Propose a Distributed System Using Middle-ware Based on API Integration

Y. Kalyani 1 , S.Srivathsan 2

Abstract— this paper presents a study of Application Programming Interface and integrating multiple APIs through middleware. Currently we are living in the computer era where we always depend on t echnologies to ful fill our daily needs. In thi s proposed scenario it was analysed about API vs APP and we learned about how to connect multiple APIs through middleware. For this scenario number of apps and API's used to test the integration and use cases.

Index Terms—API, Integration, ESB, Middleware

1 Introduction

In this modern world, technologies are growing faster and trends of businesses are changing frequently. Currently there are millions of apps available in different categories and different fields. Nowadays these apps and IT solutions are not based on single requirement or single problem. Because of this reason it is important to connect multiple apps not only to achieve business goals but also to satisfy the clients/customers.

2 Background of the Study

To get a clear idea of API integration it is essential to know about API vs APP, API Integration, ESB and Connectors.

2.1 API vs APP

APP is a self-contained program or piece of software to fulfill a particular purpose and an application, especially as downloaded by a user to a computer based devices. API is a set of functions and procedures that allow the creation of applications which the features or data of an operating system, application or the other service.

2.2 API Integration

API integration means connecting multiple APIs through a middle ware to achieve multiple business goals. For instance WhatsApp has integrated with Google Maps; if we download whatsapp then we can use that App to share our location to others. Second example would be some websites that are connected with youtube. Next example is, Spotify App integrated with Facebook. In that case we can login Spotify App using our Facebook account. This is one of the latest trends that connecting multiple APIs to use APP in an effective way.

2.3 ESB

An enterprise service bus (ESB) is a software architecture model used for designing and implementing between mutually interacting software applications in a service-oriented architecture (SOA).

2.4 Connectors

Connector is a readymade and convenient tool to reach publicly available APIs, allows us to interact with a third-party product's functionality and data through ESB message flow.

3 Research Objective

Our main research objectives are understanding API and its importance and making use of multiple APIs to create an efficient and convenient tool to fulfill business requirements. Firstly, to get clear understanding on API, documentations of APIs were studied and freely available APIs were tested. Secondly, existing simple scenarios which are integrated using two or more applications were analysed. Finally, an idea for business scenario from open source APIs was arrived at.

4 Methodology

Primarily, our research methodology is divided into four steps. The steps are discussed below in detail. As the first step APIs were analysed. When the analysis of APIs was started, some questions were raised to ourselves and answers to those questions were sought. This kind of method made us to analyse the APIs in an easy way. Following are the Questions:

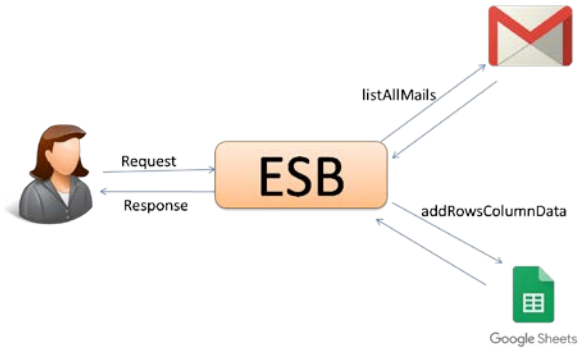
- What type of API is it?
- What kind of data format is used in the request (JSON, HTML or XML)?
- What type of authorization is used/ required (Basic, Oauth 1.0, Oauth 2.0)?
- How many functions and what functions are available for us as a developer?

Once answers were found to all the above questions, suitable APIs for our scenario was chosen according to the answers. As the third step, when more than two APIs were

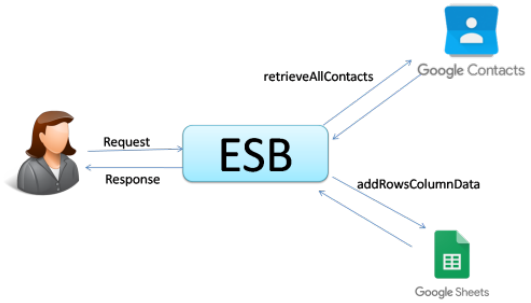
needed existing connectors from available stores such as Mulesoft and WSO2 were used. Lastly, multiple APIs through ESB were connected

