An architecture for handling heterogenous services in a speech and language therapy platform

Evanthia Faliagka
Electrical & Computer Engineering Department
University of Peloponnese
Patras, Greece
e.faliagka@go.uop.gr

Christos Panagiotou
PIKEI - New Technologies
Patras, Greece
christos@pikei.io

Konstantinos Antonopoulos
Electrical & Computer Engineering Department
University of Peloponnese
Patras, Greece
k.antonopoulos@go.uop.gr

Vasileios Stavropoulos
Electrical & Computer Engineering Department
University of Peloponnese
Patras, Greece
ece1602924@go.uop.gr

Dimitris Karadimas
PIKEI - New Technologies
Patras, Greece
dimitris@pikei.io

Georgios Keramidas
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece gkeramidas@csd.auth.gr

Christos Antonopoulos
Electrical & Computer Engineering Department
University of the Peloponnese
Patras, Greece
ch.antonop@uop.gr

Nikolaos Voros
Electrical & Computer Engineering Department
University of the Peloponnese
Patras, Greece
voros@uop.gr

***Abstract*—Modern technology platforms have a crucial role in many areas of everyday life and have applications in various fields. In this paper the architecture of a speech therapy platform is presented, which is able to support the self-management of patients with chronic speech and language disorders. The developed infrastructure comprises the main outcome in the context of Research-Create-Innovate Plan-V project (https://planv-project.gr/).**

Keywords-speech and language platform; cloud architecture; micro-service architecture

I. INTRODUCTION

The proposed platform aims at developing an integrated, technologically assisted, Speech and Language Therapy intervention platform for people with chronic neurogenic communication disorders. The Plan-V platform will allow patients to practice in their own environment, with the assistance of a virtual agent (Avatar), without the physical presence of a clinician. At the same time, the platform will constitute a valuable clinical tool to assist clinicians in the design and remote management of individualized intervention programs, as well as with the process of monitoring, assessment and evaluation. The exercises that the platform will provide are divided into 3 categories: comprehension exercises, speech production exercises and speech exercises while the exchanged data between the users and the platform can be represented and realized either in text or in multimedia format (images, audio and video).

The scope of the paper is to present and evaluate the architecture of a Speech and Language Therapy Platform that will support therapists to configure and assign to their patients' routines of exercises that will improve their speech and language skills. The impact of this solution is multifaceted as it will allow people to conduct their therapies remotely without restrictions that the pandemic introduced recently. Moreover, therapists will be supported by a powerful tool that will facilitate the evaluations of the patients' status and progress. This support will increase the capacity of therapists and medical experts, since the need for physical presence and supervision will be decreased significantly. On the technological point of view, the proposed platform incorporates technologies in a state-of-the-art, scalable architecture that manages to efficiently handle heterogeneous data in terms of data type (images, video, audio, structured data) and data volume.

The rest of the paper is structured as follows: Section 1 provides an introduction. In Section 2 the proposed architecture

is presented. In section 3 the basic components are shown and section 4 concludes this work.

II. Architecture

A micro-service architecture can be installed in various infrastructures, with Docker technology being the most widespread and prevalent at the moment. Docker technology is open source, and is used to automate the development of applications that are portable, self-sufficient, in their own environment, which can run on any infrastructure.

In a virtualization solution, we run a new operating system over a hypervisor (e.g. ESXi, Proxmox, Hyper-V), which is installed on another operating system. Running a complete system means consuming extra storage space as we use the entire stack, from the kernel to the file system. Unlike Docker, we only use the file system along with the executable files of an operating system, so we do not need to use all its stack. Docker images have much less storage requirements and this allows us to run multiple docker containers on the same machine. The structural elements of the proposed architecture, shown in Figure 1 include:

- Platform Parameter Management: It is used to manage the various parameters of the system. Enables the platform for centralized management and sharing of parameters, in all services and their change during their execution, without the need to restart the applications. It relies on the open-source solution, Zookeeper [1].
- Database: It is based on the open source PostgreSQL solution [2], and is used to store and retrieve data.
- Cache: Cache significantly improves system performance. Within Plan-V, cache is used as a means of storing products in the customer's shopping cart.
- File Warehouse: This is the main system used for file storage. In Plan-V it is used to store the audio files that will be used by the digital assistant and those that will be produced by the users of the platform as well as the images that are part of the exercises.

