

A Probabilistic and Deterministic Structured Approach of SoapUI in Microservices Architecture

Deepali Jain¹ Dr. Mehul P Barot² Dr. Lokesh P Gagnani³ Sonal Bhavsar⁴ Pravina Parmar⁵

^{1,4,5}Assistant Professor ²Professor and Head ³Associate Professor

^{1,2,3,4,5}Department of Information Technology & Computer Engineering

^{1,2,3,4,5}LDRP Institute of Technology and Research, Gandhinagar, Gujarat, India

Abstract— Software applications are created daily in the rapidly evolving landscape of software development and are created on a daily development basis which require API testing so choosing the right tool is paramount. This research paper presents a deterministic structured approach of one of the most prominent an open-source API testing tool SoapUI, in the context of the Microservices Architecture. This research paper aim is to provide -in depth analysis with the functional voluntaries analysis of SoapUI by encompassing its features, use cases and performance metrics in respect to Ecommerce based domains. In this systematic proposed approach, a step-by- step flow of integration of SoapUI with a Microservices Architecture is Proposed with the graphical evaluation and their explanation which covers all the aspects regarding to its functional testing, mocking the services, security and load testing by importing the endpoints of the Microservices Architecture. The integration of SoapUI in Microservices architecture by integrate the CI/CD pipeline increases the throughput in respect to the autonomous Ecosystem. The examination of various aspects of SoapUI covering various aspects such as features, capabilities, practical use cases, and performance metrics. Therefore, the main objective of this research is to represent a new structured approach using most reliable and automated testing tool which is SoapUI (an open-source API testing tool) in the framework of Microservices Architecture with the practical usability and characterizable solutions in various testing aspects of Web applications.

Key words: SoapUI, Microservices Architecture, Web Service Testing Tools, Testing Web Service, API Testing, Web Application Testing, Open-Source Testing Tools

I. INTRODUCTION

A. Background

Testing API is a very important also a critical phase in software development because of ensuring the reliability and robustness of web applications that is totally relay on API.[3] Emerging and development of new technologies and various trends, the selection of an appropriate API testing tool has become more challenging so here comes an open-source API testing tool i.e., SoapUI which provides user friendly interface by leveraging various features and capabilities.[4][5][25]

B. Objectives

This paper aims to analyze and represent a SoapUI which is an open- source leading API testing tool, in light of the latest trends and technologies and well known farmwork as Microservices Architecture.

SoapUI empowers the users to perform all-inclusive testing parameters and validation of Microservices also ensuring they meet performance, functional requirements and security within the Microservices Architecture.

II. MICROSERVICES ARCHITECTURE

A. Introduction

The term Microservices defines itself as small tiny services communicating with each other.[1] It is an Architectural style to build the applications and these applications are the basically software applications and by this you can structure an application as a collection of small autonomous service models around the business domain and as business domains are basically Ecommerce based domains so we are basically taking care about the required functionalities, features and testing tools.[1][2][10]

Now before micro services came into the picture the software industries were using the monolithic architecture in which the complete application stored in the single bucket.[11]

B. Literature Survey

To understand the all-historical contextual points of Microservices Architecture a tabular comparison between Microservices Architecture with the Monolithic is given below:

Characteristic	Monolithic Architecture	Microservices Architecture
Definition	It is a single, self-contained unit where the components are tightly.	It is an architectural style that structures an application as a collection of loosely coupled services.
Size of Module	Large	Small
Deployment	It is deployed as a single unit.	In these individual services can be deployed independently.
Scaling	Horizontal scaling needs replicating the entire application.	Individual services can be scaled independently as desired.

Complexity	Easier to initially develop but can become complex over time.	More complex to set up initially but provides flexibility.
Scalability	Limited scalability due to the need to replicate the entire application.	Easier scalability as individual services can be scaled based on demand.
Failure Impact	A failure in one component may affect the entire application.	Failures in one service do not impact others (isolated failures).
Dependency	Tight coupling between components.	Loose coupling between services, allowing for independent development and deployment.
Flexibility	Limited flexibility in adopting new technologies.	Flexibility to use different technologies for different services.
Technology Stack	A combined technology stack across the entire application.	Each service can use its own technology stack.
Example	Traditional monolithic web applications.	E-commerce applications with separate services for cart, user management, payment processing, etc.

