SSL Progress Report

Team Members:

1. Nafiul Onim - 180042104
2. Mirza Sadaf Shahid - 180042137
3. Muhammad Rafsan Quayes - 180042141

Summary

Since our last meeting, we were tasked to study literature related to Microservice architecture. To get started we first read these two literatures:

1. “A Study on the Most ProminentAreas of Research in Microservices”
2. “Microservices in Practice: A Survey Study”

Next we studied another paper that studied two more papers that studied how to implement Microservice based LMS and an E-learning site.

1. “Microservice architecture in E-learning”
2. “MILMS: A Microservices-based Learning Management System”

“A Study on the Most Prominent Areas of Research in Microservices”

Authors: **Shamsul Sahibuddin, Mohammad Sadegh Hamzehloui, Ardavan Ashabi**
Journal: **International Journal of Machine Learning and Computing**
Year Published: **2019**
Link: https://www.researchgate.net/publication/333033011

This was simply a study to identify and discuss the main areas of debate regarding the domain. The said areas are as follows:

1. **The Exact Definition of a Micro-service:**
   They discussed in brief that this did not have an exact definition. They also consulted a number of other research papers and have concluded that microservices must have these characteristics:
   a. They must be small in size
   b. They must up-hold the single responsibility principle
   c. They must be loosely coupled
   d. Explicitly published interfaces

2. **Identifying the Size and Boundaries**
   Defining the right size and boundaries for microservices is another debatable area. They highlight papers that have very crude examples to illustrate that, like the number of lines of code or the amount needed to develop and deploy it. They argue that there is actually no definite metric that defines the size and boundaries and that developers are using practices like Service-oriented architecture (SOA) to help define their micro-services.They also discuss some of the mentioned practices.
   a. **Service-oriented architecture (SOA)**
      SOA is used to separate all services into components. It encourages loose coupling and reusability. The issue is that it encourages inter-service dependence. Independent services that can be tested and deployed on their own is a key characteristic of a micro-service.

   b. **Domain Driven Development (DDD)**
      The literature argues that this is a good way to start designing a micro-service architecture as the boundaries are defined with its bounded domain design techniques. The domains could be bounded by business logic or such.

   c. **DevOps**
      This is basically to integrate the development and deployment techniques. It tries to address the following issues:
      i. Scaling
      ii. Automation
      iii. Continuous Delivery
      iv. Monitoring

3. Technologies Used in Implementation and Deployment

The Paper also discusses some of the popular technologies used in developing Microservice architectures and tries to identify why.

a. Cloud

i. It is also an ideal environment to apply DevOps practices.
ii. It is built based on a set of loosely coupled components. This makes microservices' structure fit perfectly to the elasticity of the cloud.
iii. Provides dynamic and scalable solutions with more efficient resource usage.
iv. Many provide features such as IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service). These features enable developers to customize their infrastructure and platform mostly by coding.
v. Allows pay-on demand. Developers get flexibility when scaling up or down. On the other end, users can minimize the running cost.
vi. Cloud providers these days are also offering a set of tools to enhance monitoring, and automation processes.

b. Virtualisation

i. Common solution to tackle hardware limitations by allowing developers to provision and resize their machines at will.
ii. It is closely tied to Cloud Technology as most cloud providers provide scalable cloud solutions.
iii. Easy to isolate different environments.
iv. Easier to use practices like DevOp and continuous integration.

There are mainly two virtualisation types used in the industry: **VMwares** and **containers**. VMware was the traditional method of virtualization that had been around for years. Containers on the other hand, are a new trend despite being around for a long time. Their recent popularity is due to the new software named Docker (more on that in the next section).

c. Docker

Docker is a new software that is taking advantage of the Linux containerization feature to create an isolated environment. The reasons why it is so popular is listed below:

i. Very light-weight.
ii. Dockerfile - List of commands to store Docker Images. These can be stored in the form of text which then makes it possible to version control them.
iii. Docker Images - Docker images are built based on the instructions in the
iv. Dockerfiles. They can be shared within the community using Docker repositories. These repositories can be both public and private.
v. Docker Ecosystem - There are a lot of different technologies built around it, like monitoring tools, rolling updates and automated scaling. This allows companies to run microservices easily.

