IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 1, Ser. 4 (Jan. – Feb. 2024), PP 48-58
www.iosrjournals.org

Database Migration Service With A Microservice Architecture

Zilani Kaluba[1], Mayumbo Nyirenda[2]
Computer Science Department, School Of Natural Sciences, The University Of Zambia, Lusaka, Zambia.

Abstract:
The study employed knowledge of microservices and existing strategies and tools for database migration. The migration framework incorporated microservices based on the microservice architecture, characterized by small and autonomous services. The database migration strategy served as a blueprint and guide in designing the entire migration framework for the process. Each Database Migration from a source Database Management System to a destination Database management system is treated as a microservice. A combination of smaller services was designed to work cohesively together and achieve the overall objective. Using the understanding of a detailed literature review on the existing tools and the microservice architecture, the results of the study examined the database migration process of transferring tables, data and constraints in the database schema using existing tools, and also developed a database migration framework that takes advantage of the microservice architecture. This can thus benefit several organizations in the database migration process, as the migration service removes the cost of redesigning a new database for new systems in a different database management system and reduces the risk of data loss while in transit .Further the framework can be used to implement services and Database migration strategy for Database migration between any other pair of Database migration Systems (Oracle, Microsoft SQL Server, and My SQL).

Keywords:*Database, Microservice, Migration, Architecture, Migration Strategy,Database Management System.*

Date of Submission: 14-02-2024 Date of Acceptance: 24-02-2024

I. Introduction

Database migration concerns migrating databases from one database management system to another database management system. According to Fairhurst [1] agile methodologies can help database migration projects produce business value faster and more efficiently while being adaptable to changing project goals.Database migration can be defined as the process of migrating a database from one or more database management system (DBMS) to one or more target database management system.

Indeed, with evolving technology, many researchers and organizations have stressed the importance of databases [2], [3] and often find the need to upgrade their systems, potentially requiring a migration to a different database environment. This process involves redesigning a new database using a different Database Management System (DBMS). The process can be intricate and involves the development of database migration scripts or the utilization of cloud-based or on-premises technologies for database migration. The main drivers that would make an institution to do database migration could be as a result of upgrading to the latest version of the database software to improve security and compliance, moving existing data to a new database to reduce cost, improve performance, and achieve scalability. Moving from an on-premises database to a cloud-based database for better scalability and lower costs .Merge data from several databases into a single database for a unified data view. The research was guided by research objectives. To examine the database migration process of transferring tables, data and constraints in the database schema using existing tools. To develop a migration framework that takes advantage of the microservice architecture. To Implement database migration services based on the developed migration framework.

II. Related works

Jesse Summan described the importance of having a database migration strategy for a database migration project. He defines database migration strategy serves as a blueprint and guides the entire migration process. Database migration strategies may vary depending on the need for migration, costs, the size of the business,

downtime tolerance, and other factors. Jesse Summan listed the factors to consider in a database migration strategy for a smooth and successful database migration [4].

The tools and technologies developed for RDB enable non-expert users to leverageeffectively. However, they cannot be directly applicable to NoSQL databases due tovarious migration challenges listed. NoSQL systems generally lackmigration tools and automation instead of demand expert knowledge of both databases.

Manual transformation relies on rules of thumb or heuristics, resulting in decreasedsystem performance. This section introduces the currently available migration tools inthe market. We have classified the tools into two categories: Academic research-basedand Industry-driven tools.

Academic research-based tools

Here we discussedsome research tools developed by academic researchers for SQL-to-NoSQL databases.Six tools are described in this category, namely R2NoSQL tool[5], Erwin Hawk [6], Digi browser[7], NoSQL migrator[8], R2G[9]and some metamorphose tools.The comparison of tools wasbased on parameters named as the supported source database, supported targetdatabase,migration mechanism, tool development language, andcustomizability.

Myller Claudino et al. proposed the R2NoSQL tool totransform the data from Relational and NoSQL document databases. Theauthors use MySQL as the source database and MongoDB as the targetdatabase. The data is mapped from the source model to the target model usinga conversion algorithm similar to a depth-first search. Java is used to developthe tool. On the other hand, the tool's availability is not specified and followsthe Semi-automatic mechanism[5].

Erwin Hawk :Tianyu Jia et al. developed the tool to migrate the schemaand data from MySQL to MongoDB with the help of query and datacharacteristics of the relational database. Description and actions tags aregenerated using these characteristics. The authors have proposed algorithms forschema and data transformation. The tool's availability is not specified andfollows the Semi-automatic mechanism [6].

