The Proceeding of the 4th International Conference in Information and Communication Technologies (ICICT2022)
15th – 16th December 2022 - Lusaka, Zambia

Decoupling of Heterogeneous Systems using the Microservices Architectural Model

Mike Mudimba[1] and Jackson Phiri[2]
Computer Science Department
The University of Zambia
Lusaka, Zambia
Email: mikemudimba1000@gmail.com[1], jackson.phiri@cs.unza.zm[2]

***Abstract* — most higher learning institutions have implemented sophisticated ecosystems with ICT solutions at the center of operations in order to advance their business needs. ICT solutions primarily are used in the access, storage and retrieval of data across different platforms and or locations. To achieve this effectively, data needs to be standardized, easy to access and secure. Distinct systems are normally developed with the task of handling different kinds of processes. This in turn causes major communication challenges in how distinct systems communicate and share information with one another. In this project we make use of an open source Enterprise Service Bus (ESB) which implements a micro services architectural model. We use Apache Kafka to decouple data from different applications (Single Sign in Authentication System, Postgraduate Project Submission System and UNZA Human Resource Management System). This decoupling of data and applications (Achieved through use of virtual servers known as brokers) ensures strengthened security and resource utilization than what otherwise exists in a normal ICT ecosystem.**

Keywords — Enterprise Service Bus, Broker, Micro services, Event driven architecture.

1. Introduction

Decoupled, or decoupling, is a state of an IT environment in which two or more systems somehow work or are connected without being directly connected. The Higher Education and TEVET sectors boasts of a myriad of systems which range from information systems, financial systems, ecological systems, computer systems to education systems [1]. The challenge arises because such heterogeneous systems built using different standards or technologies and operating in different environments cannot effectively share data. Using the micro services architectural model, this study aims to decouple heterogeneous systems within the sector to ensure improved efficiency, performance, transparency and security [1].

The advent of computer software technology in the higher education sector, the management of information seen significant improvement in the recent years. Computer software has been used to address how student information is captured, stored and accessed. Higher institutions of learning normally have different systems to handle different types of processes. This has posed a challenge in the manner in which systems share information. This in turn causes major communication challenges in how distinct systems can communicate and share information with one another. Using an Enterprise Service Bus or Event Streaming Service to ensure systems share data is the solution to the above outlined challenge.

A **micro services architecture** is a type of application architecture where the application is developed as a collection of services [3]. It provides the framework to develop, deploy, and maintain micro services architecture diagrams and services independently. An **enterprise service bus (ESB)** is a software platform used to distribute work among connected components of an application. It is designed to provide a uniform means of moving work, offering applications the ability to connect to the ESB and subscribe to messages based on simple structural and business policy rules [3].

As such, it's a tool that has use in both distributed computing and component integration. The best way to think of this tool is to visualize it as a set of switches that can direct a message along a specific route between application components based on message contents and implementation of business policies [3].

On the other hand, the essence of **event-driven architecture EDA** is that when you publish an event you don't wait for a response. The event broker "holds" (persists) the event until all interested consumers accept/receive it [4]. An **event broker** is middleware software, appliance or SaaS used to transmit events between event producers and consumers in a publish-subscribe pattern. According to Gartner, event brokers "are middleware products that are used to facilitate, mediate and enrich the interactions of sources and handlers in event-driven computing."

2. Related Works

This section of the paper gives a review of literature that focuses on similar research work as well as how similar challenges have been addressed elsewhere. The literature review mainly focuses on the Micro services Architectural Model and how it has been applied to decouple monolithic systems.

Zara Wajid Butt, [12] uses micro services to try and secure micro service communication between heterogeneous service meshes in order to sort out establishing a trust anchor between the different service meshes that want to expose their services to each other. Especially when the two entities do not share a common root CA. He studies the existing service meshes and their federation mechanisms and identifies the challenges faced when federating heterogeneous service meshes which leads him to a solution that overcomes the problem of achieving federation between heterogeneous meshes and evaluate it against the identified security requirements.

