PURCHASE MANAGEMENT SYSTEM

Alka Singh, Pooja Sinha, Mrs. Gunjan Agarwal
Student, Information Technology Department, Raj Kumar Goel Institute of Technology, Ghaziabad, Uttar Pradesh, India
Assistant Professor, Information Technology Department, Raj Kumar Goel Institute of Technology, Ghaziabad, Uttar Pradesh, India

ABSTRACT

This paper identifies the demonstration of the implementation of order management for the existing customers using the MuleSoft approach to integrate, an API-led strategy for purchasing management system was developed to automate an existing manual system. The project proposes employing a layer of APIs to integrate data from many systems, enabling the creation of a structured application network that connects applications, data , and devices via reusable APIs. The usefulness of making data available via API firmly supports the faster and easier data migration, improving the data quality review and cleanup. In terms of service deliverables APIs provide greater flexibility.

The PURCHASE MANAGEMENT SYSTEM initiates developing a RESTful Web Service or a REST API consuming JSON payload including the entire cloud-based environment system for a company. Customers and employees require data-rich, enjoyable digital experiences across a wide range of devices, from smart watches to desktop computers. The systems must be connected to one other and data must move between them in order to deliver these experiences (integration). Essentially, the project explains how to manage for good performance and better client services. It is an approach to enable the vast spread multinational companies to manage the tons of data in the IT industries for more convenient functioning and scalable growth in this fast-paced industrial environment.

Keyword : - *PURCHASE MANAGEMENT SYSTEM, REST API, Integration, MuleSoft and Layer.*

---

1. INTRODUCTION

The 'Cloud based REST API Web services an advanced approach' aims to give solutions for developing and transferring information in a simple and effective manner in the digital age, as well as reducing human pressure and time in the workplace. An API, or application programming interface, allows data to be sent and received across software program systems, or platforms to support shop collections, digital initiatives, and digital projects at external partner institutions. It offers services such as digitization of analogue items, metadata management, digital preservation, and digital collection discovery and access.

"Purchase Management System" is a rest API web service with JSON payload built for all operating systems integrated with cloud environments, with the goal of assisting users in maintaining and organizing their virtual purchasing. Companies can use the 'Online Purchase Web service' to automate tasks by linking their program and databases with existing industry applications. The purchase system's report creation features aids in gaining a better understanding of the numerous things brought by members, making it easier for users to obtain the product. It's built on RESTful technology, an architectural style, and a communication strategy that's common in web services development.

2. METHODOLOGY

To deliver a solution to the customer, we applied an API first approach, which included building API spec for several sorts of APIs (system, process, and experience) with appropriate responses to the users. Additionally, we have considered best practices for naming conventions as well as strong error handling. The 'Cloud based REST API Web services an advanced approach' aims to give solutions for developing and transferring information in a simple

and effective manner in the digital age, as well as reducing human pressure and time in the workplace. A RESTful API is a software interface that allows you to use HTTP queries to GET, PUT, POST, and DELETE data. It's based on RESTful technology, a common architectural style, and a communication approach in web services development.

2.1 Working Approach:

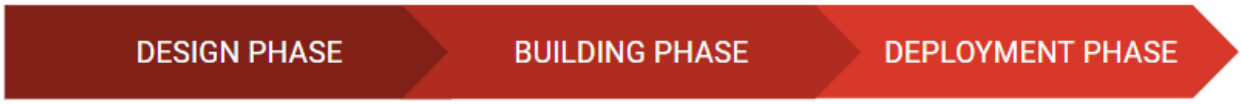

Fig. 1 View of the strategy planned for working approach

During the design process, we used the API design center to code in RAML languages and tie them to separate RAML files. All of the aspects of error management principles to security-based encoding decoding features are included in the full coded segment.

In an API-led connection approach, APIs are organized into three layers: system, process, and experience. We have splitted the entire task into 3 layered approaches aiming to adopt a methodical way to connect data through applications through reusable APIs.

2.2 Working Flowchart

The flow chart shown below depicts the entire operation of the implementation process. It represents the working strategy of the layered approach for interconnectivity from the database to the customer end.