Digibrowser: Girts Karnitis et al. migrate using a metamodeling approachusing two logical levels over the MongoDB physical data. It consists of twosteps for migration. The first step is to get the existing relational database'sphysical data structure and convert it to metadata; then, data is migrated usingthe metadata in the next step. The physical, relational tables are represented bya table-like structure (TLS). The elements of the TLS are classified into fourgroups (codifier, complex or straightforward entity, and N: N link), and thetranslation algorithm is used to generate the MongoDB physical schema fromthese table-based system [7].

NoSQL migrator ;Branko Terzi et al. created this tool. It usesMongooseDSL to migrate relational data to MongoDB. There are four stages tothe migration process: The first step involves reengineering the relationaldatabase using the IISREE too [8].

R2G : Roberto De Virgilio et al. convert relational data and queries to graphdata and queries at the application layer. It consists of four primary components:a metadata analyser that analyses relational schemas to create a schema graph,a data mapper, a query mapper, and a graph manager that migrates data andqueries from the data mapper and query mapper to the target database [9].

Industry driven tools

A single migration utility cannot handle all migration needs due to different data modelsof NoSQL databases. Different types of tools and utilities are available in the market tocope with this need. This paper introduces some of the tools named as Talend Open studio [10]Apache Sqoop [11],MongoDB tools [12],Apache drill [13]and IBM Spark Cloudant [14]. Different organizationsdevelop these tools, provide services in various fields like data integration, intelligentbusiness decisions, cloudcomputing, massive data management, and data migration, andfall under the semi-automatic category inmigration mechanisms.

Pentaho Data Integration :It was previously known as Kettle andprovides ETL capabilities. It provides users with data analysis, integration,migration, and mining solutions. It provides a graphical interface for datamigration regardless of the volume of data and eliminates the need forscriptwriting. It supports input and outputformats like CSV, JSON, and SQLtables [15]**.**

Altova MapForce is a visual data conversion and database migration tool with support for SQL Server, MySQL, PostgreSQL, SQLite, Firebird, Informix, Sybase, MongoDB, and more. Some of the features it has are, Database Connection Wizard, supports all major relational database types, Support for NoSQL databases Convert from one database type to another, Convert to XML, Excel, JSON, EDI, flat files. Map data from one

database type to another. easy to use, drag-and-drop data mapping. Ability to execute SQL statements prior to database output and define database table actions[16].

AWS Database Migration Service (AWS DMS) is a managed migration and replication service that helps you move your databases and analytics workloads to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database.AWS Database Migration Service enables both homogeneous migrations, such as from Oracle to Oracle, and heterogeneous migrations, such as from Oracle or Microsoft SQL Server to Amazon Aurora. You may also use AWS Database Migration Service to continuously duplicate data from any supported source to any supported target with low latency. For example, you can develop a highly available and scalable data lake solution by replicating data from numerous sources to Amazon Simple Storage Service [17].

Bi suggests a more refined method to web database security. She tackles some of the most pressing concerns about web databases but contends that standard security measures may no longer be adequate. She offers a "Web databases security server". which can do a variety of tasks in addition to regular database authentication [18].

Hudicka also gives a comprehensive summary of data migration processes. Although his breakdown differs slightly from that of others. The migration process from legacy systems based on hierarchical databases, according to Hudicka, must be planned especially carefully because many of these systems do not enforce referential integrity, and two cornerstones of this older structure – de-normalization and redundancy – are in direct contradiction to more modern relational theory. He then goes on to argue for a series of phases, each of which should be completed before moving on to the next [19].

III. Methodology

Baseline Study

Baseline Study was to establish the challenges that may hinder the process of database migration during the execution of a database migration project when using existing tools. A qualitative research method was used in this study.Capturing as much qualitative data as possible regarding how the existing tools are used to migrate database between different database management systems. This was also done through consultations and interviews with experts in database migration and review of related works done by several researchers.

Data Collection Methods

This research draws knowledge from already existing technological systems, and as such, the development of the new system mostly depended on the existing secondary data on the subject of database migration combined with the theoretical knowledge of the author. Secondary sources in form of published literature from journals, books and internet sources were used for this research. These were used in order to review what other people have done, or related work on the subject. However, the research also made use of primary data by interviewing experts or people that have used these database migration systems for a long time.
A Database Migration Service system will be developed to address the objectives listed. The method of analysis, design, and implementation of this project will be divided into four categories, guided mainly by the objectives of the study.