Swarna kamal Paul, [13] solves real world heterogeneous tasks with micro services through use of AI. He solves problems of low expressivity, modularity, reusability, and abstraction which integrated systems consume and alleviates these challenges by integrating AI components as micro services. Along with loose coupling, the integration platform also allows tight coupling of components to solve complex heterogeneous tasks without compromising modularity. The usability of the proposed method demonstrates that solving a heterogeneous maze problem where an agent needs to solve a reinforcement learning environment, a machine vision task, and integrate them. Solving a speech recognition problem demonstrates modularity and functional abstraction of the method. Qualitative comparison with current state-of-the-art methods based on these use cases justifies his claims.

In another study, Valdemar Lipenko, Sebastian Nigl, Andreas Roither-Voigt and Zelenay David Operationalize Heterogeneous Data-Driven Process Models for Various Industrial Sectors through Micro service-Oriented Cloud-Based Architecture [14]. They use modern machine learning capabilities to predict future production quality outcomes, model predictive control to better account for complex multivariable environments of process industry, Bayesian Networks enabling improved decision support systems for diagnostics and fault detection. They achieve this through integration of highly heterogeneous models in a holistic system, which would also be suitable for applications from the most different industries. The core elements of the underlying solution architecture constitute highly decoupled model micro services, ensuring the creation of largely customizable model runtime environments. The then deploy isolated user-space instances, called containers, further extending the overall possibilities to integrate heterogeneous models. Strong requirements on high availability, scalability, and security are satisfied through the application of cloud-based services [14].

Locally, we took a look at the ZamServices portal and the Government Eservices GSB. The two are electronic catalogs of public electronic services provided by authorities to the citizens, the business environment and to the visitors of Zambia. The purpose of the portals is to provide governmental authorities/agencies' public services in an electronic way, so that portal visitors can browse, find and apply for a service on-line, instead of going physically to request for the service on-premises. In a way, these portals are heterogeneous systems which make use of similar services such as authentication, authorization etc. to make it easier to access information online.

In his paper, Liqiang Chen [15] describes micro services as an architectural style that focuses on discrete services instead of a monolithic design. Service Oriented Architecture utilizes micro services to decouple monolithic designs. Service-Oriented Architecture (SOA) has emerged as an architectural approach that enhances the service delivery performance of existing traditional systems while still retaining their most important features. This approach, due to its flexibility of adoption, has gained the attention of both academic and business entities, especially in the development of world-leading technologies such as Cloud Computing (CC) and the Internet of Things (IoT) [6].

The reviewed literature broadly outlines how micro services have been used to implement decoupled system services. The challenge though is that none of the techniques have been used in the higher education and TEVET sectors. Our model though will be built on the idea of interconnection systems within the higher education and TEVET sectors using micro services.

3. Methodology

3.1. Research Structure

Our goal is to develop a micro services architecture model which will be used to decouple heterogeneous systems in the higher education sector which need to seamlessly communicate and share data.

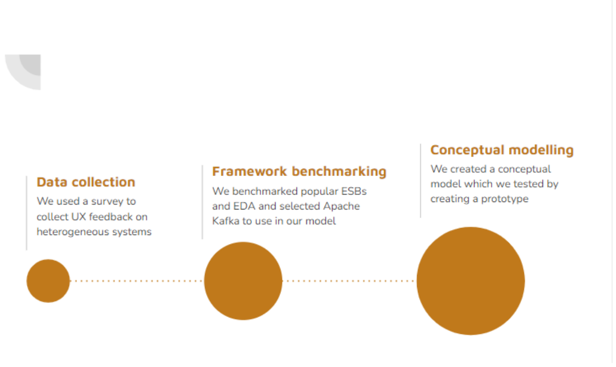

Data collection
We used a survey to collect UX feedback on heterogeneous systems
Framework benchmarking
We benchmarked popular ESBs and EDA and selected Apache Kafka to use in our model
Conceptual modelling
We created a conceptual model which we tested by creating a prototype

Fig. 1. Structure of the research.