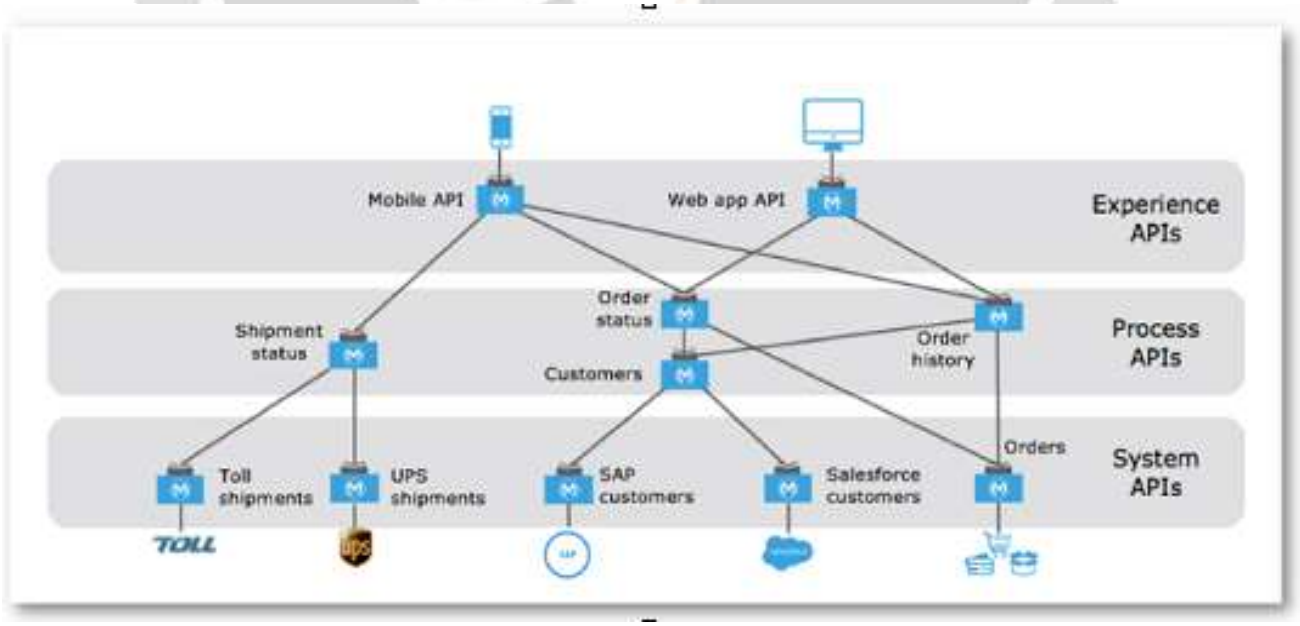

Fig.2 Flow chart for the API-Led architecture

3. BRIDGING THE THEORETICAL AND PRACTICAL DIVIDE

We have pointed out that other than the conventional approach for dealing with the order and purchase management using RESTful web service can accommodate many sorts of calls, return diverse data formats, and even change fundamentally with the correct implementation of hypermedia flexible industrial operations regarding the delivery services. One of the most appealing features of REST APIs is that they are based on the HTTP standard, which means they are format-agnostic, allowing you to use XML, JSON, HTML, and other formats.

3.1 Implementation

We have divided our project into 3 layers using API Led Connectivity approach. It demonstrates the three layer API design which is easily accessible by organizations to get most out of the systems to facilitate the entire organization to comparatively invest less development and maintenance time on fixing issues.

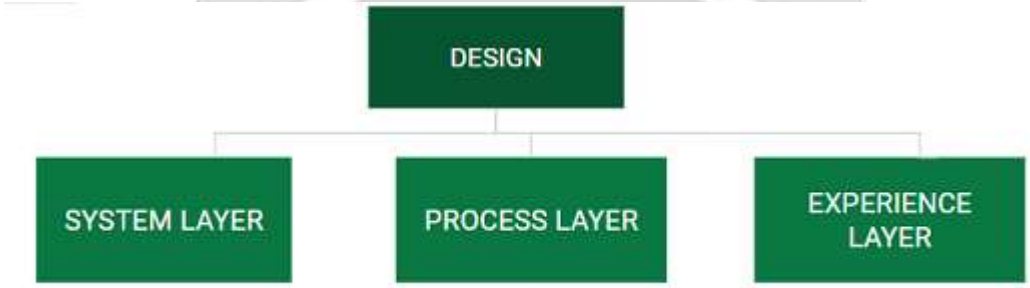

Fig.3 Representation of the layered approach implemented

3.2 Results and Discussion

The process initiates with the system layer which is entirely connected with the database. This is the top layer fulfilling the respective use cases, accessing all the underlying records with a prior connection with the database.

In this layer there are 6 different flows to fulfill the proper specifications:

- Customer Flow
- Customer Flow By Id
- Product Flow
- Product Flow By Id
- Order Flow
- Order Flow By Id

The respective flows operate according to the request made via port address using various end points and fetch the details of particular customers with their respective ids , products information and placed orders. The contract

RAML/WSDL for describing how to interface with the domain is defined by a System API. A System API for a product domain, for example, could include resources with methods such as GET, POST, PUT, and DELETE, as well as the associated schemas (XSD, JSON) and responses (200, 400, 500, etc).