Understanding the Source Database and destination database management system is key. Before beginning any database, conversion project will need to understand the source data that will populate the target database. This will be done through consultations with experts in database migration and review of related works done by several researchers.

Some important points to be considered when understanding the source database will be; The size and complexity of the database to be migrated will determine the scope of the migration project. This will also determine the time and computing resources required to transfer the data. If the source database contains tables that have millions of rows, might require to use a tool with the capability to load data in parallel.

Capture as much qualitative data as possible regarding how the existing tools are used to migrate database between different database management systems, such as MS SQL database to an Oracle one and understanding the schema conversion capabilities to successfully execute a database migration project.

A framework composed of distinct steps or phases in the development of software systems to be used in the development of the Database Migration Service. The Software Development life cycle (SDLC) consists of five stages which include Planning, Analysis, Design, Implementation and Maintenance.

Systematic approach of organizing the best ways to develop systems efficiently, which includes the descriptions of work to be performed at each stage of the development process and drafted documents. In terms of the design approach the approaches to be used are data-oriented approach (DOA), the object-oriented approach (OOA) and the service-oriented approach (SOA)[20].

Microservices were used in this project, it relies on the microservice architecture which consists of a collection of small and autonomous services. Here each service is a self-contained and is implemented using a single capability within a bounded context. Each Database migration from a DBMS will be treated as a microservice. A combination of smaller services will be developed to work cohesively together and achieve the overall objective.

The overall purpose of the database migration service design is to provide efficiency and effectiveness in database migration process. Migrating a database together with its data, constraints, and database structure from source to a target database management system.

To achieve this process we used specific database management systems which are Oracle DBMS, MS SQL Server, and MySQL. The development of the service for the database migration process involves the exchange of data and constraints between different DBMS, running non-identical DBMS and backing up the database in flexible format such as JavaScript Object Notation (JSON).

Performances and Best Practices Methods

The design followed best practices by ETL tools for better development and performance. Here each ETL tool has few tips to increase the performance which can be reused with an understanding of the tool's capabilities can aid in this. The method of parallel processing and pipelining to allow data to be partitioned are essential in designing can improve to deal with large volumes of data in ETL.For Example a join in the database query across multiple tables with large volumes embedded inside a source/SQL stage in the tool can prove to be 'expensive' and may take time to return results. Instead, use the stages provided by the ETL tool to split this query and fully utilize the application database as well [21].

Database Migration Framework

We designed a Database Migration framework which takes advantage of the microservice architecture. The Framework follow the guidelines below in the implementation of a microservice architecture.
The following have been outlined as project milestones and therefore critical to the whole project.

- Database Migration Checklist
- Database Migration Strategy
- Database Migration Project Steps
- Database Migration Challenges
- Strategy to overcome database migration Challenges Strategy
- Rollback Strategy
- Tools for Database Migration Success
- Expertise and Skills for Database Migration
- Extract, Load and Transfer process
- Database Migration Microservice Architecture Design
- Internal Logical Design
- Interface Design

Database Migration Strategy

Database migration plan serves as a roadmap and guides the whole migration process. Database migration techniques vary based on the necessity for migration, expenses, the size of the organization, downtime tolerance, and other factors. A Database Migrations Strategy acts as a blueprint for the migration process. It is therefore important to formulate a strategy which guides the implementation of database migration. In this Framework the migration strategy is also key in the implementation of database migration service which takes advantage of the microservice architecture.

System Design and Implementation

Proposed Business Process

The proposed framework will have two end points , the source and target end points, these provide connection points to the source and target DBMS. Each service in the architectural is a self-contained relying on the microservice architecture which consists of a collection of small and autonomous services which interact with each other to achieve a common goal ,using a single capability within a bounded context.The designed framework involves the process of transferring data from one database system to another. This architecture

typically includes components such as source and target database management systems, data mapping, data extraction, transformation, loading processes and validation. The framework will also follow the guidelines in the database migration strategy for a successful database migration.

The system shall be able to provide the user with an interface that will enable the user to select the source database management system and destination database management system. The system shall be able to migrate a database from the source database management system to a destination database management system. Migration of the databases will be between the three major databases MySQL, SQL server, and Oracle database management systems. The system shall be able to migrate the records and the data details in the database from one database management system to another in a short period of time.

Software Quality Attributes

Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error messages appear when something goes wrong so to prevail availability problems. Checking that the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its modules.

Additional considerations of database migration service

The following sections briefly discuss non-functional aspects that are important in the context of database migration. These aspects include error handling, scalability, high availability, and disaster recovery.