Heterogeneous systems within the Higher Education Sector lack mechanisms by which data can be shared effectively in real time. This causes redundancy in processes because of inefficient use of resources and duplication of data.

We sort out the above challenge by setting and achieving the following objectives.

a) Assess the challenges the Higher Education and TEVET sectors face in capturing, storing and retrieving data in distinct systems that are tasked to handle different institutional processes.

b) Based on the results in point (I), develop a micro services architecture model using an open source Enterprise Service Bus or Event Driven Architecture to improve system communication and data sharing.

c) To develop a prototype based on the model in (II) which decouples heterogeneous systems.

Data Collection

To collect UX data, we used an online survey which we sent out to Higher and TEVET learning institutions. The collected UX data was then analyzed using inferential analysis and the results used to formulate a hypothesis.

Framework Benchmarking

We reviewed 20 popular open source and enterprise based frameworks (EDAs and ESBs) and benchmarked them using internal, external, performance, strategic and competitive methodologies to come up with the best suited framework to use in our model.

Conceptual Modelling
We took a look at popular architectures and designs of models and come up with an architecture which uses Apache Kafka as the core.

Prototype Development and Testing
Using the Agile software development methodology we designed and developed the UNZA Post Graduate Research Submission System, also the UNZA HRMIS. The two systems will be used to test the effectiveness of our model using UX testing, Unit Testing, Integration Testing, System Testing, and Performance Testing.

4. METHODOLOGY AND SYSTEM DESIGN

3.2. Model Design

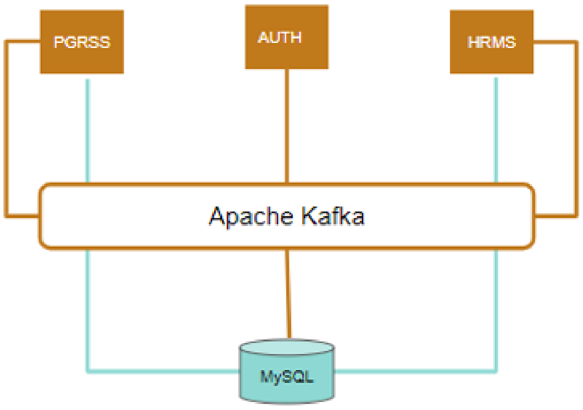


Fig. 2. Model architectural framework.

In order to decouple applications, we use an Event Streaming Framework to connect heterogeneous systems together using micro services. Some micro services will produce data while others will consume data. This ESF will act as middleware which will ingest data and from producers and publish it to consumers. Connectors will be used to link producers and consumers [9]. Any system which requires communication with other systems can simply be plugged to ESF.

3.3. Model Architecture

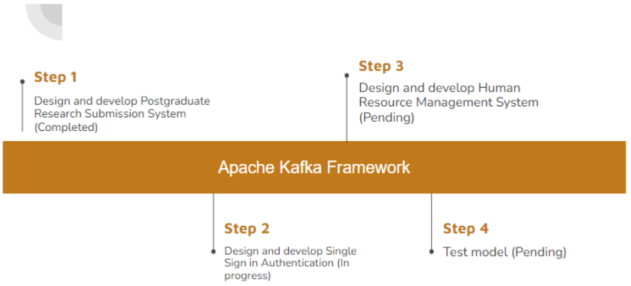

Step 1
Design and develop Postgraduate Research Submission System (Completed)
Step 3
Design and develop Human Resource Management System (Pending)
Apache Kafka Framework
Step 2
Design and develop Single Sign in Authentication (In progress)
Step 4
Test model (Pending)

Fig. 3. Conceptual model.

3.4. Results and Findings

Data collected from the TEVET sector (TEVETA, Ministry of Technology and Science VET and 29 registered MoTS TEVET institutions) reveal that heterogeneous systems pose a challenge when it comes to UX. More data is being collected from Higher Education sector to strengthen this hypothesis.