Table 1: Comparison Between Monolithic Architecture And Microservices Architecture [1][2][10][11]

Architectures of Microservices Architecture and Monolithic Architecture

In the Microservices architecture, each service is independent and can be developed, deployed, and scaled separately. This provides flexibility and scalability benefits but introduces additional complexity in managing the interactions between services.[1][2][10][11]

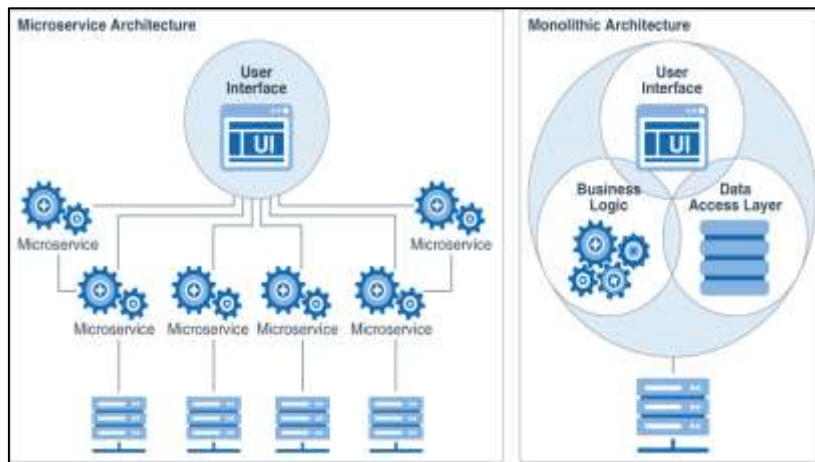


Fig. 1: Architectures of Microservices Architecture and Monolithic Architecture [10]

III. SOAPUI

(An open-source API testing tool): A Comprehensive Analysis: SoapUI, an open-source API testing tool has become a cornerstone in the realm of software testing. This research paper aims to provide an in-depth analysis of SoapUI, encompassing its features, capabilities, use cases, and performance metrics. Additionally, the paper explores the evolution of SoapUI, its integration with modern development practices, and offers insights into its potential future developments.[3][4][5][25]

A. SoapUI Features and Capabilities in relation to Microservices Architecture

It provides the ability to transform request and response messages, allowing users to handle different data formats effectively.[25]

SoapUI is enriched with the following Eight aspects –



Fig. 2: SoapUI Eight Aspects

1) Functional Testing

SoapUI allows/permits testers to write functional API tests in SoapUI with additional further capabilities such as supporting drag-drop feature which enhance script development, also provides debugging of tests and allows testers to develop data driven tests. And also, it makes it easy to switch between QA, Dev and Prod like multiple environments and allows advanced scripting (the tester can create his own custom code based on require scenarios).

2) Service Simulation (Mocking)

SoapUI Mock Services provides you simulate web services with the robustness of tests against them earlier implementing them it also eliminates the cost of building full-scale replicas of your production systems and empower consumers to access services without waiting for them to be built or available.

3) Security Testing

It protects your services from the most known security vulnerabilities on websites, using a complement of tests and scans. It provides Test Generator, SQL Injection, XML Bomb, Cross Site Scripting, Fuzzing Scan, Boundary Scan, Malicious Attachments, Custom Script, XPath Injection, Invalid Data, Malformed XML.

4) Load Testing

SoapUI easily runs load tests with high volume emulation in real-world load testing by diagonally distributing n number of LoadUI agents and also allows advanced custom reporting to capture performance parameters with end-to-end system performance monitoring gives.[25]

5) Technology support

When we are building either a web application or a web service based on various protocols it must loaded with advanced technologies, so it provides all the common protocols with the standards. So, whether you want to test and deploy SOAP services or Flex/Flash web applications, SoapUI is always available for us.

6) Automation

SoapUI is a package of providing various autonomous topographies, permitting you to intensely decrease labor costs and helpful to improve your time-to-market. It provides command-Line tools bundled so you can run your Functional/Load Tests and Mock Services from any task scheduler, even though customize the test execution to override test parameters and also controlling the tests to run or output, and much more.

7) Analytics

SoapUI makes your testing faster and helps you save countless hours and provides easy understanding of reports within the UI at the Project, TestSuite, TestCase and LoadTest levels.

8) Ecosystem

SoapUI biggest support system for the open-source community and partners around it with the accelerating the innovation one more reason why it is great is that it allows anyone to develop their own set of SoapUI features by using SoapUI Plugins.[13] [25]

IV. PROPOSED WORK-FLOW

(Proposed Structured Work Flow of SoapUI in Integration with Microservices Architecture) : A Proposed step-by-step flow of integration of SoapUI with a Microservices architecture with the explanation of each step as follows:

The thirteen Number of steps are proposed for the SoapUI integration with Microservices is given with the working of all the steps in this context. It starts with the understanding of the Microservices Architecture then we need to setup the SoapUI Project after this need to import the Microservices endpoints. After importing we need to generate and define the Microservices requests with its responses.