Error handling

Failures during database migration must not cause data loss or the processing of database changes out of order. Data integrity must be preserved regardless of what caused the failure (such as a bug in the system, a network interruption, a VM crash, or a zone failure).

A data loss occurs when a migration system retrieves the data from the source databases and does not store it in the target databases because of some error. When data is lost, the target databases do not match the source databases and are thus inconsistent and incomplete.

Scalability

In a database migration, time-to-migrate is an important metric. In a zero-downtime migration (in the sense of minimal downtime), the migration of the data occurs while the source databases continue to change. To migrate in a reasonable timeframe, the rate of data transfer must be significantly faster than the rate of updates of the source database systems, especially when the source database system is large. The higher the transfer rate, the faster the database migration can be completed.

When the source database systems are quiesced and are not being modified, the migration might be faster because there are no changes to incorporate. In a homogeneous database, the time-to-migrate might be quite fast because you can use backup/restore or export/import functionality, and the transfer of files scales.

Design Specification

System High Level Overview Design

The High-Level design has two end points , the source and target end points, these provide connection points to the source and target DBMS. Each service in the architectural is a self-contained relying on the microservice architecture which consists of a collection of small and autonomous services which interact with each other to achieve a common goal ,using a single capability within a bounded context.The designed framework involves the process of transferring data from one database system to another. This architecture typically includes components such as source and target database management systems, data mapping, data extraction, transformation, loading processes and validation.

