Application Programming Interface with a Case Study of SOA

1st Sumathi Pawar
Information Science & Engineering
Nitte(DU)
NMAMIT, Nitte
Karkala, Karnataka, India
pawarsumathi@gmail.com

2nd Suma K
Computer Science & Engineering
MITE, Mijar, MoodBidri
Karnataka, India
Sumaknair07@gmail.com

3rd Geetha Laxmi
Information Science & Engineering
Canara Engineering College
Bantwal, Karnataka, India
geethalaxmi@gmail.com

2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS) | 979-8-3503-1545-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIICS59993.2023.10421593

***Abstract*—A set of ways, procedures and tools used in programming to develop software operations is known as an Application programming interface (API). An API exposes the procedures, inputs, issues and underpinning types of a software element. An API separates features from their applications, enabling definitions and functionalities to evolve without degrading the interface. By providing all of the required building blocks, a good API makes developing a software easy. In this paper, APIs of Service Oriented Architecture (SOA) are implemented and discussed along with SOAP, JSON-RPC, XML-RPC, REST. APIs for securing Web services also focused by coding examples. A Web service request message and response messages are resulted through SOAP envelop and discussed in this paper.**

Keywords—API, SOA, REST, Open Source

I. INTRODUCTION

An API describes the linking between distinct systems or software programmes (API). It is a form of software interface that enables interaction with various programmes. The API specification refers to the document or standard that outlines the construction and use of associations and interfaces. API can be created or existing APIs can be called.

Contrary to user interfaces, application programming interfaces allow software or computers to communicate with one another. Software developers may use it. Numerous components that serve as developer tools or services make up an API. A programme or programmer is said to be invoking that piece of the API when they use one of these components. API calls are referred to by terms like subprocesses, procedures, requests, and endpoints. These calls are described in the API standard, along with instructions on how to use or implement them.

The core workings of the system should be hidden, only the parts that programmers would use should be exposed, and those sections should remain constant even if inside workings varies. Programming interface, might be tailored for a specific set of structures or it might be a widely accepted standard that allows a selected number of systems to connect with one another.

Through an application programming interface API, businesses can also acquire the information and functionalities of their application available to internal departments, business partners and third parties.

A specified interface allows products and services to interact and take advantage of one other's stats and abilities. It is not necessary for developers to understand how the API functions, all they need to know is how to use this interface to link with some different goods and services. In recent years, the use of APIs has grown to the point that it would be challenging to create many of the popular and in-demand Internet products without them.

The tenure API is frequently used for referring web API that enables message among computers connected via the Internet. Software libraries, hardware, operating systems, and programming languages all contain APIs. Although APIs have been present since the 1940s, it wasn't until the 1960s and 1970s that the phrase began to gain popularity. Microservices are services that are loosely connected as a result of recent improvements in API usage. [1]

A programming interface (API) is a set of guidelines that define how a computer and software program communicate with each other. Among a web server and an application, APIs serve as a data transfer intermediate.

The API works as follows:

- Client applications make API calls, also known as queries, to get information. API URIs (Uniform Resource Identifiers) are used by applications to send requests, including request verbs, headers and possibly request bodies to a web server.
- Upon receiving an authentic request, API will call the external software or web server.
- The server returns the data requested by the API.
- Data returned to the genuine requesting system through the API.

All queries and responses are handled through an API, even though the data transit relies on the web service. User interfaces are created for human usage, whereas APIs are intended to be utilized by computers or applications.

Since they allow for the separation of functions between two units, APIs are meant to provide safety. The API endpoint separates the programme that receives data from the system that provides for its function as a mediator. API calls usually contain authorization credentials and gateway of API can impose access restrictions to limit security risks. This reduces the likelihood of server attacks. The data transfer is further secured with the usage of cookies, HTTP headers and query string parameters.

As an example, think about a provider of payment processing API. Users can enter their payment card information on the application interface of an online retailer provides security against hacker attacks.

A. Features Of Api

- An application programming interface (API) is software program that permits distinctive types of applications to speak with each other.

- An application programming interface allows data and functionality to be readily exchanged between apps.
- The API is described as a linking between two systems.
- Data monetization is aided via application programming interfaces (APIs).
- An application programming interface is used to improve collaboration. [2]

II. Literature Review

An application's communication interface is referred to as an API. Modern API, on the other hand, offers unique properties that have increased its value. These are detailed further down[3].