When we receive the responses then the main task of the SoapUI will start it starts implementing the Load Testing which is optional as load testing is require for parametrized request then we create and produce the test suits and next to this configure it with the assertions. If tester wants to arrange and set up data-driven testing then it is also optional as it is also like load testing for parametrized requests and this is also applicable same as for integrating the CI/CD pipelines.

After above all steps, we execute the tests with SoapUI, after execution analyze and review the results. Then at last Monitor and optimize throughput of the Microservices architecture.

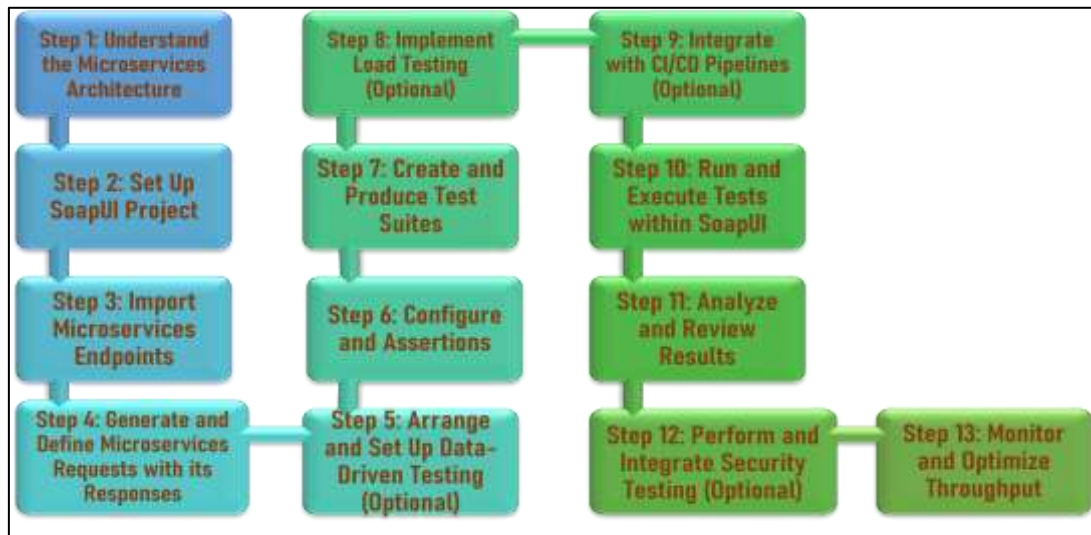


Fig. 3: A Proposed structured Workflow of SoapUI in Integration with Microservices Architecture

1) *Step 1: Understand the Microservices Architecture*

In the beginning of integration of SoapUI with Microservices Architecture it's important to have a clear understanding of the architecture including its services with endpoints.

The Endpoints are usually URLs and URIs that is use by the client to interact with the individual separate services and it is typically discovers one or more endpoints for performing various operation and they serve as the starting point to the functionalities provided by the service.

For Ecommerce based domain the following are the examples to understand the points:

- GET /products: Fetching a list of products.
- GET /products/{id}: A particular product details Fetching with the id.
- POST /shopping cart/additem: Adding an item to the shopping cart using additem.
- PUT /shopping cart/updateitem: Update the number of particular items in the cart.

2) *Step 2: Set Up SoapUI Project*

- 1) For this we need to Create a New SoapUI Project - Open SoapUI and create a new project
- 2) Give a name and select the WSDL or API definition file if applicable.

3) *Step 3: Import Microservices Endpoints*

In the project of SoapUI, create a new SOAP API by giving the base URL of the Microservices which includes the host name and the port numbers.

4) *Step 4: Generate and Define Microservices Requests with its Responses*

- Generate Requests: Inside API, generate requests for every endpoint of the Microservices with defining all the parameters, headers and body if applicable.
- Describe Expected Responses: Specify the expected response format and content for each request.

5) *Step 5: Arrange and Set Up Data-Driven Testing (Optional)*

This is an optional step for mainly Parameterize Requests to test different scenarios and can be done using data sources.

6) *Step 6: Configure and Add Assertions*

Define and Add assertions to validate the responses from Microservices by including checking status codes, response content, and more.

7) *Step 7: Create and Produce Test Suites*

- 1) Create and Organize Tests: within the SoapUI project to regarding to related group test cases organize and create test suites.
- 2) Add Test Cases: Create separate test cases for each Microservice endpoint, adding and incorporating the requests and assertions.

8) *Step 8: Implement Load Testing (Optional)*

Set Up Load Tests in load test cases: It is an optional step If load testing is required then we need to create load test cases within the SoapUI project to simulate high traffic scenarios.