“Microservices in Practice: A Survey Study”

Authors: **Markos Viggiato, Ricardo Terra, Henrique Rocha, Marco Tulio Valente, Eduardo Figueiredo**

Conference: **VI Workshop on Software Visualization, Evolution and Maintenance, São Carlos - Brazil**

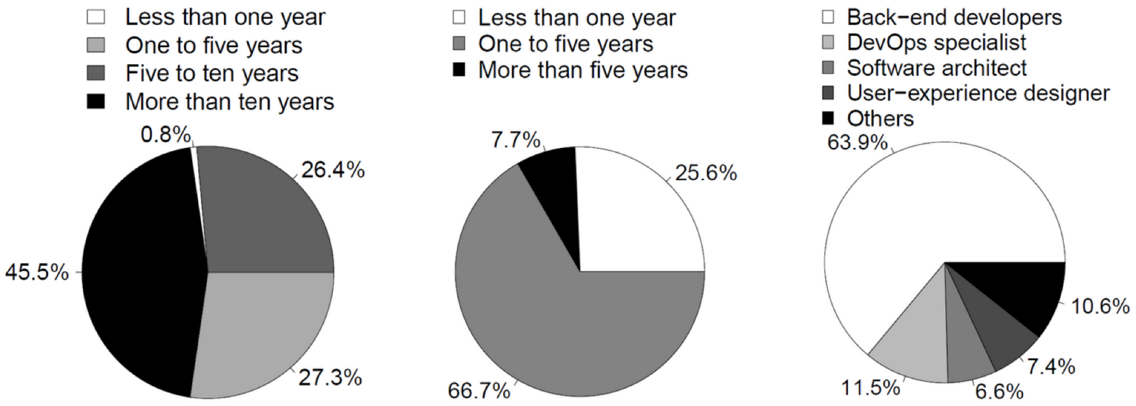
Year Published: **2018**

Link: https://www.researchgate.net/publication/326610698

This is a survey study to identify how the industry uses microservice architecture as well as to identify its advantages and disadvantages from the people directly involved.

Participants’ Background

The participants of the survey were first assessed with a few basic questions to determine their background. The results are shown in Figure 1.



(a) Development experience

(b) Microservices experience

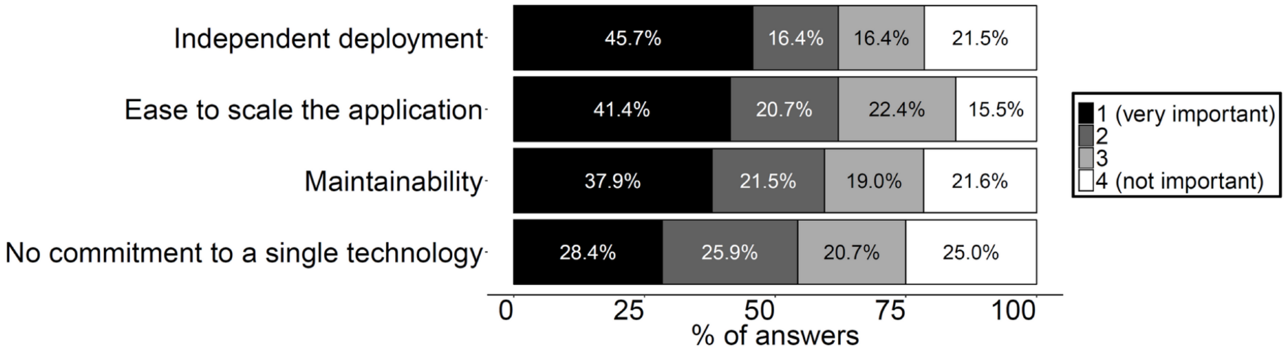
(c) Participant’s role

Figure 1. Participants’ background.