- They adhere to requirements that are developer-friendly, accessible, and clear.
- They are aimed at a specific audience, such as mobile developers, and hence can be consumed by them. These APIs are designed in such a way that they should be able to meet servicing and lifetime expectations.
- Because they are more standardized, they are extra disciplined in phrases of safety and performance.
- They have a software development lifecycle (SDLC) that comprises testing, management, design, construction, and versioning. Rather than acting as code, they act as a product. Modern APIs are also extremely extensively documented.

The organization of this paper is in section 2, related work regarding APIs is discussed, in section 3 implemented examples of SOA APIs are given, in section 4, APIs uses are discussed.

APIs are useful tools that will continue to gain in popularity as programmers develop new applications that will appeal to a wider audience. API providers are crucial in providing event-driven APIs to API consumers[4].

Network layer of the TCP/IP protocol will be fully rewritten in terms of improving packet transformation in low- and fluctuating-coverage networks. With the advent of HTTP3, this is now possible. App developers will be able to minimize their time to market as a result of being able to incorporate extra APIs into their mobile apps. As a result, mobile app performance will increase even if coverage is patchy. Access to the APIs will be quicker[5].

Providers of API are critical in providing event-driven APIs to API consumers. It signifies that the logic of the business has shifted to the API customer from the API provider[6]. When booking a Uber ride, for example, the app assigns the driver and provides the data immediately.

A. Types of Api

APIs for the Web expose information and functionality of applications over the Web and are the most popular application programming interface today[7].

Web APIs are divided into four categories:

1) Open APIs : These are an open source HTTP-based application programming interface. Also known as the Public API because it contains predefined API endpoints and request and response forms. This is also known as open APIs because it's accessible to the whole public. Data and services can be accessed by third-party users through open APIs. It's a free and open-source HTTP-based application programming interface.

2) Partner APIs: Business partners access these APIs. To contact these APIs in mode of self-service, the bulk of developers use a public API developer site. So as to use partner APIs, they need to still undergo an induction program and receive access credentials. It' a kind of API that needs further rights or licenses from the developer in order to utilize. The Partner APIs aren't accessible to the overall public.

3) Internal APIs: These APIs aren't available to the general public. The purpose of the private APIs, which are not open to the public, is to boost productivity and coordination among the numerous development teams employed by the company. This kind of API is commonly referred to as a private API because it is only offered by an internal system and will not be shared.

4) Composite APIs: APIs that combines different data into one. With the help of these services, developers can navigate across several endpoints at once. When a single operation requires data from multiple sources, composite APIs come in quite handy in a microservices design. It is a kind of API that mixes information and services from numerous sources. Increasing performance, quickening the execution process and enhancing observer performance in web interfaces are the main reasons to invest in Composites APIs.

5) Microservices Architecture: This is a distinctive architectural style in which a complex application is divided into smaller, independent parts. It is easier to implement, administer, and scale the programme, if it is implemented as a group of independent services. Because it enables developers to concentrate on one aspect while another is being constructed, this technique has gained popularity in the age of cloud computing.

III. Taxanomy Of The Research

This research is focused on implementing APIs of SOA with WSDL helper classes[8].

In practice, these API protocols allow for the exchange of standardized data,

- SOAP
- XML-RPC
- JSON-RPC
- REST

A. Soap (Simple Object Access Protocol)

SOAP APIs exchange data between Web services and this is the abbreviation for Simple Object Access Protocol. It's well-known for being expandable, neutral, and self-contained, and it's built to be just that.

Some of the SOAP standards are as follows[9]:

- The Processing Model

- Extensible Model
- Protocol Binding Rules
- Message Construct

A web service could be a program which will be accessed by a URL. As a result, each web service is in effect, an API since it discloses an application' info and capabilities. A web service may not always be the same as an API[10].