9) *Step 9: Integrate with CI/CD Pipelines (Optional)*

Configure CI/CD Integration: Using the command -line interface Set up SoapUI in the CI/CD pipeline to automate testing as part of the deployment process and we can also do it by using the Jenkins plugin.

10) *Step 10: Run and Execute Tests within SoapUI*

Execute and run Test Suites within SoapUI to test the Microservices and also analyze the results to identify any errors and bugs or failures.

11) Step 11: Analyze and Review Results

Analyze and review the detailed test reports by generated SoapUI to increase the performance and throughput of the Microservices.

12) Step 12: Perform and Integrate Security Testing (Optional)

In case the Integrate Security, Testing is required then configure SoapUI to perform valuations /assessments on the Microservices.

13) Step 13: Monitor and Optimize Throughput

Observe and do the monitoring of the performance as needed based on SoapUI test results

V. PRACTICALITY IN E-COMMERCE DOMAIN

Practical Serviceability and User Experience of SoapUI in Microservices Architecture in current scenario for E-commerce based companies

There are three E-commerce companies as Company A, Company B and Company C that has implemented a Microservices architecture to empower its online platform. The architecture consists of various Microservices responsible for different characteristics and aspects of the E-commerce based domain system as following:

- 1) Product Service: This service includes and manages product detailed information, pricing, and availability.
- 2) Cart Service: This service handles the all the functionalities regarding, consenting users to add, remove, and modify items.
- 3) Order Service: This service smooths and facilitate the aspects like order creation and management, including processing payments.
- 4) User Service: This service manages all the user related information's like accounts, registration, authentication, and profiles.
- 5) Review Service: This service includes the sum up /collects the revies of products as well as displays product reviews from users.
- 6) Recommendation Service: The service provides all the recommendations of product based on user behavior.

Let's see a comparative analysis of SoapUI in Microservice Architecture across various E-commerce companies, so to understand this we are taking an example of three E-commerce based companies as



Fig. 4: SoapUI based integration of E-Commerce domain-based Companies A, B and C using CI/CD, load and security testing, Open API

- 1) E-Commerce Company A: The Company A has implemented SoapUI effectively by using API testing features also integrated it with the CI/CD pipeline with respect to data-driven testing in their Microservices.
- 2) E-Commerce Company B: The Company B is in the under process of implementing SoapUI and pull all its features for load testing and security testing.
- 3) E-Commerce Company C: The Company C has all the well- planned approach with an emphasis on Open API integration for implementing SoapUI with detailed reporting.

VI. PERFORMANCE ANALYSIS

Analyze the Performance: SoapUI Performance analysis with conclusion in various aspects of Microservices Environment

Aspects /Characteristics	Performance analysis Serviceability and User Experience	Company A	Company B	Company C
Integration	CI/CD Pipeline Integration for automatic testing for post -Deployment	Implemented	Planned	Under Review
Reporting in Detailed with Analytics	Analytics for throughput monitoring with the detailed reports	Used	Not Used	Used
Security Testing	To identify vulnerabilities, check capabilities for Security testing	Implemented	Planned	Under Review
Parallel Testing	Capacity to conduct parallel or simultaneous testing for multiple Microservices	Implemented	Not Implemented	Planned

Groovy Scripting	Ability to power Groovy scripts for customer validations	utilized	utilized	Planned
Virtualization of service	Mock service creation for testing dependencies	Implemented	Not Implemented	Planned
Evaluation and versioning	To ensure background compatibility support for versioning	Implemented	Planned	Under Review

Table 2: Soapui Performance Analysis with Conclusion in Various Aspects of Microservices Environment

VII. GRAPHICAL EVALUATION

Graphical illustration of SoapUI in Microservice Architecture across various E-commerce companies

In following graphical representation, I explain and illustrate the integration of SoapUI in the Microservices Architecture of three E-commerce companies such as Company A, Company B, and Company C.

A. Aspects /Characteristics in respect to integration of SoapUI in the Microservices Architecture of E-commerce Company A

