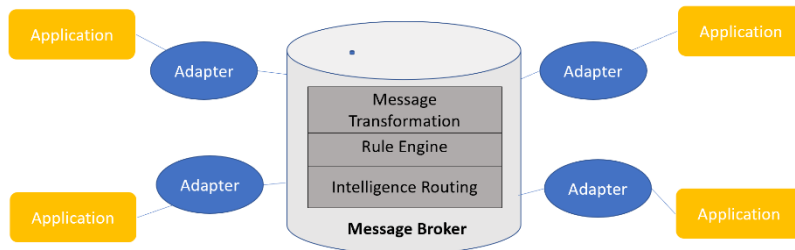


Fig. 2: Centralized Integration Approach.

IV. STANDARD-BASED INTEGRATION APPROACH

EAI introduced the bus architecture to overcome single point of failure issue of the broker model. This model uses the bus concept found in computer systems which distributes the load over a network of nodes instead of a single hub [10]. A typical implementation of the bus concept is what is known as Enterprise Service Bus (ESB). 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 [2][5]. 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 [7].

What Makes ESB unique is the standard-based integration and high-distribution capabilities. Standard-based integration helps businesses in leveraging the existing IT staff instead of outsourcing specialized integration architects. It also, reduces proprietary vendor lock-in and increases the ability to integrate with business partners using standard interface and protocols [11]. The loosely-coupled nature 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 [5].

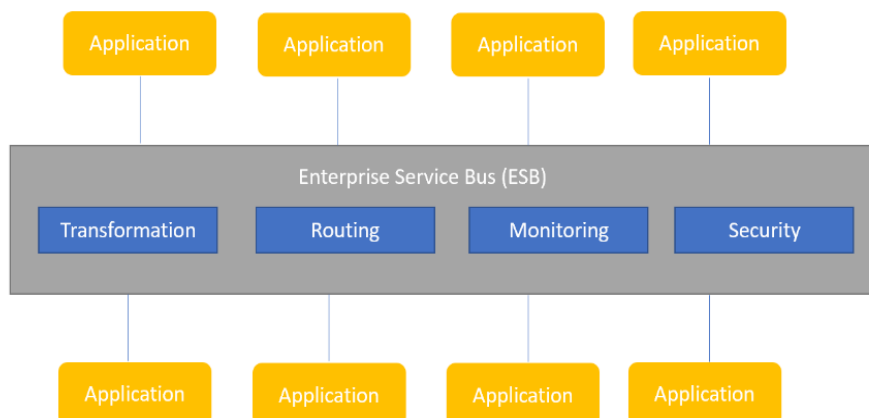


Fig. 3: Enterprise Service Bus (ESB).

V. CLOUD-BASED INTEGRATION APPROACH

In a modern enterprise, IT departments are facing new integration challenges as they are adopting different cloud-based applications and services. Therefore, they are looking for ways to achieve better integration with these cloud-based applications. In today's evolving technologies and as business prefer to use SaaS (Software-as-a-Service) instead of on-premise solutions, integration as a service may be preferable as well. Integration-Platform-as-a-Service (IPaaS) can be considered as the cloud-based equivalent to EAI. It provides cloud-based services to support variety of integration scenarios that includes integration platform services like message transformation, routing, protocol conversion, adapters and development tools [6]. Also, it provides design and runtime for the whole integration. In addition, it serves multiple integration scenarios including cloud to cloud integration, cloud to on-premise, on-premise to on-premise scenarios [6].

The main components of IPaaS are similar to other EAI technologies which include integration processes that explain the logic and flow of how data is exchanged, data mapping between source and target attributes, rebuilt adapters to connect to different applications and functionalities to develop custom adapters and integration process. A key difference between IPaaS and other EAI technologies is the simplicity of developments. Many aspects that require code development with other EAI technologies are simplified with drag and drop mechanism which offers predefined adapters and integration processes template resulting in reduced development time significantly [9]. IPaaS combines the advantages of EAI functionality and SaaS applications such as lower development and maintenance cost and the ease of use. So, it represents a good alternative for organizations that do not want to invest in on-premise integration solutions [6].

When moving to IPaaS, some factors need to be addressed well by organization which include compliance with internal organization policies and country regulations might conflict with IPaaS integration approach. In addition, security and privacy is another important aspect IPaaS as the data will be transferred over the Internet and hosted over external platforms. Another factor is performance of the Internet connection and the which could impact the data exchange between different systems [9].