Following this the next layer Process Layer comes across to operate. Within the process layer the entire business logic is being implemented. Data collection, division, filtering, and routing are all part of the orchestration process. The basic goal of Process APIs is to isolate the business process from the source systems (System APIs) from which the data is derived. In a purchase order procedure, for example, the customer must engage with multiple organizational domains.

Experience Layer is the topmost layer in this layered structured approach. Data is consumed in a variety of formats by a variety of consumers. For example, the purchase order API (Process Layer) exposes data in JSON format, while a consumer application only takes CSV or XML format, or vice versa. In some situations, we may be required to provide filtered data for various consumer applications. The Experience Layer implements all of the transformation logic, and the implementations are available via Experience APIs.

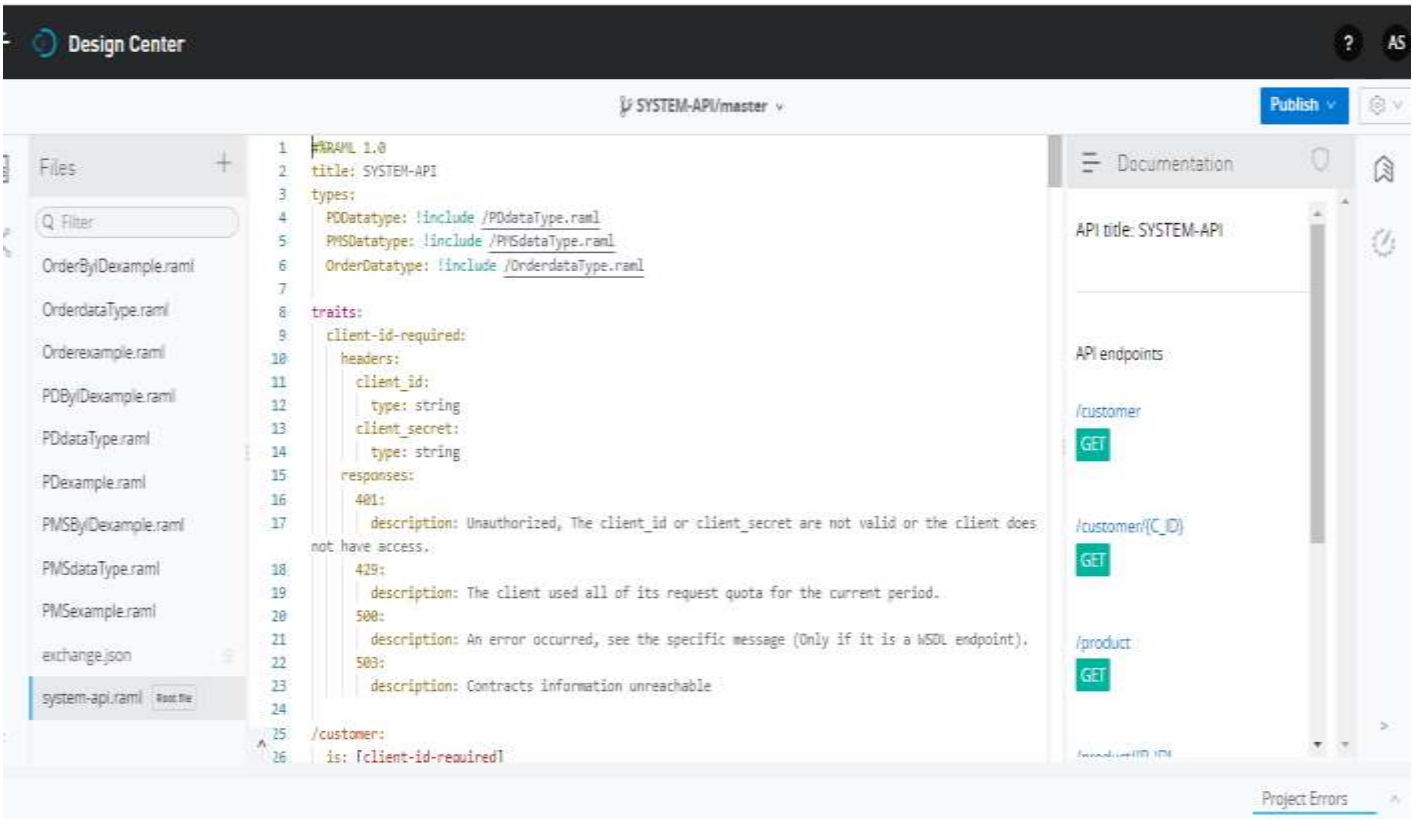

Fig. 4 Screenshot of the code implemented for System API

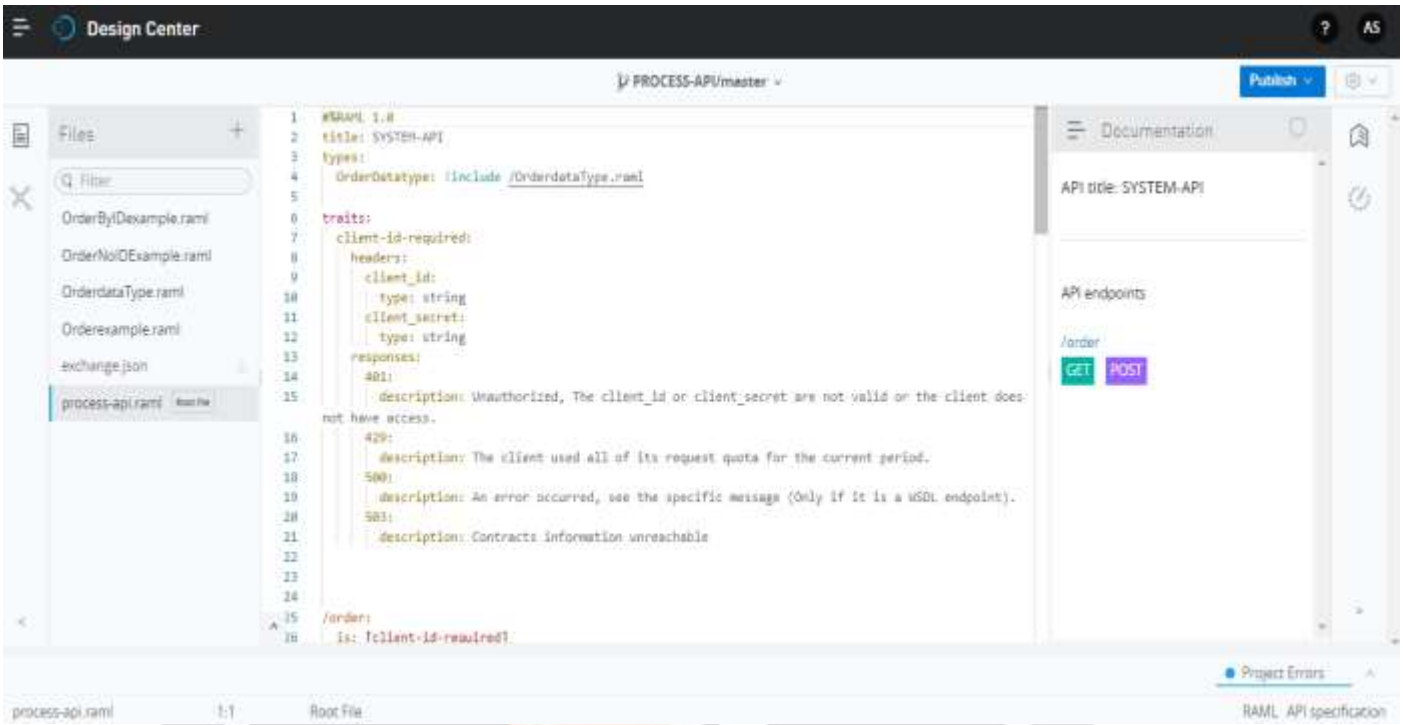

Fig. 5 Screenshot of the code implemented for Process API

4. CONCLUSION