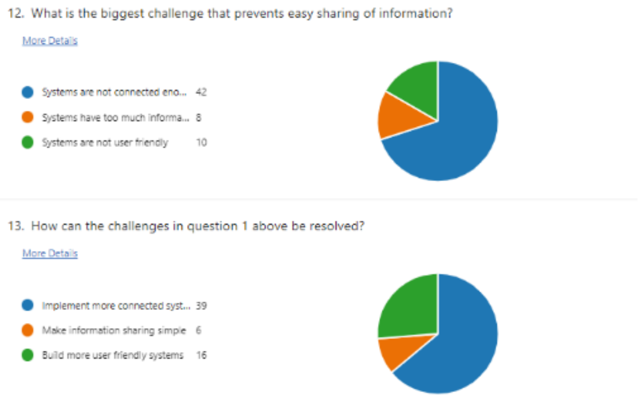


Fig. 4. Survey results sample.

In the first question which aims to find out what is the biggest challenge that prevents easy sharing of information, 70% of the sample says "unconnected systems" is the biggest challenge. In the second question which aims to find out the best way of addressing the challenge, 65% suggest that implementing interconnected systems is the solution. This clearly shows the challenge people are facing in higher learning institutions. Our model will therefore be built to address the aforementioned challenge.

3.5. Prototype Design

This section outlines how the prototype is designed and which technologies have been used to develop and implement. The prototype is developed around a concept of a postgraduate research submission system which allows students submit their work efficiently online.

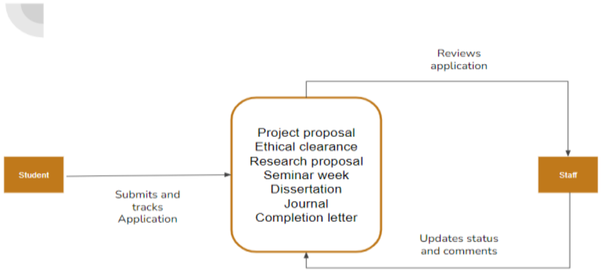


Fig. 5. Conceptual design of model.

The prototype is a web based application called Postgraduate Research Submission System built atop the Laravel framework. This application will be used in the storage and management of student research information and will use Apache Kafka to effectively communicate and transfer massages within a cluster of other systems.

5. Results

4.1. Prototype

The PGRSS is a web based system built on top of the Laravel framework. It can be used to submit and track research work for postgraduates. Users include students, supervisors, heads of departments, coordinators, deans, assistant deans PG and assessors.

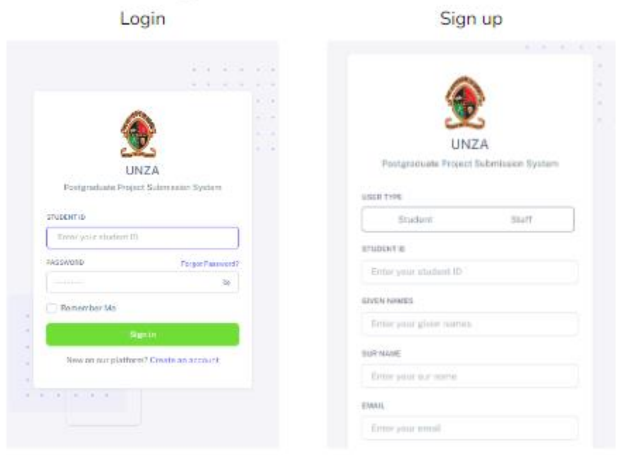


Fig. 6. Login and Signup pages.

The function of the PGRSS are outlined below detailing the functions of both students and members of staff.

- **Students can submit**
  - ○ Project proposals
  - ○ Ethical clearance
  - ○ Research proposal
  - ○ Seminar week
  - ○ Dissertation
  - ○ Journal
- **Students can get**
  - ○ Completion letter
  - ○ Comments
  - ○ Notifications
  - ○ Tutorials
  - ○ Form templates
- **Students have**
  - ○ Tracking capability
  - ○ Account management

- Student can apply and track applications. Students can update or retract applications which have not been approved.

