International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Integration of Data on Different Platforms Using Web Services

Venkata Nagendra Kumar Kundavaram
Goodwill Easter Seals, Minnesota., USA

Venkata Krishna Reddy Kovvuri
krishna.kovvuri09@gmail.com

Abstract

This paper discusses the usage of 'web services' to facilitate an easy integration of heterogeneous systems at the data level. Prioritizing the 'HTTP' and 'XML' protocols, the research points to their effectiveness in delivering 'real-time interoperability' and 'reusability' to the trends of 'SOA' and 'data synchronization'. A qualitative study in the form of a secondary approach and thematic analysis discloses the presence of integration potentials and challenges in distributed digital contexts.

Keywords: Data sources, Data integration, Protocol, Real-time, reusability, HTTP, XML, SOA, data synchronization, interoperability, flexibility, web services

I. INTRODUCTION

The contemporary digital environment makes use of the increasing number of data sources scattered across the 'heterogeneous systems and platforms on which organizations depend. Enterprise information architecture requires the use of 'data integration' as a major feature that allows unification and simplification of data accessibility. Using the 'web services' as the standardized protocol, and interface provides a formidable means of integration with respect to the smooth flow of information and interoperability between heterogeneous systems [1]. 'Web services' facilitate real-time integration, scalability, and reusability across dynamic environments by facilitating communication using the internet, via common open protocols including 'Hypertext Transfer Protocol (HTTP)' and 'Extensible Markup Language (XML)'.

The paper discusses the use of web services in bringing seamless data integration within the platform, focusing on 'service-oriented architecture (SOA)', and 'RESTful APIs' services. The framework examines the use of these technologies in 'distributed computing', 'data synchronization', and connectivity of enterprise applications [2]. Moreover, the study focuses on the use of web services in obviating past challenges in terms of 'integration', including the 'inability' of systems to interact, and manual transfer of data.

Aim and Objectives of the Research:

This study aims at the general application of 'protocols', 'service-oriented architecture (SOA)', and 'data synchronization' in promoting ease of data flow to facilitate a data integration process across different platforms. Using web services to understand the workflow of integrating information and data without constraint across a wide variety of databases.

- To determine 'web services' capabilities that facilitate 'real-time', 'large-scale', reusable integration of data sources based on open protocols, including 'HTTP' and 'XML'.
- To assess the 'service-oriented architecture (SOA)' and 'data synchronization' techniques in solving the integration problems of distributed systems.

'Data synchronization' is applied in encouraging a data flow process with attributes of seamless inter-database and inter-application integration [3]. The standardization of web services as a protocol and interface is a powerful way to integrate in terms of the free exchange of information and interoperability among heterogeneous systems.

II. LITERATURE REVIEW

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

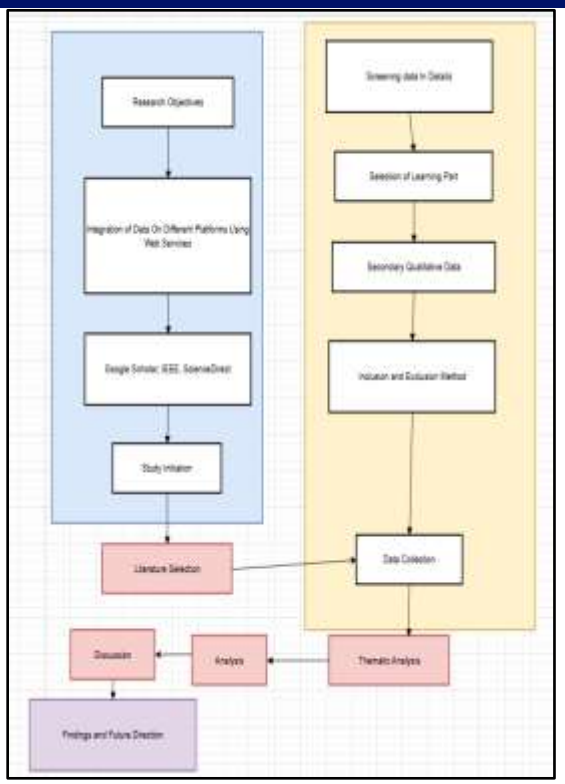

Fig. 1: Flow of the Research

The systematic literature review method entails the following steps:

I. *Key concepts:* The review is focused on 'data combination', 'web services', 'service-oriented architecture (SOA)', 'HTTP', 'XML', and reusability between numerous systems.