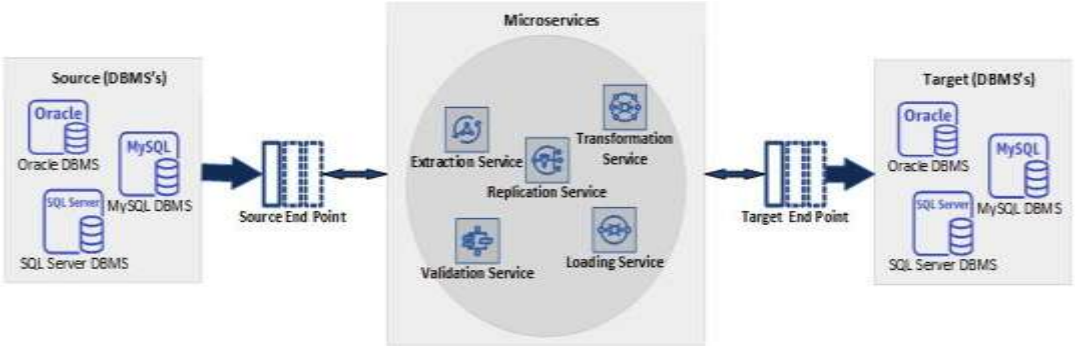

Figure 1. System High Level Overview Design

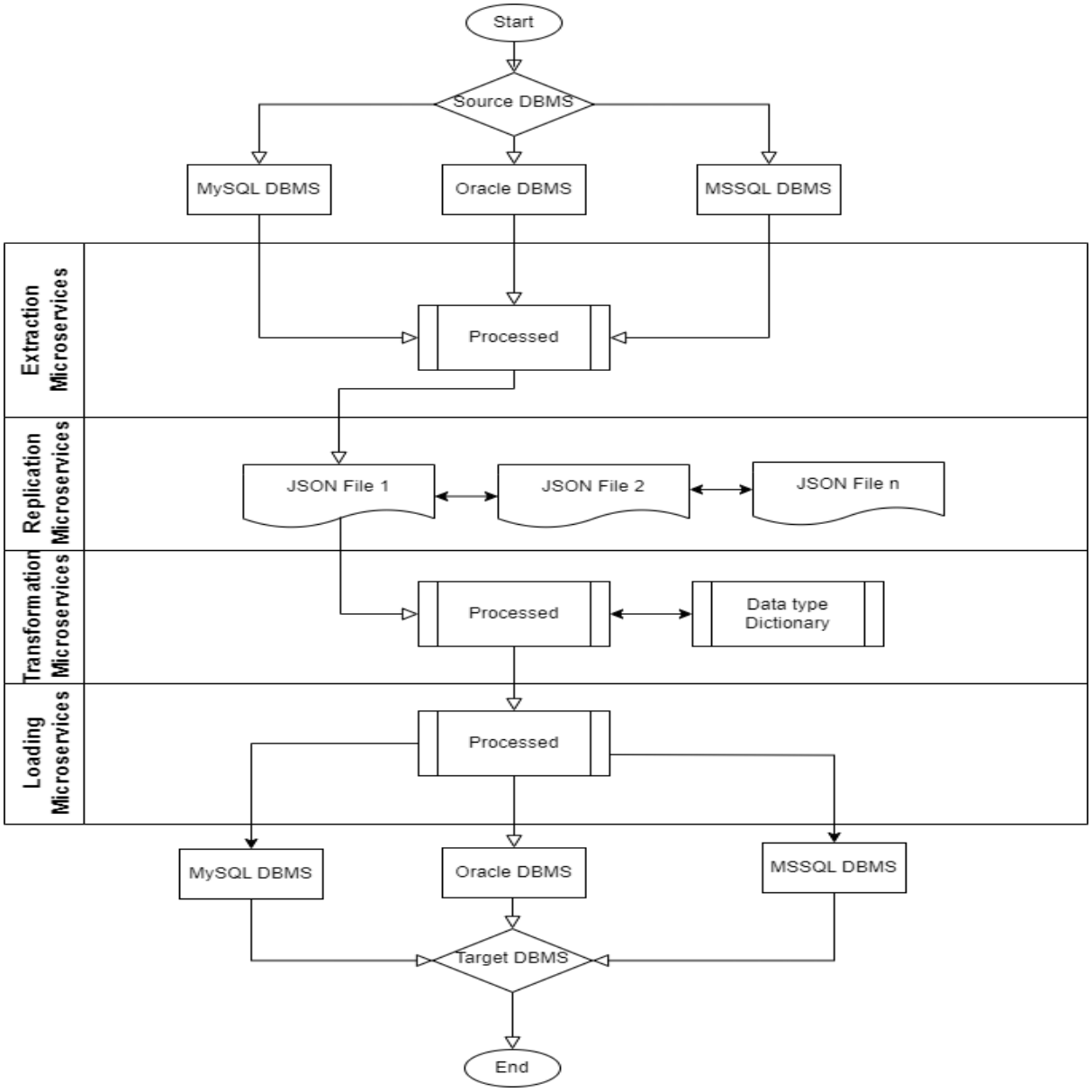

Figure 2. Logical Design.

The design for the migration framework comprises of microservices relying on the microservice architecture which consists of a collection of small and autonomous services. Here each service is a self-contained and was designed using a single capability within a bounded context. A combination of smaller services are designed to work cohesively together and achieve the overall objective which is migrating a database from source DBMS to target DBMS.

Firstly, the metrics service is executed which gets hold of the count of tables and records in each table from the source database. These metrics will later be used by the validation service for comparison with what has reached the destination DBMS after migration.

Then follows by invoking a microservice which extract the constraints from the source database and replicates these constraints in a JSON file by interacting with the constraints replication service.

Interface Design

The interface gives the user a friendly environment, which enables the user to use the system without any difficulties. Database migration will be a web-based service accessed through the browser.

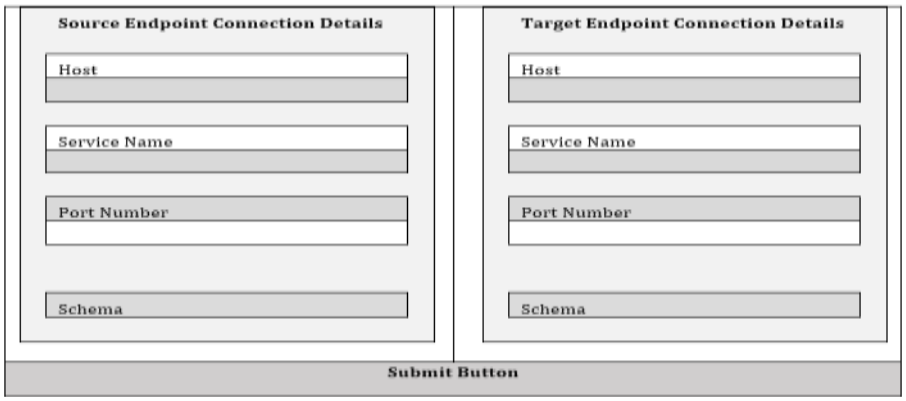

Figure 3. Interface Design

System Implementation

The design of database migration service was effectively implemented using the technologies elaborated in the research methodology and would operate using the following system features.

Java Spring tool Suite

Java spring Boot is an open-source Java-based framework It is designed to simplify the process of building and deploying production-ready applications with the Spring Framework. Spring Boot aims to provide a convention-over-configuration approach, reducing the need for developers to manually set up configurations and boilerplate code, allowing them to focus on writing business logic.

mysql:mysql-connector-java:8.0.22, this java library was used to connect MySQL database management system from java.