Almost 72% participants have at least 5 years of experience with software development and about 7.7% participants have at least 5 years of experience with microservices but about 66.7% participants have about 1 to 5 years of experience with microservices. In addition, 64% of the respondents are back-end developers and 11.5% are DevOps specialists.

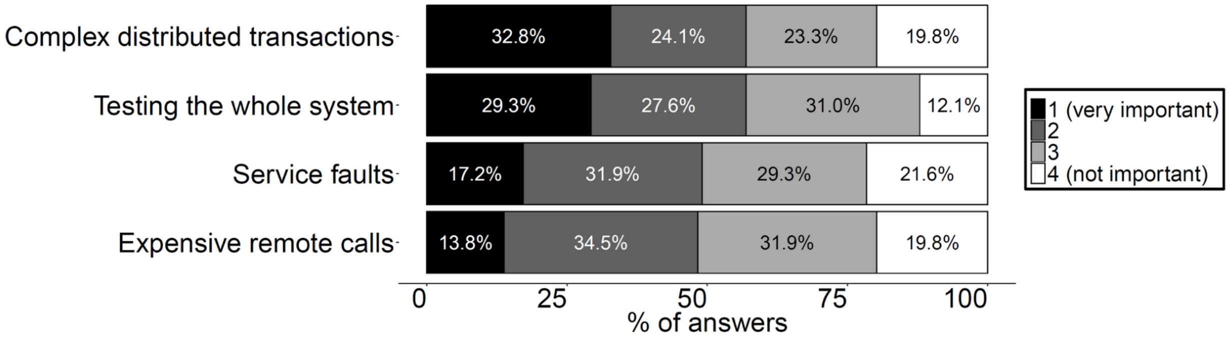
Advantages

When asked about the advantages of using microservices, more than 55% participants agreed that Independent Deployment, Ease to scale & Maintainability are very important advantages. But the last point has an even split; i.e. 50% participants deem it important and the other don’t agree that it's an important advantage. The total results are shown in the chart below



Disadvantages

Similarly the four challenges presented were Complex distribution transactions, Testing the whole system, Service faults and Expensive remote calls. The whole distribution of results is shown below.



Limitations

1. The122 participants may not represent the entire population of microservice practitioners.
2. There may have been terms in the survey that some software engineers in certain regions were unfamiliar with. For example, the term DevOps may have been new to some practitioners in South Asia.

Conclusion

Industry practitioners find certain aspects beneficial:

- Deployment
- Independent
- Ease to Scale
- Maintainability
- No commitment to a single technology stack

However, they also find some aspects to be challenging:

- Complex distributed transactions
- Testing the whole system
- Service Faults
- Expensive Remote Calls

“Microservice Architecture in E-learning”

Author: **Ana Milovanović**
Conference: **E-Business Technologies Conference 2021**
Year Published: **2021**
Link: https://ebt.rs/journals/index.php/conf-proc/article/view/81

The aim of this paper is to present a possible solution of a custom microservice architecture that is based on integration with some existing LMS. The approach that was used to identify microservices is Domain Driven Design. Integration of the Cloud Data Platform into this architecture is also considered.

The paper considered a number of existing cases and determined that the integration of the custom microservices with some existing LMS is lacking. The goal of this paper was to provide a possible solution of the custom microservice architecture that will be part of the e-learning flow and that is based on integration with some existing LMS (e.g. Moodle). Also, possible design of Cloud Data Platform and its place in microservice architecture is presented. In order to identify microservices Domain Driven Design should be considered, which approaches based on Bounded Contexts and breaking down systems into smaller parts. The API Gateway and Message Broker are also used.

Most Commonly Used Architectural Patterns in Microservice Architecture

These patterns were divided into three different
groups according to their purpose:

- Orchestration and coordination patterns - deal with logical communication and coordination of microservices.
- Deployment patterns - hosting is done via containers or virtual machines.
- Data Management Patterns – especially Data Storage options.