II. *Selection of Literature:* The selection uses targeted literature searches by using keywords including 'HTTP', 'XML protocols', and 'real-time synchronization' of data.

III. *Thematic Analysis:* It is implemented to determine emerging trends, issues in integration, and transition to 'service-based architecture'.

Scholarly Database and Source Utilization:

I. *IEEE Xplore*: It is used to find technical contributions and applications of web services to enterprise integration.

II. *ScienceDirect*: Offers peer-reviewed articles regarding 'SOA', 'protocol design', and 'distributed computing'.

III. *Google Scholar:* Subsidized entry into wide-ranging academic discourse on synchronizing data across platforms, and interoperability of platforms.

A. Searching Study:

This study uses specific keywords in designing its search strategy, including 'secure protocol', 'reusability', 'HTTP', 'XML' and 'SOA', and 'data synchronization'. The use of academic databases listed in 'Google Scholar', 'ScienceDirect', and 'IEEE Xplore' is used, connected with 'interoperability' and 'platform integration' of web services.

B. Selection of Journal Articles

The journal articles selected are relevant to the study and focus on real-time data integration, reusability, and 'protocol-driven communication' with the use of 'HTTP', and 'XML'. It is preferred that recent research and reviews, published as peer-reviewed papers, explore the sources of data, the 'SOAs framework', and 'data synchronization' among numerous systems.

C. The Goal of the Review

This review aims to assess the role of web services in the improvement of 'data integration' because it employs common protocol frameworks like 'HTTP' and 'XML' in accessing 'data sources. It analyzes the capabilities of 'service-oriented architecture (SOA)' to facilitate synchronization of data in 'real-time', and increase the 'reusability' of systems, and the 'interoperability' of systems in distributed environments.

D. Study of Previous Literature

'Protocol-Driven' Access to Information Sources

Combining 'distributed data sources' requires a uniform message technique so they can easily interoperate across diverse machines. 'Web services' use open standards (HTTP and XML) and allow homogeneous, 'real-time access to various data systems [4]. This protocol-based strategy provides a simplified data integration route, and it also increases the level of system scalability and reusability.

'SOA' and 'Real-Time Data Synchronization'

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

'Service-oriented architecture (SOA)' is a way of 'real-time synchronization' of distributed data with modularity of service communications [5].

Fig. 2: Execution of 'data synchronization'

'SOA' inculcates unified and constant information coordination of varying sources through standard modes of protocol and data arrangement, including 'HTTP' and 'XML'.

Using 'HTTP' and 'XML' to increase Reusability

'HTTP' and 'XML' are pre-requisite protocols that foster the 'reusability' of various 'data sources' that allow seamless communication and 'data fusion' between distributed systems [6]. In this context, they improve system interoperability and decrease 'redundancy' within enterprise settings and real-time interchange of information.

'Web Services' to Flawless 'Data Integration'

'Web services' support data integration in a seamless manner because they provide standard interfaces that ensure sources of data are harmonized in heterogeneous platforms [7].

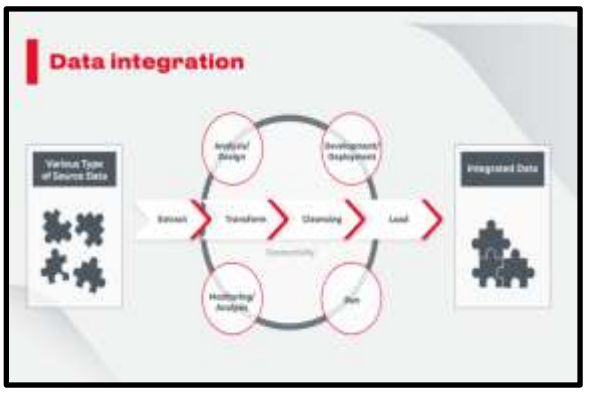

Fig. 3: Execution of 'data integration'

In this context, this makes use of open protocols like 'HTTP' and 'XML' to enable 'real-time data exchange', 'reusability', and 'interoperability'. 'Synchronization' of 'data flow' and consistent connectivity across 'distributed systems' is further enabled by the application of the 'SOA'.