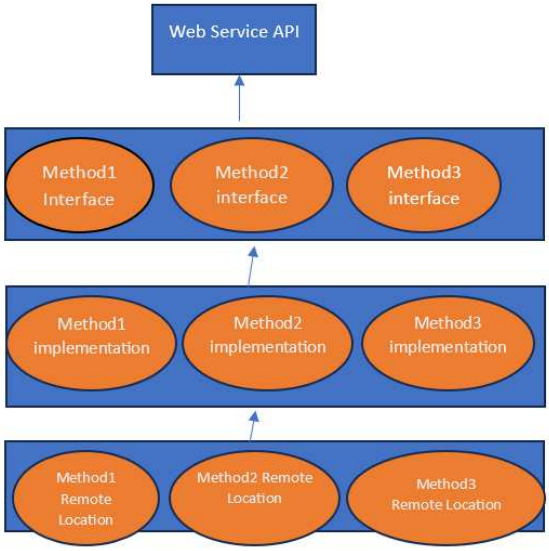

Fig. 1. Architecture of SOA Web Service APIs

SOA is a technique of software system development within which functions are split and created accessible as separate network services. In SOA implementations, internet services are often used for creating the practical codes accessed via typical protocols of communication. Developers will produce these programs from the bottom up and will provide service interfaces.

Some examples of Webservice APIS are given here.

WSDLHelper class API and its methods

WSDLHelper wsObject=new WSDLHelper()

Definition defObj= wsObject.getDefinition(file);

portTypes= wsObject.getPortTypes(defObj);

opsObj= wsObject.GetOperations(portTypes);

In the above APIs WSDLHelper API is a class having methods of WSDL processing. getDefinition() is a method which take entire WSDL file as parameter. Method getPortTypes() returns the different ports available in the webservice and GetOperations() is a method that returns various functions available in the web services.

B. *Xml-Rpc*

This is an XML-based protocol that, unlike SOAP, delivers data in a branded XML format. Despite being elder than SOAP, XML-RPC is a much easier and more effective at measuring information.[11].

An example of API belong to this is

xmlrpc.client (EndpointUrl, process, parameters, responseFormat, EndFunction (error, datas)

In this API, an URL is passed as first parameter, function name is passed through second parameter, arguments of the functions are passes as an array of parameters. The fourth parameter of this method is format of the response and fifth parameter is an error message if the requested function is not executed.

An instance of all these parameters are given here.

const urlEndpoint = " https://free-web-services.com/web-services/geo/weather/rpc2";

const process = "weather.getCity";

const parameters = [5]; //an array containing one element, the number 5

const responseFormat = "xml"

C. *Json-Rpc*

JSON is employed to convey data. Each protocols are simple to follow. Calls could have many parameters, but they solely expect one outcome.

An example of JSON-RPC request & response is given here.

```
curl --data '{"method":"web3_clientVersion
","parameters":[],"identifier":1,"jsonrpc":"2.0"}' -H
"Content-Type: application/json" -X POST
localhost:7345
"response": "OpenEthereum//v5.0.1-stable-8ca7089-
20200601/x86_64-unknown-linux-gnu/rustc1.44.1"
```

In above API, method name indicates the process requested by the client and parameters are arguments of method. Response returned is result of requested operation.

D. *Rest API*

The two standards are commonly viewed as rivals even though SOAP protocols are used to create RESTful APIs. This is the most well-known is the REST API design. These are a few of the requirements for a REST API that the API must adhere to i.e. Statelessness, Client-Server architecture, Layered Systems and Cache Capability. As an example of REST API:

requestMessage['X-DreamFactory-API-Key'] = 'M_API_KEY'

responseMessage = http.request(requestMessage)

In this API, key of the API is passed as M-API-KEY.

E. *Layer of Security*

Instead of exposing entire server, only the essential data is communicated, and only little data packets will be used to interact between them. APIs have become so popular that they are now being utilized to make money. Because APIs produce revenue, they are an important part of any business. The API economy is the term for the marketplace for security of APIs [12].

Security is crucial when developing an API with a public interface. Common dangers include SQL injection, denial-of-service attacks, shoddy authentication, and the release of private information. If proper security precautions are not taken, intruders may acquire access to information they shouldn't and even have the authority to modify server's configuration. Common security procedures include needing One common security measure is to upgrade the Robots

exclusion standard, and another is to need an API key to access the service. An issued API key is required by the majority of public-facing API services, and they may also refuse to fulfil the request if it don't include the key[13].