5 Implementation

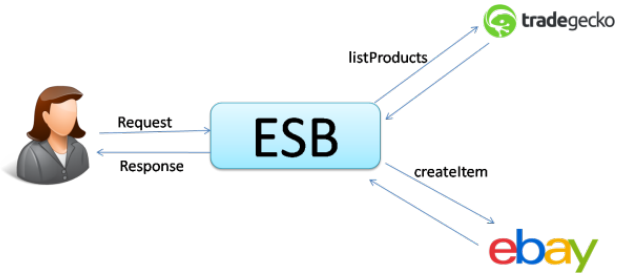
A simple scenario using Gmail and Spreadsheet connectors was tested. The use case is to get all mail details from the Gmail using *listAllMails* operation, and add them into the Spreadsheet using the *addRowsColumnsData* operation.



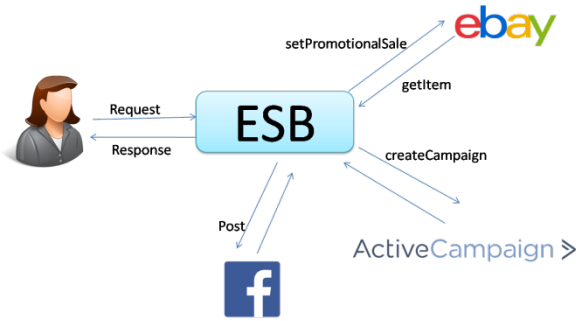
The second simple use case is using GoogleContacts and Spreadsheet, get the all contact details from GoogleContacts using *retrieveAllContacts* operation, and add them into the Spreadsheet using the *addRowsColumnsData* operation.



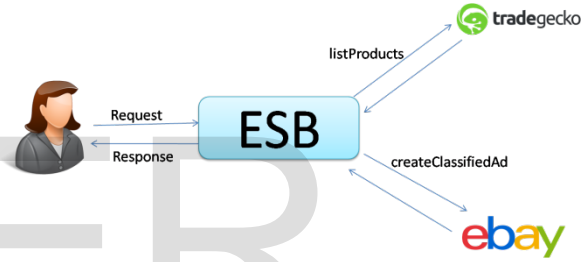
The third complex use case is using multiple connectors such as Ebay, Facebook, Tradegecko and ActiveCampaign. This use case was divided into three sub goals. Creating Ebay items by retrieving product details from TradeGeckowas the first goal.



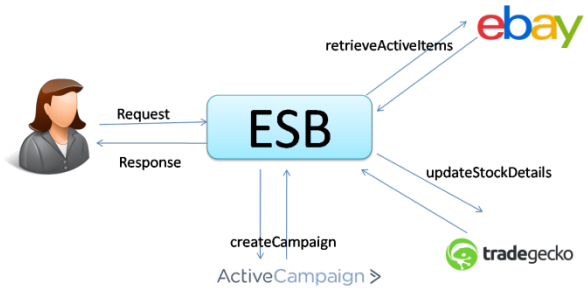
Second task of this use case was, converting already existing Ebay item as a promotional sale item. It was followed by creating a campaign in ActiveCampaign for promoted item in ebay and creating a facebook post on seller's wall with newly promoted Ebay items details.



The third goal was, creating Ebay classified ad listing by retrieving product details from the TradeGecko.



The last sub task was, updating TradeGecko stock adjustments and adding buyers contact details into ActiveCampaign subscriber list by retrieving running date Ebay sold items.



6 Conclusion

In this paper, we have discussed about integration of multiple APIs through ESB. Both business people and customers can get more advantages through connecting multiple APIs. For instance, some of the pros are, saving time, using applications in an effective manner and multiple usages from single operation. If we could implement this kind of integration using multiple Apps, this would make people to get more and more benefits.

7 Future work

We plan to identity and implement business scenarios using freely available APIs in future. In addition, plan to test the usablility of exisiting freely available APIs and propose distributed systems according to that.

References

[1] M. A. Razzaque, A. Packard, M. M. Jevric, A. Palade, Middleware for Internet of Things: A Survey in IEEE Internet of Things Journal, Volume: 3, Issue: 1, Feb. 2016

[2] J. Al-Jaroodi, N. Mohamed , Hong Jiang, Distributed systems middleware architecture from a software engineering perspective, in IEEE International Conference on Information Reuse and Integration, pp. 572 – 579, 2003