Oracle JDBC (ojdbc8.ja); this java library was used to connect Oracle database management system from java.

SQL server (mssql-jdbc-8.2.0.jreVERSION.jar); this java library was used to connect SQL server database management system from java.

MySQL DBMS

MySQL database management system was used to store database in MySQL format (.frm). MySQL database management system stores data in separate tables. The database structures are organized into physical files .The logical model, with objects such as databases, tables, views, rows, and columns.

MS SQL Server DBMS

Microsoft SQL Server is a database management system developed by Microsoft. As a database management system, it is a software product whose primary function is to store and retrieve data as requested by other software applications.in our system the SQL server database management system was used to store database in SQL server format (.mdf).

Oracle Database Management System

An Oracle database Management system is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. Oracle database management system was used to store database in Oracle database format (.dbf).

Data Type Dictionary

A Datatype dictionary was used to map datatypes for different DBMS's,it defines a collection of names, definitions, and attributes about data elements that are being used or captured in abetween database management systems.

Oracle data type	SQL Server data type	MySQL data type
BLOB	VARBINARY(MAX)	LONGBLOB
CHAR([1-2000])	CHAR([1-2000])	VARCHAR(MAX)
CLOB	VARCHAR(MAX)	VARCHAR(MAX)
DATE	DATETIME	DATETIME
FLOAT	FLOAT	FLOAT
INT	NUMERIC(p)	DECIMAL(p,s)

Table 1: Data type Dictionary

The designed for the migration framework was implemented using microservices relying on the microservice architecture which consists of a collection of small and autonomous services. Here each service is a self-contained and was designed using a single capability within a bounded context. A combination of smaller services are designed to work cohesively together and achieve the overall objective which is migrating a database from source DBMS to target DBMS.The final result of the system was the database in the format that the user specified when he/she was interacting with the system. The three formats MySQL database management system (.frm), SQL Server database management system (.mdf) and Oracle Database (.dbf).

IV. Results And Discussion

System Automation and Implementation Results

Firstly, the metrics service is executed which gets hold of the count of tables and records in each table from the source database. These metrics will later be used by the validation service for comparison with what has reached the destination DBMS after migration.

Then follows by invoking a microservice which extract the constraints from the source database and replicates these constraints in a JSON file by interacting with the constraints replication service.

Once Constraints are replicated, the constraints are dropped , which creates a temporal database of tables and records without constraints for easy transition.

Typically, database migration follows the Extraction, Transformation, Loading (ETL) process, so each ETL processed has been designed as s service. We have the extraction service which involves extracting the data from the tables of the source database management system, this service interacts with the replication service for copying the data from the tables into JSON files.

Transformation service transforms the source database data into a compatible format with the target database management system. It includes adding fields, reinstating the constraints, changing data formats, structure or types using a datatype dictionary.

Loading service involves loading the newly transformed data into the new database management system in small increments (batches). After loading all the data into the new database management system, the target DBMS needs testing to ensure the database migration is complete and data is consistent this is achieved by the validation service. Validation service interacts with the metrics services from the source and target DBMS to compare the metrics of the two databases. Validating the data in the target DBMS against the source DBMS .