Literature gap

Several studies have been conducted regarding 'data integration' through 'web services', the discussion of this paper showcases a lack of attention to extensive empirical assumptions of 'interoperability' issues among heterogeneous systems. Technical protocols, including 'HTTP' and 'XML', are widely discussed in numerous studies, and scaling challenges like 'latency', 'security', and working with dynamic data are not discussed thoroughly. Moreover, less attention is paid to the 'real-time synchronization assessment' of 'SOA' under multi-platform, multi-complex scenarios of the comprehensive, context-based analysis.

III. METHODOLOGY

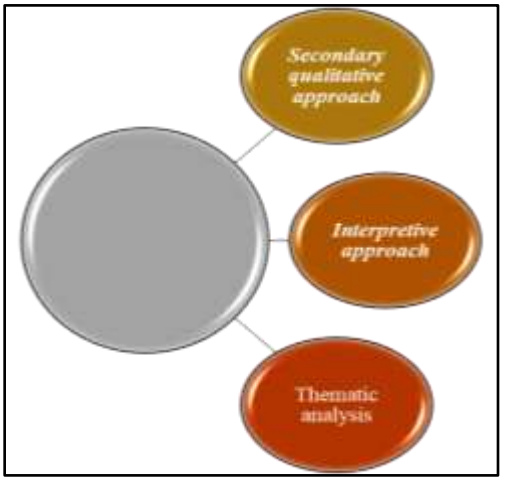

Fig. 4: Methodology

The given study will consist of a ***"secondary qualitative research"*** design with an interpretive basis that aligns with the purpose of determining 'web services' capabilities that facilitate 'real-time', 'large-scale', reusable integration of data sources based on open protocols, including 'HTTP' and 'XML' [8]. The ***"interpretive approach"*** helps in enhanced conceptualizing of frameworks, models, and

automated methodologies that operate to facilitate real-time integration, scalability, and reusability across dynamic environments. It helps the 'service-oriented architecture (SOA)' and 'data synchronization' techniques in solving the integration problems of distributed systems [9]. The research entails a scientific study of peer-reviewed journal articles, technical reports, and case studies retrieved with the help of scholarly databases including 'ScienceDirect', 'IEEE Xplore', and 'Google Scholar'. A strict selection approach was utilized where attention was only directed at publications that were made within the past 4-5 years, which discuss themes including the use of open standards, including 'HTTP' and 'XML', and allow homogeneous, 'real-time accesses to various data systems [10]. These types of materials are analyzed thematically with "***thematic analysis"*** keywords including 'security Protocol', 'real-time data synchronization', 'HTTP', 'XML', 'SOA', and 'data synchronization'.

IV. DATA ANALYSIS

A. Thematic Analysis:

Theme 1: 'Web Services' enable the 'large-scale reusable data' to be integrated in real-time with the use of open protocols

The need for uninterrupted access to 'distributed data sources' promoted the use of potent integration frameworks in the digital transformation era. The 'web services' have a key pillar in supporting the integration of data that exists in diverse systems by taking advantage of 'standardized protocols' that support 'interoperability' and communication within the system [11]. These protocols include 'Hypertext Transfer Protocol (HTTP)' and 'Extensible Markup Language (XML)' as the main protocols in developing a homogeneous process of representation and exchange of data. In this context, it is a discussion of the capacity of 'web services' towards 'large-scale', 'real-time', and 'reusable data integration' based on open protocols across multiple platforms.