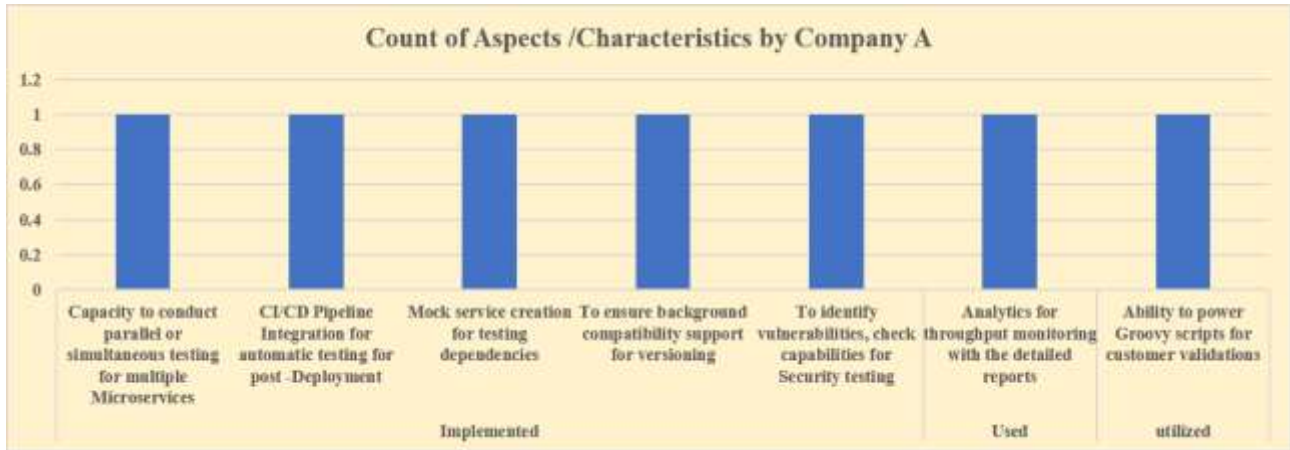


Fig. 5: Graphical Evaluation of characteristics of integration of SoapUI in the Microservices Architecture of E-commerce Company A

Evaluation: The integration in this is implemented for the CI/CD Pipeline Integration with Load testing and parallel testing also Mock service creation for testing dependencies is implemented. It utilized the Groovy Scripting for customer validation.

B. Aspects /Characteristics in respect to integration of SoapUI in the Microservices Architecture of E-commerce Company B

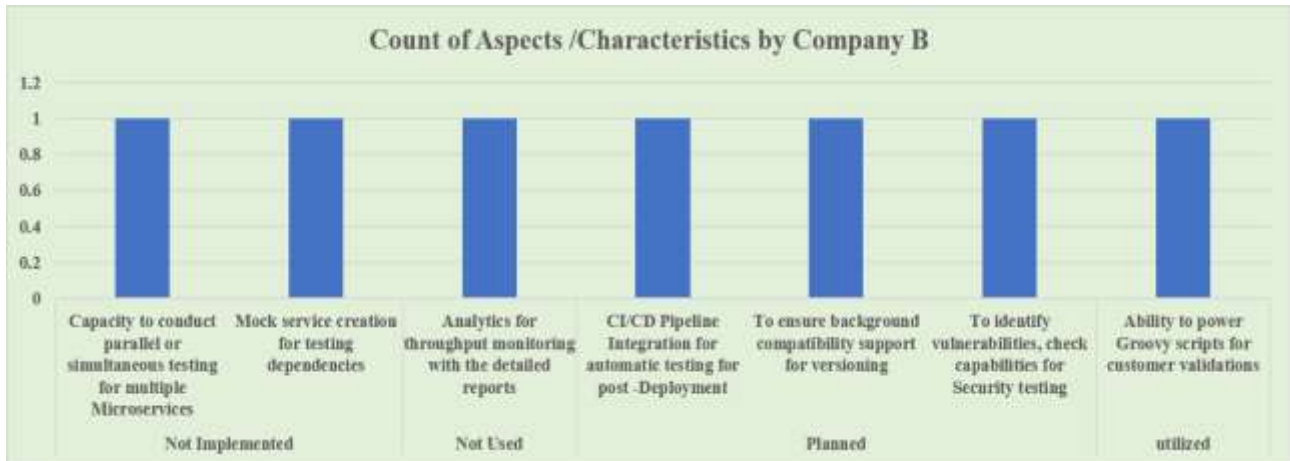


Fig. 6: Graphical Evaluation of characteristics of integration of SoapUI in the Microservices Architecture of E-commerce Company B

Evaluation: In this the integration is planned in respect to security testing by utilizing the Groovy scripting and they planned to pull the SoapUI's capabilities for security measures.

C. Aspects /Characteristics in respect to integration of SoapUI in the Microservices Architecture of E-commerce Company C