In this paper, we have provided an overview of the purchase management system, including a classification of the approach used in every step of implementation. With the research and usefulness of API we came up with this idea to apply an API led approach to bring revolution and can utilize the real strength of this cloud technology. Because of its simplicity and familiarity, the REST API is simple to understand and learn. Being able to arrange complicated program and make it easy to access resources is possible with REST API.

5. REFERENCES

1. MuleSoft and its components https://developer.mulesoft.com/tutorials-and-howtos/getting-started/hello-mule/

2. Layered API approach - https://blogs.mulesoft.com/learn-apis/api-led-connectivity/what-is-api-led-connectivity/

3. Anypoint studio : Eclipse based platform

https://www.infomentum.com/mulesoft-plain-language

4.https://www.tutorialspoint.com/mulesoft/mulesoft_dataweave_language.htm#:~:text=DataWeave%20is%20basically%20a%20MuleSoft,strongly%20integrated%20with%20Mule%20runtime.

5. https://docs.mulesoft.com/mule-runtime/3.9/mule-expression-language-mel

6. Testing with mule https://www.tutorialspoint.com/mulesoft/mulesoft_testing_with_munit.htm

7. SQL Server 2000 Bible- By Paul Nielsen

8. SQL Server 2000 Unleashed

9.https://blogs.mulesoft.com/api-integration/how-to-future-proof-digital-transformation-with-an-api-strategy/

10. https://blogs.mulesoft.com/dev/api-dev/what-is-api-led-connectivity/