The 'web services' are interfaces supporting standard mechanisms available through the abstraction of system-specific details of communication. Protocol-oriented services use 'XML' and 'HTTP', which facilitates uniform communication between two different systems. In this context, this is particularly useful in the case of large-scale businesses in which data would need to be compiled across a variety of departments and multiple platforms [12]. The 'reusability' of integration components entails enormous savings of effort in such situations and efficient development. 'HTTP' offers a stateless, weightless model of communication that is scalable, and 'XML' structures data in a machine and human-readable format where data is understood and processed at all ends.

The 'web services' facilitate 'real-time interaction', which is essential in data-driven decisions, and the implications of failing to synchronize data in time in the likes of financial services, healthcare, logistics, and online businesses [13]. 'Transmission of systems can occur in a loosely coupled, deployable, and independent manner through the use of 'service-oriented architecture (SOA)' services in response to real-time events. In this context, this is an architecture paradigm that supports a 'real-time retrieval', transformation, and transmission of information in the distributed system [14]. 'Event-driven processes' are helped by 'HTTP-based' 'RESTful APIs' in 'SOA', allowing the business to respond immediately to changes in data and making them more responsive and competitive.

'Reusability' is another important benefit of real-time data integration with the help of web services. Components of 'web services' are constructed using standardized interfaces so that they can be reused in many applications and business arenas without long customization efforts [15]. In this scenario, an example is a payment gateway service that is developed according to the 'XML' and 'HTTP' protocols and is able to interact with several e-commerce systems with less configuration.

The possibility to interoperate distinct systems is one of the fundamental tasks to deal with, which web services solve as one of the key issues. In IT complex ecology, it is usually a reality that organizations have both legacy systems and innovative infrastructures based on 'cloud computing' [16]. The lack of uniform integration mechanisms makes 'synchronization' of these systems take more time to respond and prone to errors due to manual data transfer between them. The use of 'protocol-agnostic communication' layers by web services fills this gap since it simplifies the process of harmonizing disparate data [17]. In this context, it relies on 'HTTP' and 'XML' formats all the time means that data interchanged will always be structured, validated, and platform-free within 'cross-system integrations.

Utilization of 'data synchronization processes' that are enhanced by the use of 'web services' plays a crucial role in providing consistent updates on data [18]. For instance, an 'inventory information' has been changed in a warehouse management system, and gets

synchronized instantly with an e-commerce portal via service calls. In the thematic analysis, it is identified that the strength of protocols such as 'HTTP' and 'XML' in the realization of integration has been implemented through assessments of the technologies in 'large-scale real-time' settings.

Theme 2: Challenges in integration within the 'distributed systems' are addressed using 'SOA' and 'Data Synchronization Techniques.

The seamless 'interoperability' and 'consistency' across platforms have been some of the most common challenges found in the literature in the context of integrating the enormous and diverse body of information found in distributed systems. In a 'distributed system', by default, it has heterogeneous databases, different communication standards, and different storage formats. The result of this fragmentation is complexity in 'data integrations and 'latency', 'redundancy' and 'information silos. Moreover, with the advent of the concept of 'web services', designed with 'Service-Oriented Architecture (SOA)' and augmented with 'data synchronization approaches', there is an approach to determining these problems in an organized way [19]. The 'Service-Oriented Architecture (SOA)' has a greater role in resolving the shortcomings of integration in distributed systems. 'SOA' facilitates the modular design concept, and it means that individual services are capable of interacting with each other through standard protocols like 'HTTP' and 'XML' [20]. These types of modularity enhance the re-use of services, enabling organizations to expand their operations without necessarily having to redesign the 'system architectures'.

'Distributed application' is capable of ensuring real-time communications and dynamic reactions to synchronize across geographically or functionally distant platforms with the help of 'SOA' [21]. The 'regular adoption' of open protocols will make sure that there is no constraint on integration in 'proprietary technologies' that limit 'flexibility' and 'slacken' the procedure of interoperability. 'Data synchronization' is one of the key parts of such an integration framework, and 'Synchronization' guarantees that the alteration of a single source of data is reflected in all of the pertinent platforms immediately [22]. The introduction of 'synchronization systems' based on 'SOA', helps enterprises to bring their data sources into a single coherent structure. Following this approach ensures a smooth flow of information without the involvement of manual reconciliation of data, and it is 'time-consuming'.