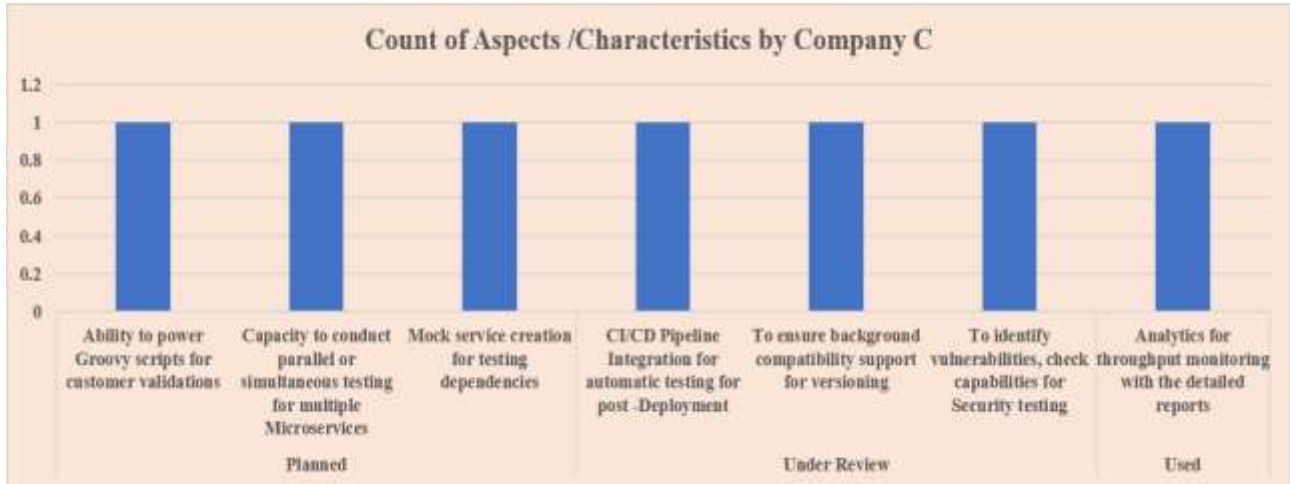


Fig. 7: Graphical Evaluation of characteristics of integration of SoapUI in the Microservices Architecture of E-commerce Company B

Evaluation: It is used the well- structured approach for reporting in detailed with analytics by emphasizing Open API integration.

D. Aspects /Characteristics in respect to integration of SoapUI in the Microservices Architecture of all three E-commerce Companies A, B and C

Evaluation: It is representing an analytical comparison of all three companies which shows the characteristics in the factors of planned, implemented, under review and used aspects.

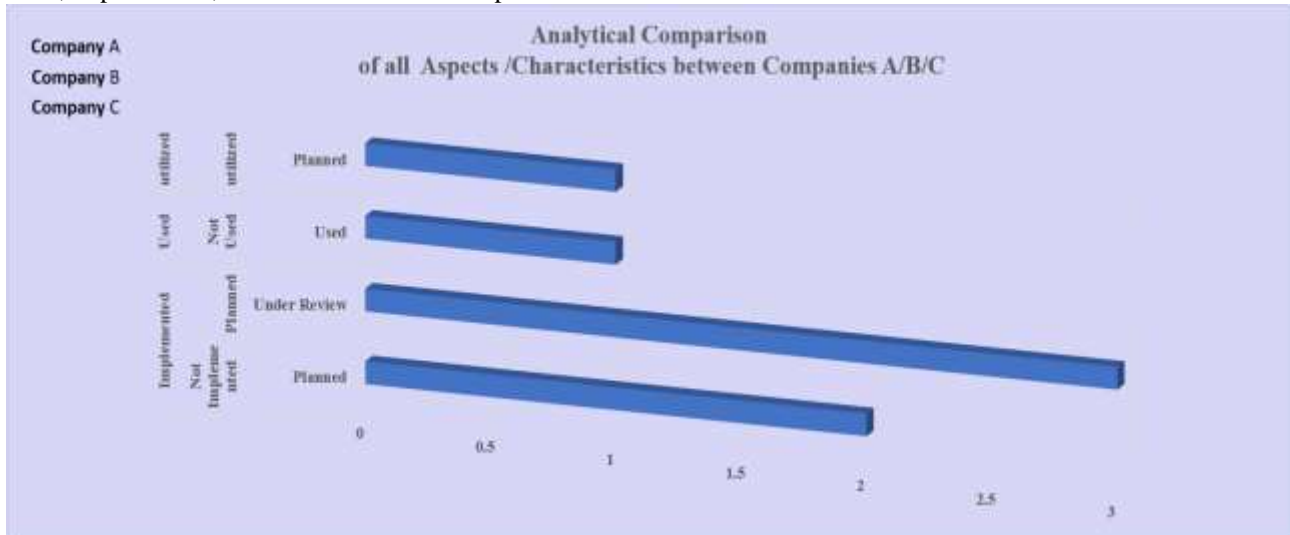


Fig. 8: Graphical Evaluation of Analytical comparison of all aspects/characteristics of integration of SoapUI in the Microservices Architecture of E-commerce between Companies A/B/C

VIII. CONCLUSION AND RESULTS

After finalizing this proposed structured approach, it is concluded that: In this paper, I have described a structured step by step Workflow of Microservices Architecture testing integration with the most reliable open-source testing API tool i.e., SoapUI by using a detailed and context- specific aspects in related to using specific use cases. The E-commerce based companies shows the test results in the context of planned, used, under review and utilized (The contradiction representation is given below under the all three companies). SoapUI bids a rich network of plugins which is allowing users to cover and extent its functionality and integrate with other tools or services.