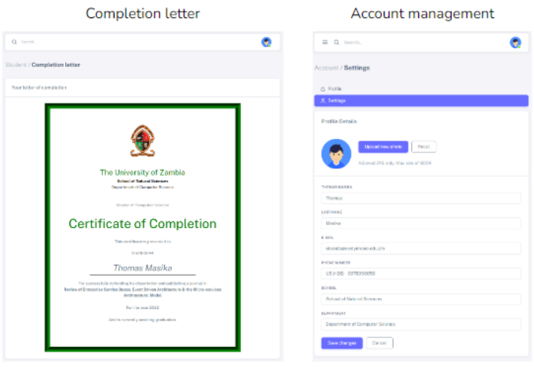


Fig. 7. View completion letter.

- Member of staff dashboard. Menu is automatically generated as per role assigned. Staff can view student projects assigned to them. Staff can communicate with students via sending comments.

- Staff can generate reports of applications received. Staff can filter report results by school, department, programme, level of training, mode of training and intake. Staff can export report to Excel or PDF.

- Staff can view application by section applied for. Staff can add comments to assist student. Staff can update status of application section. Staff can print application section details.

- System is capable of automatically generating letters of completion once student completes all stages of their research. System acts as student records management system. System has account management tools.

The following are some images of the prototype system that can be accessed online using a web browser on the following address.

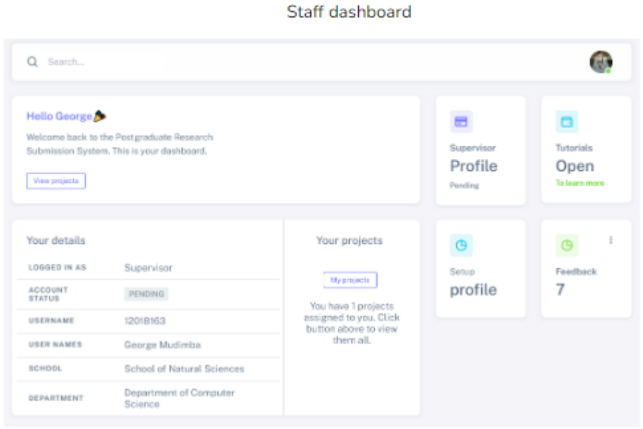
(http://155.0.15.193/pgrss/public/auth/login).

Security

Security will be achieved through the use of authorization and authentication techniques between shared messages. Authentication can be enabled between brokers, between clients and brokers and between brokers and ZooKeeper. Auditing is also another important feature that will enable us to keep track of all transactions that will be happening within the cluster. Lastly, encryption will be used in the form of SSL or TSL to encrypt all traffic between communicating entities.

Resource Utilization

ESF achieves low latency message delivery through Sequential I/O and Zero Copy Principle. The same techniques are commonly used in many other messaging/streaming platforms. Zero copy is a shortcut to save the multiple data copies between application context and kernel context.


Staff dashboard

Hello George

Welcome back to the Postgraduate Research Submission System. This is your dashboard.

Your details	
LOGGED IN AS	Supervisor
ACCOUNT STATUS	PENDING
USERNAME	1201B163
USER NAMES	George Mudimba
SCHOOL	School of Natural Sciences
DEPARTMENT	Department of Computer Science

Your projects

You have 1 projects assigned to you. Click button above to view them all.

Fig. 8. Member of staff Dashboard.