Database Migration Framework With Microservices

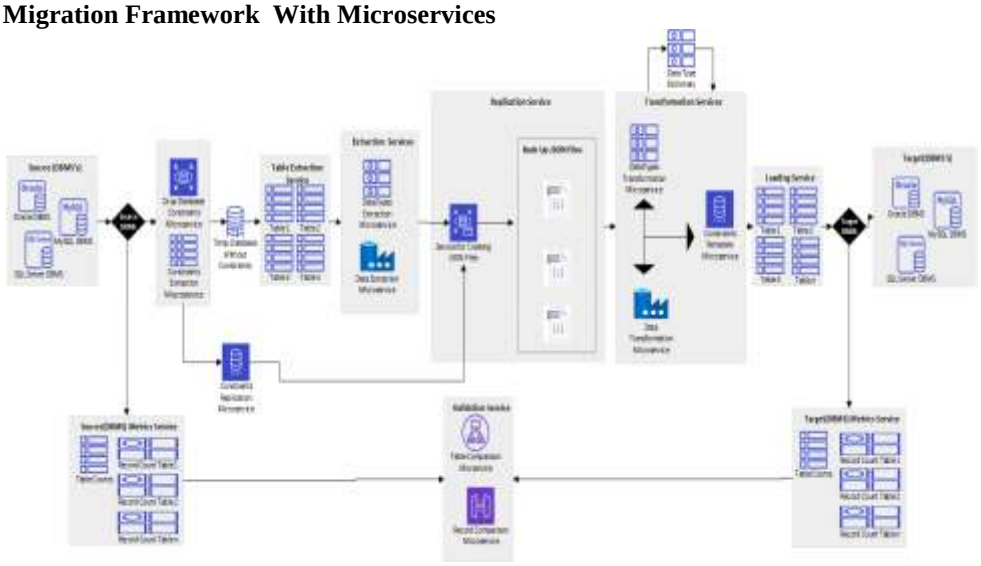

Figure 4. Database Migration Framework

System Implementation Results

Unit Testing

Unit testing involves the testing of individual components or modules of the system. Unit testing was done on individual modules to determine their functionality. There is a total of six modules in this system. Testing the classes for Converting a database to JSON fileThese classes are responsible for converting a database to an JSON file. These classes were tested separately to ensure that they are capable of reading the database, and converting the database to JSON file.

Sample Data from Oracle DBMS

```
SELECT* FROM EMPLOYEES
```

EMPLOYEE_NUMBER	FIRST_NAME	LAST_NAME
2401	JANE	BANDA
2402	MWANSA	CHANDA
2403	MULENGA	KALILO
2404	SILIO	KANGWA
2405	JONES	DAKA
2406	NAOMI	PHIRI
2407	CHILESHE	KALUBA
2408	MAIA	MTONGA
2409	MATEYO	LUNGU
2410	SERAH	TEMBO
2411	HELLEN	MUTALE
2412	RHODA	BANDA

Figure 5 Database Table Oracle

sample Snippet Data in JSON Format

```
[
        {
                "EMPLOYEE NUMBER": "2401",
                "LAST NAME": "BANDA",
                "FIRST NAME": "JANE"
        },
        {
                "EMPLOYEE_NUMBER": "2402",
                "LAST NAME": "CHANDA",
                "FIRST NAME": "MWANSA"
        },
        {
                "EMPLOYEE_NUMBER": "2403",
                "LAST_NAME": "KALILO",
                "FIRST NAME": "MULENGA"
        },
        {
                "EMPLOYEE_NUMBER": "2404",
                "LAST_NAME": "KANGWA",
                "FIRST NAME": "SILIO"
        },
        {
                "EMPLOYEE_NUMBER": "2405",
                "LAST_NAME": "JONES",
                "FIRST_NAME": "DAKA"
        },
        {
                "EMPLOYEE_NUMBER": "2406",
                "LAST_NAME": "NAOMI",
                "FIRST_NAME": "PHIRI"
        }
.....
]
```

Snippet JSON Datatype Dump

```
{
"columns" : [ {
"name" :"EMPLOYEE_NUMBER",
"dataType" :"VARCHAR2"
  }, {
"name" :"FIRST_NAME",
"dataType" :"VARCHAR2"
  }, {
```

```
"name" :"LAST_NAME",
"dataType" :"VARCHAR2"
  } ]
}
```

Sample Data from MySQL DBMS

```
SELECT* FROM EMPLOYEES
```

Result Grid | Filter Rows:

EMPLOYEE_NUMBER	FIRST_NAME	LAST_NAME
2401	JANE	BANDA
2402	MWANSA	CHANDA
2403	MULENGA	KALILO
2404	SILIO	KANGWA
2405	JONES	DAKA
2406	NAOMI	PHIRI
2407	CHILESHE	KALUBA
2408	MAIA	MTONGA
2409	MATEYO	LUNGU
2410	SERAH	TEMBO
2411	HELLEN	MUTALE
2412	RHODA	BANDA
2413	LUCY	LUNGU

Figure 6 Data from MySQL

	Testing scenarios	Expected results	Actual results
2	Mysql to json file	Json file	Mysql database was converted to jfon file
3	Sql to json file	Json file	Sql database was converted to json file
4	Oracle to json file	Json file	Oracle database was converted to json file

Table 2: Testing of classes from Database to JSON file

	Testing scenarios	Expected results	Actual results
2	Json file to mysql	Mysqldb	Json file was converted to mysqldb
3	Json file to sql sever	Sql server db	Json file was converted to sql server db
4	Json file to oracle	Oracle db	Json file was converted to oracle db

Table 3: Testing of classes from JSON file to Database

V. Conclusion

The migration framework comprising of microservices relying on the microservice architecture is developed. The designed framework is a comprehensive process that manages the migration process effectively and reduces risks involved in database migration.