Some examples of Security APIs[18]

```
Encryptor eObj=new Encryptor (secApi.getAsDocument(), key, AlgorithmType. TRIPLEDES);
Decryptor dObj =new Decryptor(secApi .getAsDocument(), key, new XPath("//*[name()='xenc:EncryptedData']"));
```

The above APIs uses "TrippleDES" as security algorithm.

The above APIs uses "TrippleDES" as security algorithm.

IV. Results

Software requirements for this implementation included WSDLHelper classes and Rest API classes. Using these APIs remote web services are invoked by passing parameters and results are received back. A SOAP request and response message of Web service request is given in figure 2 and figure 3.

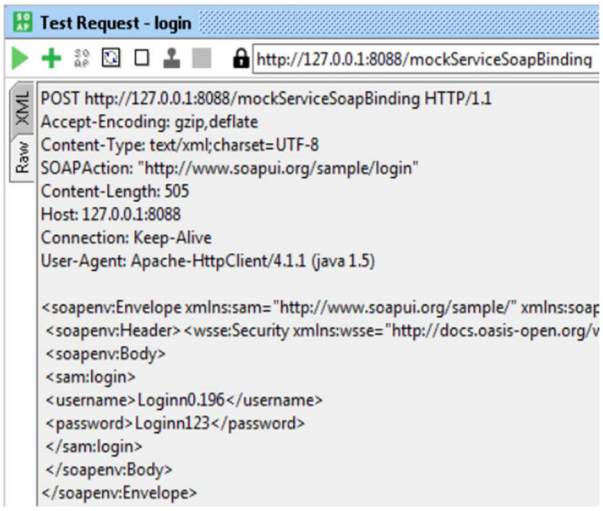

Fig. 2. SOAP request message

Fig. 3. SOAP Response message

V. Discussion

API can help with both managing existing tools and developing new ones[14]. Some of the most significant advantages of APIs are as follows:

- Improved Collaboration
- Easier Innovation
- Data Monetization
- Added Security

A. Improved Collaboration

1200 cloud applications are networked together through APIs. Businesses can automate processes and improve employee collaboration by using this connectivity. If APIs weren't available, many firms would become isolated and suffer with data silos, which would have a negative impact on productivity and performance.

Even though different libraries share the same programming interface, it can be seen different implementations of a single API.

Due to the independence of the API and its implementation, programmes developed in one language may also be able to use a library written in a different language[15]. For instance, Scala developers can utilize any Java API because Scala and Java both compile to the same bytecode. The manner that APIs are used varies depending on the programming language. According to several surveys, the majority of API-using applications only make use of a small portion of the API's capability. An API's usage is influenced by its popularity and number of users. A word used to describe APIs is "language bindings." Language bindings allow libraries or services built in one language to be used while functioning in any other language by translating the performance of one language to a specified interface in another language.

B. Easier Innovation

With APIs, businesses can have interaction with new partners, deliver new services to present day users and sooner or later obtain access to new markets, generate extensive revenue and speed up digital transformation. For example, Stripe started out as a seven-line API. Since then, the agency has partnered with a number of considerable organizations around the world, multiplied its mortgage and company card products and these days obtained a $36 billion estimate (link resides exterior of IBM).

C. Data monetization and added security

Selling access to these assets can be used to monetize APIs that give users access to important digital goods (this is regarded as API Economics). In just 10 months, AccuWeather (link leaves IBM) has established a self-service platform for programmers to sell a range of API packages, hired 24,000 programmers, sold 11,000 API keys and built a vibrant community.

Between the server and the data, there is a layer of protection provided by the API. Through tokens, TLS, signatures APIs can be secured. A software framework can also be related with an API. A framework can be built on many libraries that implement several APIs, but communication to the framework's functionality is mediated by adding new classes to the framework's content, unlike with traditional API usage.

Moreover, inversion of control or a similar method can transfer control of the entire flow of control from the caller to the frameworks.

An API is usually coupled with a software library and the API specifies expected behavior, while the library is responsible for imposing these rules into action.

It is vital in today's environment to design APIs that are fit for purpose. Cloud native app development necessitates the use of a microservices application architecture. APIs are used

to exchange information with several other customers, such as clients. The idea of a service-oriented architecture underpins microservices architecture (SOA).