'HTTP' and 'XML', along with 'SOA', facilitate the exchange of structured data in a platform-free way [23]. The application of these technologies not only leads to interoperability but also enhances the 'reusability' of components because information can be posted amongst various systems. For instance, 'XMLs' are specifically structured and can be precisely defined and parsed data fields, which is essential to integrate systems [24]. The method is well known to be efficient in the process of 'integrating data' in real applications that are distributed, at the time of standardization in the communication.

The other problem that is solved by 'SOA' and 'synchronization technologies' is that there is no real-time responsiveness in any traditional integration techniques. The legacy systems usually involve the use of 'batch processing' and 'periodic data dump', and these make information updated late and limit the rapidness of key insights [25]. 'SOA' allows the services to be event-driven, which means that the integration processes that are vital to maintain dynamic and responsive. In this context, this real-time feature, combined with well-defined 'synchronization protocols', allows updating all the systems in the distributed environment at the same time.

Overall, 'SOA' and 'data synchronization' have their advantages, and it is indicated that gaps earlier existed in the comprehension of implementing the two features in complex systems [26]. Other possible weaknesses brought about by 'real-time protocols' are vulnerabilities unless handled by tight access and encryption functions. Future research focuses on securing such integration layers in an attempt to guard crucial information in the process of 'synchronizing' and 'transmitting' information. The analyzed articles still support the view that the strategic application of web services is based on SOA, and enabled by synchronization techniques the basis of modern and scalable integration solutions. Ultimately, a combination of the benefits of 'SOA' and 'data synchronization' attempts to address the fundamental integration issues associated with distributed systems by normalizing the communication form, enabling modular reuse, and enabling real-time synchronization of the data.

V. RESULT AND DISCUSSION

In the thematic analysis of this research, the themes together illustrate the role of 'web services' in ensuring a smooth and 'easy integration' of data in distributed platforms. The themes work directly and explain the key point that the open protocols, modular

'SOA framework', and strategy of data synchronicity act as prime enablers of the complex data environment of 'interoperability'.

The first theme underlines the fact that web services made it easy to integrate large-scale, reusable, and 'real-time data' through the use of 'standardized protocols' like 'HTTP' and 'XML'. In the findings, the abstraction that is provided by the 'web service' interfaces has helped the organizations to achieve 'data integration' by merging the 'local data' based on the 'standardized communication' facilities [27]. In this context, this will make it easier to integrate data between different systems and scale up the system. Specifically, it is possible to exchange data using 'HTTP' and 'XML' without any problems of compatibility caused by using 'HTTP' as the server-side messaging mechanism and 'XML' to express structured 'machine-readable' data. Following this approach amplifies directly in strengthens the reusability of their components in diverse applications, which facilitates the liveliness of enterprise activity.

The second theme is concerned with the overcoming of inherent integration concerns in distributed systems using the 'Service-Oriented Architecture (SOA)' and 'data synchronization' processes. 'Modularity' of the 'SOA' enables services to communicate flexibly with the platform, and 'synchronization' makes the updates to any data source be replicated in real-time throughout the data network [28]. These factors minimize 'redundancy', and facilitate connections with uncertainties that slow down systems due to manual data reconciliation. In addition, this paper strengthens the argument for secure and scalable protocol-integrated procedures of integration to support data consistency and operational flexibility.

Implications:

This research is valuable in the sense that it represents the tendency that implementation of 'web services with standardized 'protocols and 'SOA' platforms allows organizations to attain strategic interoperability, and scalable real-time data integration. Additionally, it does not just represent a technological improvement and it is transitioning to more agile, reusable, and competitive digital infrastructures.

Limitations:

The main weakness of this study is based on secondary data and a thematic analysis, which, being conceptually rich, lacks validation by the practical application. The cross-platform integration of data using web services in a practical way should be evaluated in future studies done on a primary basis of organizations using web services in an active manner [29].