Essential Services

The LMS should provide standardized services, some of those standardized solutions are:

1. **SCORM (Sharable Content Object Reference Model).**
   This is a set of specifications for creating and sharing e-learning content.
2. **xAPI** for the metadata on learning analytics
3. **LTI (Learning Tools Interoperability) protocol**
   This is for authentication and assessment.

The researchers also pointed out that management of user and course-related information, report generation and analysis are also important.

The Design

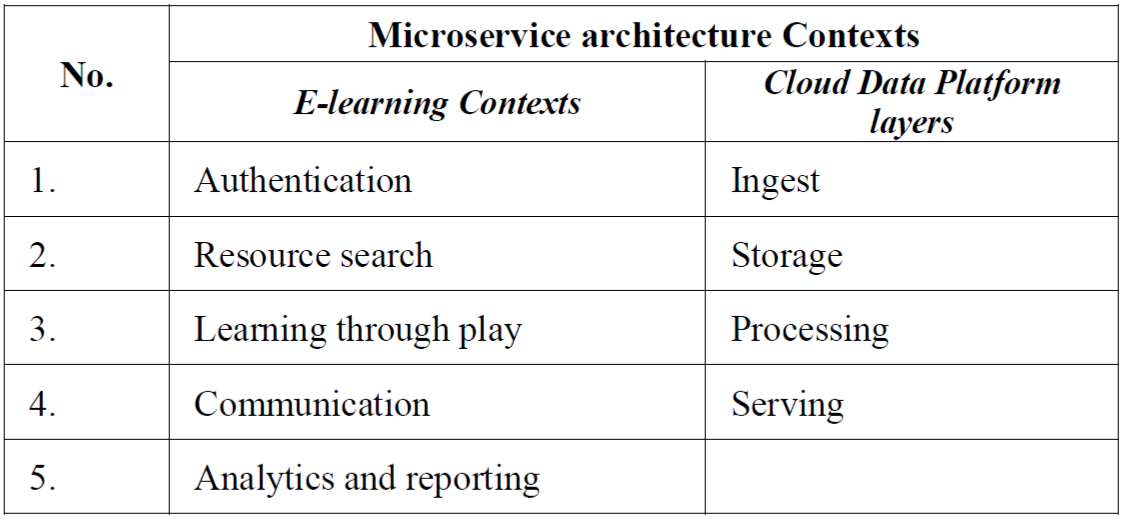
In order to create custom microservice architecture and to identify the microservices that make up the system, Domain Driven Design approach and the Bounded Contexts are used. In the paper, they first consult with an existing LMS platform called “Moodle” and identify that it has the following contexts:

1. Authentication service
2. Teaching materials search service – searches for teaching materials by certain criteria
3. Education games service – implementation of an educational quiz,
4. Communication service – implements chat functionality,
5. Analytics and reporting service – generating reports on student activities

These microservices communicate with Moodle LMS and with one another using an asynchronous publish / subscribe pattern in the form of a Message Broker.

As this is a data management platform, context identification is facilitated and relies on the layers identified in the universal architecture of the Cloud Data Platform. The contexts for the layers in the Cloud Data Platform are shown alongside their E-Learning Contexts in TABLE 1.