A standard message mechanism similar to RESTful APIs is utilized to provide open communication throughout an operating system. There is no need for additional data translation or integration layers in this system. Any service or feature can also be removed, replaced, or improved without impacting the other services. By maximizing cloud resources, this lightweight dynamic APIs enables for improved API testing, performance, and scalability[16].

VI. Conclusion

An API might make it much easier to programme complex software. An API could make integrating new features into existing apps more easier. Different programmes can share data through an API, enabling them to combine and enhance their functionality. Over the past decade or so, APIs have become increasingly popular among scholars and practitioners[17]. One cause for this is that they are increasingly serving as a major force behind most breakthroughs that are occurring. Therefore, it is essential to comprehend and value their significance to both practice and research. For businesses, a lack of understanding about API design and integration can be quite difficult. This article presents a thorough analysis of API research articles from 2010 to 2018. It gives a summary of the current research with implemented examples of SOA APIs. It identifies areas that need more study and more consideration. The results indicate that the key themes in API research are related to the design and usefulness, which both fall under the classification system's technological domain.

References

[1] M. E. Fayad and D. C. Schmidt, “Object-Oriented Application Frameworks,” Communications of the ACM, Vol. 40, No. 10, 1997, pp. 32-38. doi:10.1145/262793.262798.

[2] Richard Boateng, Joshua Ofoeda, John Effah, Application Programming Interface (API) Research: A Review of the Past to Inform the Future, International Journal of Enterprise Information SystemsVolume 15,Issue 3,Jul 2019, pp 76–95

[3] A. S. and V. R. , "API Features Individualizing of Web Services:REST and SOAP," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, July 2019.

[4] I. O.-A. C. O. and H. O. , "Cloud and Application Programming Interface – Issues and Developments," in World Congress on Engineering, London, U.K, 2018.

[5] A. S. and V. R. , "API Features Individualizing of Web Services:REST and SOAP," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, July 2019.

[6] J. O. R. B. and J. E. , "Application Programming Interface (API) Research: A Review of the Past to Inform the Future.," International Journal of Enterprise Information, pp. 76 - 95, 2019.

[7] Gray Brooks , " Benefits of APIs" https://digital.gov/2013/03/12/benefits-of-apis/ , last accessed date 14/9/2022

[8] J. K. P. F. and R. B. , "A History and Future of Web APIs," it - Information Technology, January 2014.

[9] Meng, Michael et al. “How developers use API documentation: an observation study.” Communication Design Quarterly (2019), Volume 48, issue 3.

[10] A. S. and V. R. , "API Features Individualizing of Web Services:REST and SOAP," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, July 2019.

[11] I. O.-A. C. O. and H. O. , "Cloud and Application Programming Interface – Issues and Developments," in World Congress on Engineering, London, U.K, 2018.

[12] M. F. and D. C. S. , "Object-Oriented Application Frameworks," Communications of the ACM, 10 October 1997

[13] M. A. and M. P. , "The Future of API Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities," Journal of International Technology & Information Management , vol. 29, no. 3, pp. 25-45, 2020.

[14] I. O.-A. C. O. and H. O. , "Cloud and Application Programming Interface – Issues and Developments," in World Congress on Engineering, London, U.K, 2018.

[15] M. M. S. S. and A. S. , "Application Programming Interface Documentation: What Do Software Developers Want?," Journal of Technical Writing and Communication, 48(3):295–330, 2018.

[16] David Berlind, "The advantage of APIs," Dec 03, 2015, https://www.programmableweb.com /news/what-are-benefits-apis/analysis /2015/12/03

[17] J. K. P. F. and R. B. , "A History and Future of Web APIs," it - Information Technology, January 2014.

[18] M. M. S. S. and A. S. , "Application Programming Interface Documentation: What Do Software Developers Want?," Journal of Technical Writing and Communication, pp. 1 - 36, 2017.

[19] Sumathi, Dr. Niranjan N Chiplunkar “Open Source APIs for Processing the XML Result of Web Services”, International Conference on Advances in Computing, Communications and Informatics (ICACCI) IEEE, MIT – Manipal, September 14-16 2017.