VI. FUTURE RESEARCH

In future research, it is anticipated that an attempt will be made to perform empirical research through the collection of primary organizational data, which is active in the entire 'web services' to continue 'cross-platform data integration' [30]. The research will discuss practical evaluation criteria like improvement of 'latency', 'scalability', and 'responsiveness' of the system to changes in data. The research will also evaluate the effects of 'SOA', 'HTTP', 'XML', and 'data synchronization' in complex operational engagements. Additional 'real-time protocol security' implications are also studied, and an optimization of 'encryption' and 'access control' is proposed.

VII. CONCLUSION

This paper critically assessed the integration of data on heterogeneous environments via the use of 'web services and the contribution of 'standardized protocols', 'service-oriented architecture (SOA)', and 'data synchronization methods. Using open protocols like 'HTTP' and 'XML' to develop 'web services aids substantially in 'real-time data integration', 'reusability', and 'interoperability' across distributed systems. The 'latency', 'redundancy', and 'manual data reconciliation concerns are entry points of the integrating challenges that are divided into two, including the modularity of 'SOA' and the 'real-time responsiveness'. Thematic analysis ascertained the fact that the technologies assist in providing 'flexible' and 'scalable' communication models required by contemporary digital environments. The use of 'secondary data' and the lack of empirical confirmation restricts the usefulness of the generalization of the findings. Primary study in organizations that use web services is also what future research should be about, as they will help in evaluating further their business functionality.

VIII. REFERENCES

[1] Beuchelt, G., (2017). Securing Web applications, services, and servers. In *Computer and information security handbook* (pp. 183-203). Morgan Kaufmann.

[2] Neinstein, A., Wong, J., Look, H., Arbiter, B., Quirk, K., McCanne, S., Sun, Y., Blum, M. and Adi,

S., (2016). A case study in open source innovation: developing the Tidepool Platform for interoperability in type 1 diabetes management. *Journal of the American Medical Informatics Association*, 23(2), pp.324-332.

[3] Makamba, M., Mtsweni, J.S. and Ngassam, E.K., (2016). Exploring the dynamic integration of heterogeneous services.

[4] Naik, N., Jenkins, P., Davies, P. and Newell, D., (2016), December. Native web communication protocols and their effects on the performance of web services and systems. In *2016 IEEE international conference on computer and information technology (CIT)* (pp. 219-225). IEEE.

[5] KESKAR, A., 2020. Enhancing Software-Defined Vehicles with Service-Oriented Architecture: A Framework for Scalable and Modular Mobility Solutions.

[6] Yang, G., Jan, M.A., Rehman, A.U., Babar, M., Aimal, M.M. and Verma, S., (2020). Interoperability and data storage in internet of multimedia things: investigating current trends, research challenges and future directions. *IEEE Access*, 8, pp.124382-124401.

[7] Fortino, G., Savaglio, C., Palau, C.E., De Puga, J.S., Ganzha, M., Paprzycki, M., Montesinos, M., Liotta, A. and Llop, M., (2018). Towards multi-layer interoperability of heterogeneous IoT platforms: The INTER-IoT approach. *Integration, interconnection, and interoperability of IoT systems*, pp.199-232.

[8] Del Esposte, A.D.M., Santana, E.F., Kanashiro, L., Costa, F.M., Braghetto, K.R., Lago, N. and Kon, F., (2019). Design and evaluation of a scalable smart city software platform with large-scale simulations. *Future Generation Computer Systems,* 93, pp.427-441.

[9] Alshinina, R. and Elleithy, K., (2017). Performance and challenges of service-oriented architecture for wireless sensor networks. *Sensors*, 17(3), p.536.

[10] Jaimes, C.I.R. and Alvares, A.J., (2019). Integrated inspection system step-compliant for the exchange of dimensional metrology data. *Procedia Manufacturing*, 38, pp.1205-1212.

[11] Serhane, Y., Sekkaki, A., Benzidane, K. and Abid, M., (2020). Cost effective cloud storage interoperability between public cloud platforms. *International Journal of Communication Networks and Information Security*, 12(3), pp.440-449.