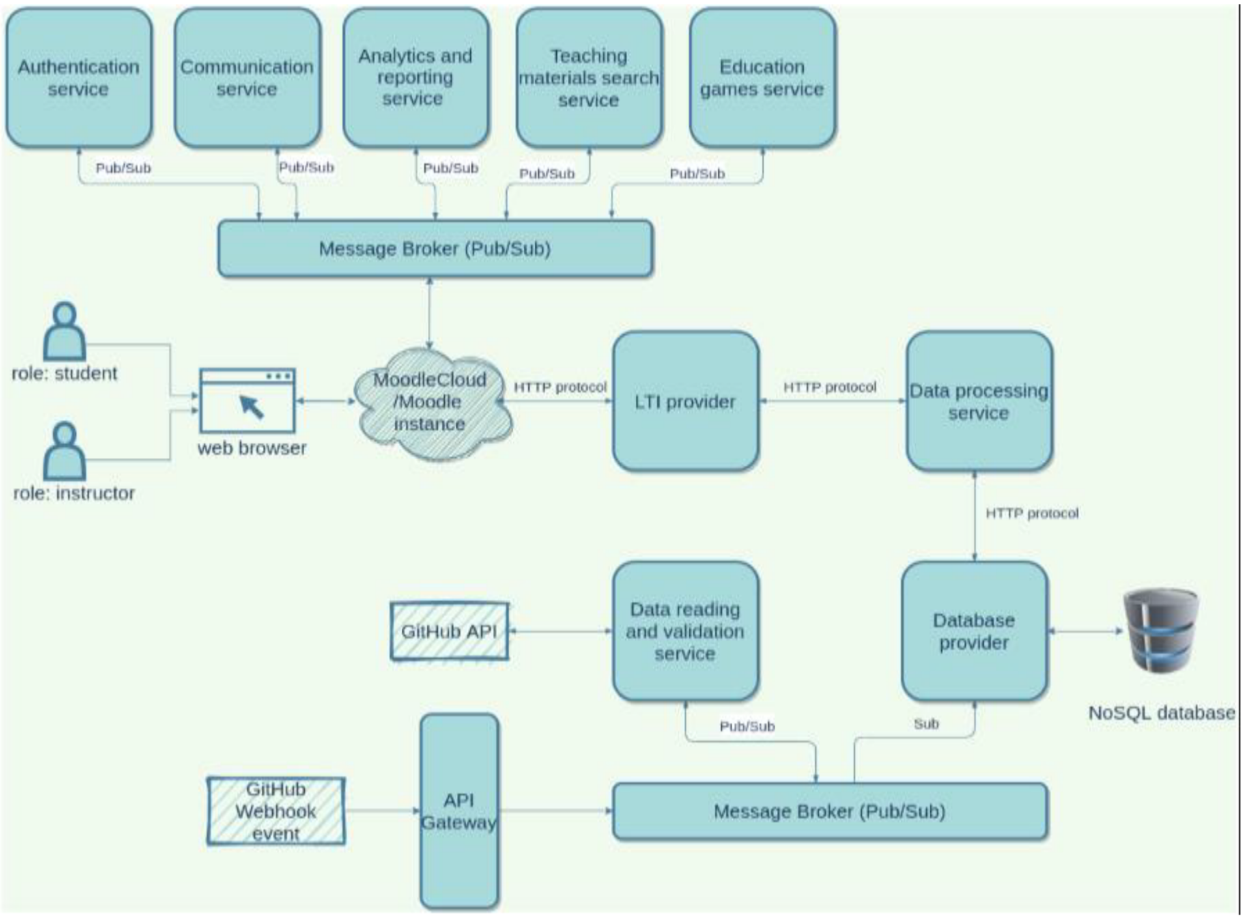
TABLE I. Indentified Contexts



No.	**Microservice architecture Contexts**	
	E-learning Contexts	***Cloud Data Platform layers***
1.	Authentication	Ingest
2.	Resource search	Storage
3.	Learning through play	Processing
4.	Communication	Serving
5.	Analytics and reporting	

Each of these layers: Ingest, Storage, Processing, Serving corresponds to a Microservice respectively: Data reading and validation microservice, Database provider microservice, Data processing microservice and microservice LTI provider.

The final design is shown in the figure below



Conclusion

In conclusion, it is important to state that, as the given microservice solution suggests, there is a possible way to implement microservice architecture by integrating with existing LMS systems.