VI. CURRENT POSITION OF AEI AND FUTURE CHALLENGES.

Looking over the different generation of EAI, we can see that EAI has gone through a long journey and evolved over different generations to overcome many challenges. Different research suggests that ESB is the most useful integration approach so far [4] [10] and IPaaS is relatively new to the industry and further studies need to be conducted to explore it [9]. IPaaS has the potential to be the ideal solution for future EAI that fits the distributed nature of modern IT organizations which involve distributed information systems, networks and platforms. Different research organizations like Granter expect significant growth in IPaaS market size in the next few years.

The different generations of EAI are still being used by IT organizations depending on the integration requirements. Small organizations that own legacy systems may find the traditional approach to satisfy their requirements where other large organization may still look for more sophisticated integration solutions. Choosing the right integration approach is influenced by a number of integration scenarios including the applications that need to be integrated, the protocols that will be used for the integrations, how strong the integration infrastructure need to be, the expected future integration requirements for the enterprise, and the expected growth in the number of integration and applications [10].

Today, EAI is evolving with new emerging technologies in IT like IoT, Blockchain and AI. These technologies are expected to reshape EAI. The nature of IoT services and devices is bringing new challenges to EAI. These aspects include wide geographical distribution of IoT devices and the processing of large amount of data [14]. In addition, the lack of standardized description, poor context awareness and classification of IoT devices service [12]. On the other hand, Blockchain technology has great potential to contribute to enhancing the integration of Business Process Management (BPM) in many industries like healthcare, supply chain, insurance and energy management [13]. Block chain technology can interoperate with BPM in building the trust between different devices and parties, reducing the cost by eliminating the need for intermediates and automate transactions [13].

VII. CONCLUSION

Enterprise Application Integration has evolved to use different technologies over the past three decades to address different integration issues and challenges arising as the technology world is changing. In this paper, we have reviewed the different forms of EAI approaches and we covered the characteristics, architectures and challenges of each approach. EAI future is promising as it is moving towards more standardization which facilitates more seamless interaction between homogenous applications.

REFERENCES

- [1] Linthicum, David S. Enterprise application integration. Addison-Wesley Professional, 2000.
- [2] Anurag Goel, Enterprise Integration EAI vs. SOA vs. ESB, <http://hosteddocs.ittoolbox.com/Enterprise%20Integration%20-%20SOA%20vs%20EAI%20vs%20ESB.pdf>, Retrieved November 2011
- [3] Du, Yan, Wuliang Peng, and Li Zhou. "Enterprise application integration: An overview." 2008 International Symposium on Intelligent Information Technology Application Workshops. IEEE, 2008.
- [4] Soomro, Tariq Rahim, and Abrar Hasnain Awan. "Challenges and future of enterprise application integration." *International Journal of Computer Applications* 42.7 (2012): 42-45.
- [5] Periorellis, P., and P. Watson. "Enterprise Service Bus: An overview." School of Computing Science Technical Report Series (2007).
- [6] Marian, Matei. "iPaaS: Different ways of thinking." *Procedia Economics and Finance* 3 (2012): 1093-1098.
- [7] 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.
- [8] Hérault, Colombe, Gaël Thomas, and Philippe Lalanda. "Mediation and enterprise service bus: A position paper." *Proceedings of the first international workshop on mediation in semantic web services (MEDIATE 2005)*. 2005.
- [9] Ebert, Nico, Kristin Weber, and Stefan Koruna. "Integration platform as a service." *Business & Information Systems Engineering* 59.5 (2017): 375-379.
- [10] Abbasi, Eram, Usman Bin Danish Chawla, and Lareb Hussain. "Trends and Future for Enterprise Integration." *International Journal of Innovation, Management and Technology* 6.1 (2015): 71.
- [11] Chappell, David A. *Enterprise service bus*. " O'Reilly Media, Inc.", 2004.
- [12] Uviase, Onoriode, and Gerald Kotonya. "IoT architectural framework: connection and integration framework for IoT systems." *arXiv preprint arXiv:1803.04780* (2018).
- [13] Viriyasitavat, Wattana, et al. "Blockchain and internet of things for modern business process in digital economy—the state of the art." *IEEE Transactions on Computational Social Systems* 6.6 (2019): 1420-1432.
- [14] Viriyasitavat, Wattana, et al. "New blockchain-based architecture for service interoperations in internet of things." *IEEE Transactions on Computational Social Systems* 6.4 (2019): 739-748.
- [15] Vernadat, François B. "Interoperable enterprise systems: Principles, concepts, and methods." *Annual reviews in Control* 31.1 (2007): 137-145.