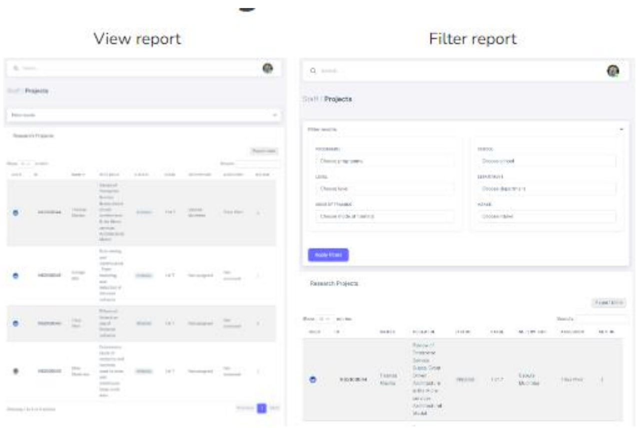
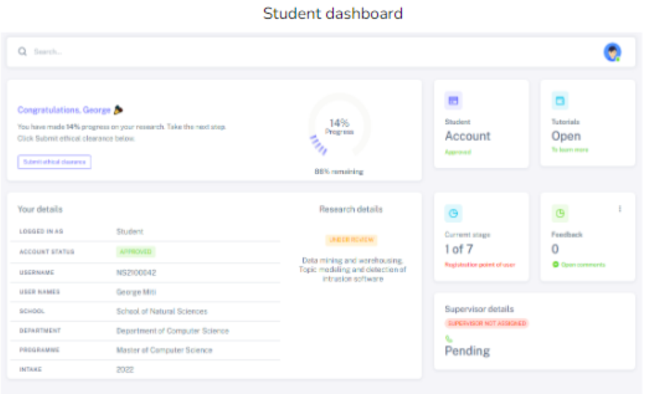


Fig. 9. Viewing and filtering reports.


Student dashboard

Congratulations, George

You have made 14% progress on your research. Take the next step. Click Submit ethical clearance below.

Your details	
LOGGED IN AS	Student
ACCOUNT STATUS	APPROVED
USERNAME	NS2100042
USER NAMES	George Miti
SCHOOL	School of Natural Sciences
DEPARTMENT	Department of Computer Science
PROGRAMME	Master of Computer Science
INTAKE	2022

Research details

UNDER REVIEW

Data mining and warehousing. Topic modeling and detection of intrusion software

Fig. 10. Student Dashboard.

6. Discussion

5.1. Recommendation

The better choice of ESB to implement depended on the type of systems we built and their purpose. Some of the factors we considered can be as follows;

Table 1 Attributes of Enterprise Service Bus

Scalability	Amount of services you intend to scale your system with.
Load per service	How many people will be accessing each of the deployed services?
Infrastructure available	Distributed or single server infrastructure.
Resources	Server resources available such as memory, connections and storage
Reusability	Use of existing assets within the software for new tasks.
Security	How secure is the platform to data leaks and sabotage.
Cost	Is it open source or proprietary? How much it costs per use.
Ease of use	How is it to learn and interact with for the developers and administrators?
Protocol support	It should support a wide array of protocols to connect to
Efficiency	How efficient is the ESB in utilization of available resources

The best ESB as per recommendation based on the attributes listed in the table above is WS02. This is because:

- It is open source
- Is well documented
- Large online community support
- Efficient
- Secure
- Scalable

On the other hand, sometimes instead of going full-fledged to use an ESB, it is easier and simpler to implement an event broker. This entirely depends on the output you require to achieve. Event brokers have the following advantages over ESBs.

- Loose coupling
- Fault tolerance
- Reduced technical debt

Apache Kafka in this regard would be a great choice as it is robust, fault tolerant and extremely efficient (See section on EDA above). Apache Kafka is also widely supported, open source and easier to implement [6].

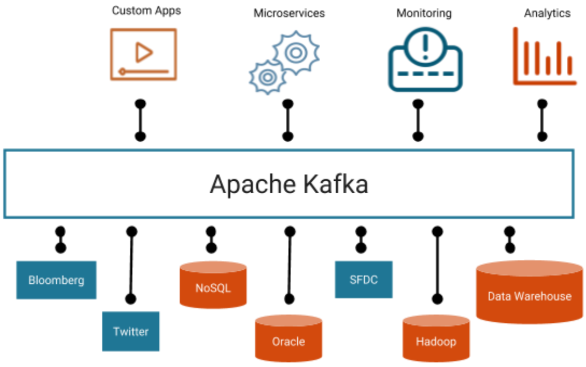


Fig. 11. Apache Kafka Framework.