“MILMS: A Microservices-based Learning Management System”

Author: **Odysseas Tsilingeridis, Alexandros Karakasidis**
Conference: 2020 IEEE International Conference on Big Data (Big Data)
Year Published: **2020**
DOI: 10.1109/BigData50022.2020.9378285

This paper provides a vision for a framework based on new technologies and tools, the authors named it MILMS. Applying methodologies for data storage and manipulation, namely Event Sourcing (ES) and Command Query Responsibility Segregation (CQRS), to achieve high scalability. This also proposes a domain decomposition into subdomains, each of which will be handled by a microservice. Finally it concludes by a brief discussion on some benefits of the proposed architecture.

ENVISIONED DOMAINS, SUBDOMAINS AND THEIR RESPECTIVE SERVICES OF MILMS

User Management, for user related operations	**Courses**, for the learning process	**Communication** among MILMS users	**Administration** for managing MILMS	**Evaluation** for learning outcome assessment
Group Management, which is responsible for forming typical (classes, teams) or informal user groups(collaborators). **Access Control**, controlling the access of the registered users. **Authentication**, confirming user identification. **Social Integration**, integration and communication with social media platforms.	**Study programs** which cater for assigning and organizing users (tutors and learners) with lectures and material. **Educational material** responsible for organizing, storing and distributing educational material based on user privileges. **Multimedia streaming** responsible for storing and streaming video lectures and podcasts	**Forum**, for user communication and collaboration around a multitude of topics. **Messaging**, for private text based communication between users. **Instant Messaging**, for real - time text based communication between users or groups of users. **Noticeboard**, for general announcements, or ones targeted toward specific audiences. **Teleconferencing**, for allowing real-time online courses.	**Control panels**, allowing system configuration for administrators, course configuration for tutors and portfolios for learners. **System analytics**, for allowing system administrators evaluate the performance and behavior of the system. **Learning analytics**, in order to provide analytical insights to tutors over personal and group learning progress. **Integration**, which allows integration and configuration based on the policies and infrastructure of the organization that deploys MILMS.	**Assignments**, which are created, assigned, evaluated. **Quizzes**, for self assessment and online examination. **Tools**, such as validation and plagiarism detection.

Table - 1

MILMS System

Design

- Asynchronous approach (Event Driven Architecture)
  - Database per service pattern
  - Micro Frontends (ongoing research of decoupling the frontend in separate microservices.
- For data storage
  - Event Sourcing along with CQRS.
  - Checkpointing.
- Domain-Driven Design
  - Further decomposed into sub-domains seen in Table 1

Operation

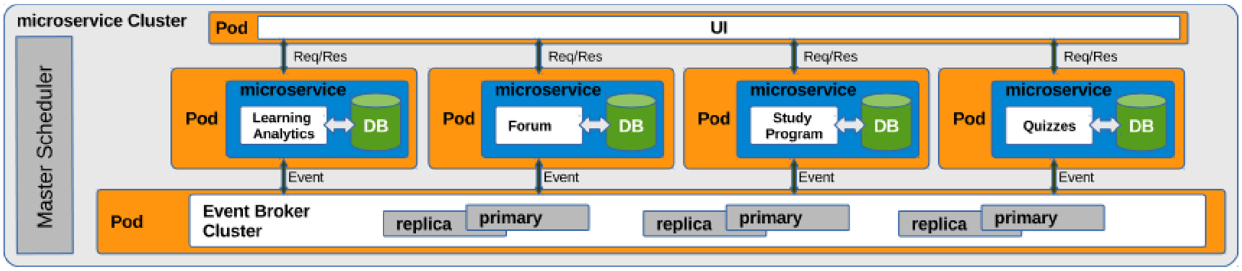


Fig. 1. Concept of MILMS Architecture

- Communication between microservices under publish / subscribe pattern.
- Client-side discovery pattern
- To communicate with external clients with API Gateways.
- For Load balancing and replication, a cluster methodology may be considered
  - Virtual machines
    - Simpler and straightforward way of isolating and encapsulating each microservice
    - But slow deployment and pre-defined static computing resource allocation
  - Containers
    - Lightweight solution due to operating system sharing and the absence of system libraries
    - Allowing the efficient handling and execution of multiple microservice instances in a single machine