Once the system is fully developed, the process will be fully automated which reduces the cost of database migration. The study identified challenges which hinders the process of database migration in a migrations project. The challenges like Data inconsistency, poor management of database migration process, lack of a database migration strategy, improper data mapping, and Failure to validate the database migration.

A framework was designed based on the model that was developed to provide efficiency and effectiveness in the database migration process after the challenges were identified.

We recommend that database migration service with microservice architecture should be implemented , and follow the guidelines as stipulated in the framework. Given the above findings of this research, it is recommended that the designed migration framework should be implemented and used. The designed migration framework addresses a number challenges that are associated with the database migration process. The implementation of the implementation of this framework will ensure that there is efficiency and effectiveness in

the application and use of the database management system. This will in turn translate to more productivity for the institutions that use these database management systems.

VI. References

[1] Sam. Fairhurst. Agile, "Agility in Data migration," Alfa, 2006.pg 14

[2] Simukanga, A., Phiri, J., Nyirenda, M., &Kalumbilo-Kabemba, M. (2018). E-Governance Systems: A Case Study of the Development of a Small-Scale Farmer Database. *Zambia ICT Journal*, *2*(1), 7–15. https://doi.org/10.33260/zictjournal.v2i1.41

[3] Simukanga, A.; Phiri, J.; Nyirenda, M.; Kalumbilo-Kabemba, M.M. ICT in Governance Systems: A Case Study of the FISP Farmer Registration System. In Proceedings of the Proceedings of the ZAPUC Internation Conference, Livingstone, Zambia, 29 April–3 May 2018

[4] Jesse Summan. https://streamsets.com. "Database Migration Strategy". February 2023

[5] Souza D, Carolina Salgado A, Claudino de Freitas M, Yluska Souza D. Conceptual Mappings toConvert Relational into NoSQL Databases. ICEIS 2016 - Proceedings of the 18th InternationalConference on Enterprise Information Systems. 2016;1:174–81.

[6] Jia T, Zhao X, Wang Z, Gong D, Ding G. Model transformation and data migration from relational database to MongoDB. Proceedings - 2016 IEEE International Congress on Big Data, BigData Congress 2016. 2016 Oct 5;60–7.

[7] Karnitis G, Arnicans G. Migration of Relational Database to Document-Oriented Database: StructureDenormalization and Data Transformation. Proceedings - 7th International Conference onComputational Intelligence, Communication Systems and Networks, CICSyN 2015. Oct;113–8.

[8] Terzić B, Kordić S, Čeliković M, Dimitrieski V, Luković I. An Approach and DSL in support ofMigration from relational to NoSQL Databases. In: 6th International Conference on InformationSociety and Technology ICIST. 2016.

[9] De Virgilio R, Maccioni A, Torlone R. R2G: A tool for migrating relations to graphs. In: Advances in Database Technology - EDBT 2014: 17th International Conference on Extending Database Technology, Proceedings. 2014. p. 640–3.

[10] T. O. S. [cited, "https://www.tutorialspoint.com/talend/talend_talend_open_studio.htm,"2021.

[11] S.c. 2021, "Sqoop - [Internet]. [cited 2021 Sep 14]. Available from: https://sqoop.apache.org/,"2021

[12] I. a. E. D. —. M. C. [Internet].,"https://docs.mongodb.com/compass/master/import-export/," 2021.

[13] D. I. A.D. [Internet]., "Drill Introduction - Apache Drill [Internet].," http://drill.apache.org/docs/drill-introduction/, 2021.

[14] h.Cloudant - Overview| IBM[Internet], "Cloudant - Overview | IBM [Internet]. https://www.ibm.com/cloud/cloudant," Available from https://www.ibm.com/cloud/cloudant, 2021.

[15] https://help.hitachivantara.com,"PentahoDataIntegration,"https://help.hitachivantara.com/Documentation/Pentaho/8.3/Products/Pentaho_Data_Integration, 2021.

[16] https://www.altova.com/mapforce, "Mapforce Database Migration Tool".

[17] Sait A, "Strategies for migrating Oracle Databases to AWS, Amazon Web Services" 2014.

[18] C. &. J. N. Bi, "A security paradigm for Web databases. ACM Southeast Regional Conference." 1999.

[19] J. R. Hudicka, "The Complete Data Migration Methodology" September 2004.

[20] https://microservices.io, "Miscroservice".

[21] S. S. Sarmah, "Data Migration," 2018.