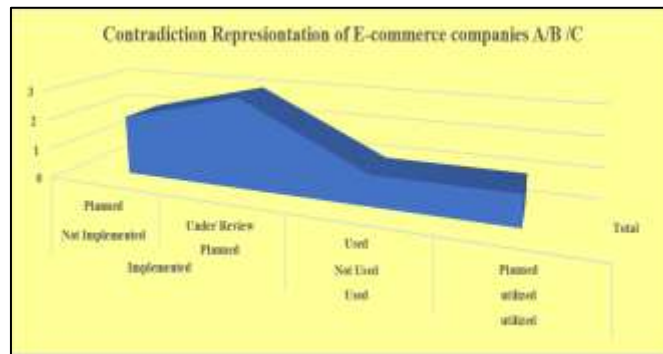


Fig. 9: Representation of all aspects/characteristics of integration of SoapUI in the Microservices Architecture of E-commerce between Companies A/B/C

IX. FUTURE WORK

It can be challenging to predict the future work and development of a specific tool or technology as it is totally depending upon the needs of the software industries as well as the direction taken by the software developers. However, I can provide some latest probable domains of future work for SoapUI in the context of Microservices architectures such as SoapUI could incorporate the tools and features of most popular GraphQL by using advanced security testing capabilities. In future work it includes the proposed structured to step by step practical demonstration which includes the endpoints of microservices architecture.

ACKNOWLEDGMENT

I extend my deepest gratitude to my husband who provided not only his expertise, encouragement and boundless belief in my abilities have been a pillar of strength but also his unwavering belief in the potential of this work. His wisdom has been the compass guiding in this research journey.

REFERENCES

- [1] Nada Salaheddin Elgheriani and Nuredin Ali Salem Ahmed. "Microservices Vs. Monolithic Architectures [The Differential Structure Between Two Architectures]". International Journal of Applied Sciences and Technology ISSN: 2717-8234. DOI:10.47832/2717-8234.12.47. October 2022.
- [2] Omar Al-Debagy and Peter Martinek. "A Comparative Review of Microservices and Monolithic Architectures". CINTI 2018 • 18th IEEE International Symposium on Computational Intelligence and Informatics • Nov. 21-22, 2018, • Budapest, Hungary. .2018
- [3] Vatsya Tiwari, Sachin Upadhyay, Jayati Krishna Goswami and Sanjiv Agrawal. "Analytical Evaluation of Web Performance Testing Tools: Apache JMeter and SoapUI". Conference: 12th IEEE International Conference on Communication Systems and Network Technologies. <https://ieeexplore.ieee.org/xpl/conhome/10134573/proceeding> . June 2023.
- [4] Arora A. and Sinha M. "Web Application Testing: A Review on Techniques, Tools and State of Art". International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February-2012 1 ISSN 2229-5518. February-2012
- [5] Charitha Kankanamge. Web Services Testing with SoapUI. Published by Packt Publishing Ltd, Livery Place 35 Livery Street, Birmingham B3 2PB, UK, ISBN 978-1-84951-566-5, www.packtpub.com.
- [6] Sana Azzam, Mohammed Naji Al-Kabi and Izzat Alsmadi. "WEB SERVICES TESTING CHALLENGES AND APPROACHES". 2012 International Conference on Communications and Information Technology (ICCIT), IEEE Conference, IEEE Xplore.01-01-2012. January 2012.
- [7] Ravi Kumar and Pratibha Sharma . "A Comparative Study and Analysis of Web Service Testing Tools". International Journal of Computer Science and Mobile Computing, Vol.4 Issue.1, January- 2015, pg. 433-442. January- 2015.
- [8] Thorsten Rangnau , Remco v. Buijtenen ,Frank Fransen and Fatih Turkmen. "Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines." 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC). <https://doi.org/10.1109/EDOC49727.2020.00026> . 2020
- [9] Neha Sharma, Pratibha Sharma and Hakam Singh. "A Comparative Study of Web Service Testing Tool". Journal of Systems and Software Volume 150, April 2019, Pages 77-97. <https://doi.org/10.1016/j.jss.2019.01.001>. April 2019.
- [10] Paolo Di Francesco, Patricia Lago and Ivano Malavolta. "Architecting with microservices: A systematic mapping study". 2017 IJEDR, Volume 5, Issue 2, ISSN: 2321-9939, IJEDR1702227.2017.
- [11] Tomas Cerny, Amr S. Abdelfattah, Abdullah Al Maruf , Andrea Janes and Davide Taibi . "Catalog and detection techniques of microservice anti-patterns and bad smells: A tertiary study" Journal of Systems and Software , Volume 206, December 2023, 111829, <https://doi.org/10.1016/j.jss.2023.111829>. December 2023.
- [12] Florian Auer, Valentina Lenarduzzi, Michael Felderer, Davide Taibi. "From monolithic systems to Microservices: An assessment framework". Information and Software Technology 137 (2021) 106600. <https://doi.org/10.1016/j.infsof.2021.106600> . 2021.