[12] Bentaleb, A., Taani, B., Begen, A.C., Timmerer, C. and Zimmermann, R., (2018). A survey on bitrate adaptation schemes for streaming media over HTTP. *IEEE Communications Surveys & Tutorials*, 21(1), pp.562-585.

[13] Halper, F. and Stodder, D., (2017). What it takes to be data-driven. *TDWI Best Practices Report, December*, pp.33-49.

[14] Dadras, A.S., (2016). IT agility through service-oriented architecture. *The University of New South Wales, Kensington, Australia*.

[15] Jiang, P., Elag, M., Kumar, P., Peckham, S.D., Marini, L. and Rui, L., (2017). A service-oriented architecture for coupling web service models using the Basic Model Interface (BMI). *Environmental Modelling & Software*, 92, pp.107-118.

[16] Gholami, M.F., Daneshgar, F., Beydoun, G. and Rabhi, F., (2017). Challenges in migrating legacy software systems to the cloud—an empirical study. *Information Systems*, 67, pp.100-113.

[17] Raviglione, F., Malinverno, M. and Casetti, C., (2019), September. A Flexible, Protocol-Agnostic Latency Measurement Platform. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)* (pp. 1-5). IEEE.

[18] Faiz, M. and Shanker, U., (2016), March. Data synchronization in distributed client-server applications. In 2*016 IEEE International Conference on Engineering and Technology (ICETECH)* (pp. 611-616). IEEE.

[19] Cerny, T., (2019). Aspect-oriented challenges in system integration with microservices, SOA and IoT. *Enterprise Information Systems,* 13(4), pp.467-489.

[20] Pirnau, C. and Botezatu, M.A., (2017). Service-oriented architecture (SOA) and Web services. *Database Systems Journal*, 7(4), pp.32-39.

[21] Espi-Beltran, J.V., Gilart-Iglesias, V. and Ruiz-Fernandez, D., (2017). Enabling distributed manufacturing resources through SOA: The REST

approach. *Robotics and Computer-Integrated Manufacturing*, 46, pp.156-165.

[22] Lomotey, R.K., Nilson, J., Mulder, K., Wittmeier, K., Schachter, C. and Deters, R., (2016). Mobile medical data synchronization on cloud-powered middleware platform. *IEEE Transactions on Services Computing*, 9(5), pp.757-770.

[23] Yeşiltepe, M. and Bozkurt, Ö.Ö., 2016, April. SOA layers and layers safety definitions. In 2016 International Conference on Computing, Communication and Automation (ICCCA) (pp. 813-816). IEEE.Yeşiltepe, M. and Bozkurt, Ö.Ö., (2016), April. SOA layers and layers safety definitions. In *2016 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 813-816). IEEE.

[24] Achakulvisut, T., Acuna, D.E. and Kording, K., (2020). Pubmed parser: A python parser for pubmed open-access xml subset and medline xml dataset xml dataset. *Journal of Open Source Software*, 5(46), p.1979.

[25] Tulsyan, A., Garvin, C. and Ündey, C., (2018). Advances in industrial biopharmaceutical batch process monitoring: Machine-learning methods for small data problems. *Biotechnology and bioengineering*, 115(8), pp.1915-1924.

[26] Cheng, B., Zhu, D., Zhao, S. and Chen, J., (2016). Situation-aware IoT service coordination using the event-driven SOA paradigm. *IEEE Transactions on Network and Service Management*, 13(2), pp.349-361.

[27] Fikri, N., Rida, M., Abghour, N., Moussaid, K. and El Omri, A., (2019). An adaptive and real-time based architecture for financial data integration. *Journal of Big Data*, 6, pp.1-25.

[28] Tamraparani, V., (2020). Automating Invoice Processing in Fund Management: Insights from RPA and Data Integration Techniques. *Available at SSRN 5117121*.

[29] Kaoudi, Z. and Quiané-Ruiz, J.A., (2018), April. Cross-platform data processing: use cases and challenges. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)* (pp. 1723-1726). IEEE.

[30] Zelenkauskaite, A., (2017). Remediation, convergence, and big data: Conceptual limits of cross-platform social media. *Convergence*, 23(5), pp.512-527.