5.2. Conclusion

The goal is to assess challenges faced by the Higher Education sector in the capturing, storing and retrieval of data and how this can be resolved by use of a micro services architectural framework which utilizes an Enterprise Service Bus or Event Driven Architecture solutions. A survey was conducted to assess quality of service in heterogeneous systems using a conceptual framework. It was found that 70% of the sample suggested none interconnected systems pose a communication challenges. 65% of the sample suggested that system integration is the best method to address the aforementioned challenge. A prototype was developed in the form of a Postgraduate Research Submission System which sits atop an Apache Kafka and publishes messages that can be used by other heterogeneous systems real time. Communication is something that is in the need of every people in the world. This matter is what drive the telecommunication industries to upgrade their services to accommodate the orders of the customers. Thus, Service Oriented Architecture (SOA) with the Enterprise Service Bus (ESB) is the solution for the problem [8], because it is able to be flexible and follows the ever changing business process and needs of the customers. With the micro services, the system will be able to be modified partly on what needs to be added or fixed, it does not require the service to be re-built from scratch. The implementation of the new system based on ESB is a new approach that can be considered for ordering system, the implementation in this case is able to meet the management's expectation of success rate.

7. References

[1] Techopedia, "Decoupled", November 16, 2021. (View source material)

[2] Ortiz S. "Getting on board the enterprise service bus" Computer (2007)

[3] Namiot D.Sneps-Sneppe M, "On Microservices Architecture". International Journal of Open Information Technologies (2014)

[4] Dunkel J.Fernández A.Ortiz R.Ossowski S., "Event-driven architecture for decision support in traffic management systems", Expert Systems with Applications (2011)

[5] Omer AzizMuhammad Shoaib FarooqAdnan AbidRubab SaherNaeem Aslam, "Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study", IEEE Access (2020)

[6] Bhole Rahul HiramanM. Chapte VireshC. Karve Abhijeet, "A Study of Apache Kafka in Big Data Stream Processing" 2018 International Conference on Information, Communication, Engineering and Technology, ICICET 2018 (2018)

[7] Ahmed Shaheen, Apache Kafka: Real Time Implementation with Kafka Architecture Review Javed International Journal of Advanced Science and Technology (2017)

[8] A Systematic Mapping Study on Microservices Architecture in DevOps Muhammad WaseemPeng LiangMojtaba Shahin Journal of Systems and Software (2020)

[9] Security in microservices architectures Nuno Mateus-CoelhoManuela Cruz-CunhaLuis Gonzaga Ferreira Procedia Computer Science (2021)

[10] From monolithic architecture to microservices architecture Lorenzo De Lauretis Proceedings - 2019 IEEE 30th International Symposium on Software Reliability Engineering Workshops, ISSREW 2019 (2019)

[11] Microservices architecture: Challenges and proposed conceptual design Raja Mubashir MunafJawwad Ahmed[...]Tauseef Rana 2019 International Conference on Communication Technologies, ComTech 2019 (2019)

[12] Zara Wajid Butt, "Secure microservice communication between heterogeneous service meshes", Aalto University, 29.07.2022.

[13] Swarna kamal Paul, "On Solving Heterogeneous Tasks with Microservices", October 2021Journal of The Institution of Engineers (India) Series B 103(4) DOI:10.1007/s40031-021-00676-5

[14] V. Lipenko, S. Nigl, A. Roither-Voigt, and Z. David, "Operationalizing Heterogeneous Data-Driven Process Models for Various Industrial Sectors through Microservice-Oriented Cloud-Based Architecture", in AI and Learning Systems - Industrial Applications and Future Directions. London, United Kingdom: IntechOpen, 2021 [Online]. Available:

https://www.intechopen.com/chapters/72819 doi: 10.5772/intechopen.92896

[15] Sang M. LeeLiqiang Chen, "The impact of flow on online consumer behavior" Journal of Computer Information Systems (2010)