- [13] Mustafa Bozkurt, Mark Harman and Youssef Hassoun. "Testing Web Services: A Survey" <https://www.researchgate.net/publication/228920258> 01-01-2010. 2010.
- [14] Muhammad Waseem and Peng Liang, Gastón Márquez and Amleto Di Salle. "Testing Microservices Architecture-Based Applications: A Systematic Mapping Study". 2020 27th Asia-Pacific Software Engineering Conference (APSEC). <https://www.researchgate.net/publication/344679617>. 2020.
- [15] Mojtaba Shahina, Muhammad Ali Babara and Liming Zhub. "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices". March 2017 IEEE Access PP(99). <http://dx.doi.org/10.1109/ACCESS.2017.2685629>. March 2017.
- [16] Mr. B. B. Patel and Dr. D. B. Choksi. "A Comparative Study of Web Service Testing Tools". 2018 JETIR October 2018, Volume 5, Issue 10 www.jetir.org (ISSN-2349-5162). .2018.
- [17] Shikha Dhiman and Pratibha Sharma. "Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools". IJCSMC, Vol. 5, Issue. 6, June 2016, pg.507 – 512. June 2016.
- [18] Shariq Hussain, Zhaoshun Wang, Ibrahima Kalil Toure and Abdoulaye Diop. "Web Service Testing Tools: A Comparative Study." IJCSI International Journal of Computer Science Issues, vol.10, no.1-3, 2013, pp.641-647. <https://doi.org/10.48550/arXiv.1306.4063>.2013
- [19] Israr Ghani, Wan M.N. Wan-Kadir and Ahmad Mustafa. "Web Service Testing Techniques: A Systematic Literature Review". (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 8, 2019.
- [20] <https://www.softwaretestingmagazine.com/tools/open-source-api-testing-tools/>
- [21] Rakesh Kumar Lenka, Meenu Rani Dey, Pranali Bhanse and Rabindra Kumar Barik. "Performance and Load Testing: Tools and Challenges." 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE). DOI: 10.1109/ICRIEECE44171.2018.9009338 . 2018.
- [22] S.Susila, S.Vadivel and Aditya Julka. "Broker Architecture for Web Service Selection Using SOAPUI". 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM). 08-10 December 2012. Publisher: IEEE Conference. <https://doi.org/10.1109/ICCCTAM.2012.6488102>.2012.
- [23] Anuja Bokhare, Niraj Bohra and P. S. Metkewar. "Benchmarking of testing tools used for web services". International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 23 (2015) pp 43772-43775. <http://www.ripublication.com>. 2015.
- [24] Abdul Wahab Mohamed and Ahmed M. Zeki. "Web services SOAP optimization techniques". 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS). Date of Conference: 29 November 2017 - 01 December 2017. Date Added to IEEE Xplore: 01 February 2018. INSPEC AccessionNumber:17559238. <https://doi.org/10.1109/ICETAS.2017.8277881>. Publisher: IEEE Conference. .2017
- [25] "soapUI – The Home of Functional Testing", <https://www.soapui.org/>
- [26] Ilona Bluemke, Michał Kurek and Małgorzata Purwin. "Tool for Automatic Testing of Web Services". Electronic ISBN:978-83-60810-58-3. Publisher: IEEE. <https://doi.org/10.15439/2014F93>. 2014.
- [27] Animesh Chaturvedi and Atul Gupta. "A Tool Supported Approach to Perform Efficient Regression Testing of Web Services". 2013 7th IEEE International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA). DOI: 10.1109/MESOCA.2013.6632734. 2013.
- [28] Roopesh Kevin SUNGKUR and Sachin DAIBOO. "Combining the Best Features of SOAP and REST for the Implementation of Web Services". International Journal of Digital Information and Wireless Communications (IJDWC) 6(1): 21-33 The Society of Digital Information and Wireless Communications, 2016 ISSN: 2225-658X (Online); ISSN 2412-6551, 2016.