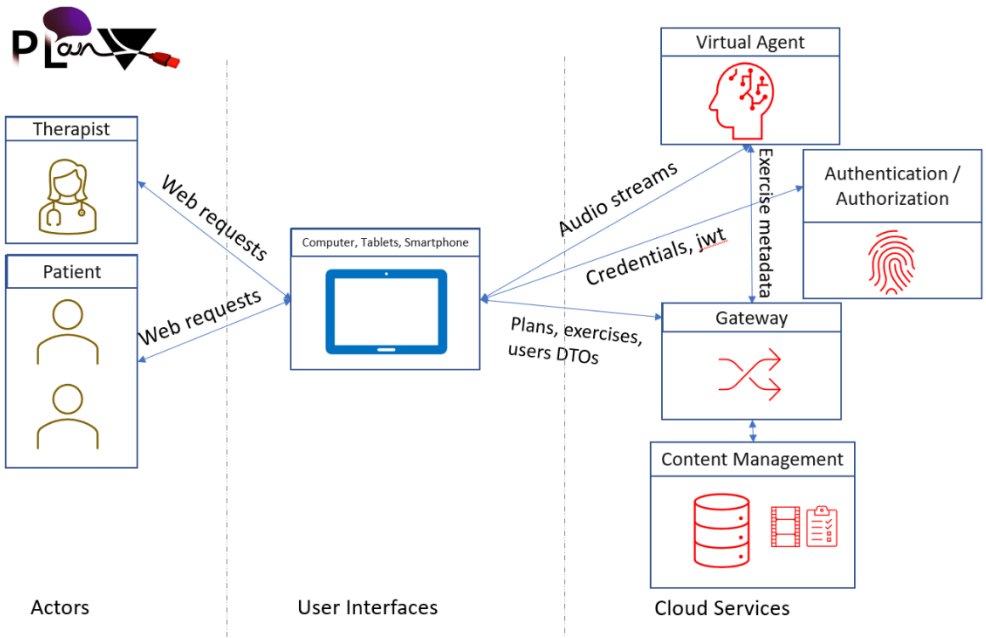

Figure 1: PLAN-V architecture

III. BASIC COMPONENTS

A. Cloud Architecture

The proposed infrastructure shown in Figure 2, is based on Docker technology and follows the micro-service architecture.

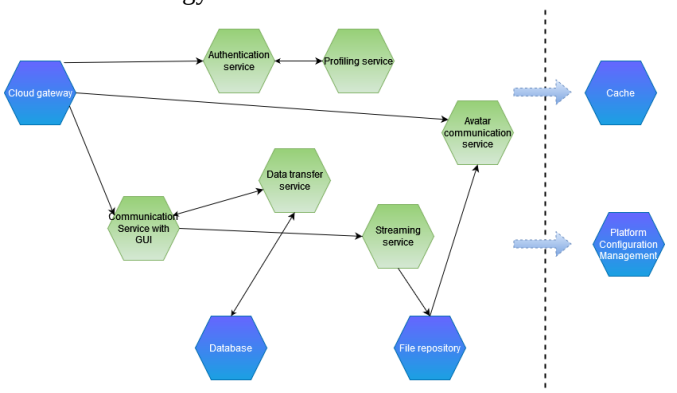

Figure 2: Micro-services to be implemented

B. PLANV - Restful API

Interfaces are a stand-alone service in relation to the other subsystems of the platform that can interact through well-defined programming interfaces (APIs). The technologies used are Restful services [3] and Websocket [4] and as a result a loose connection is achieved with the rest of the platform. This allows the easy adaptation of the interfaces to new requirements and their unaffected operation. Plan- V Restful API implements the following functionalities as described below (with their corresponding API Endpoints):

Plans Functionality tracks the plans that each patient has performed. Information that can be retrieved from plans functionality, is a correlated title for the plan along with a predefined (estimated) time that a patient needs for his daily session. The provided endpoints are shown in Figure 3.

plans-api

GET	/api/v1/plans/{plansId}
PUT	/api/v1/plans/{plansId}
DELETE	/api/v1/plans/{plansId}
GET	/api/v1/plans
POST	/api/v1/plans

Figure 3. Plans API

Templates Functionality tracks the configuration of a given template, which is correlated to a specific plan. Information that can be retrieved from templates functionality are the template category, word category and subcategory, syllabi number, difficulty level and presentation delay. The provided endpoints are shown in Figure 4.

plan-template-api

GET	/api/v1/plans/template/{plantemplateId}
PUT	/api/v1/plans/template/{plantemplateId}
DELETE	/api/v1/plans/template/{plantemplateId}
GET	/api/v1/plans/template
POST	/api/v1/plans/template
GET	/api/v1/plans/{planId}/template

Figure 4. Templates API

Exercises functionality is used to retrieve the corresponding image names according to the id that is stored in the file repository. It also used to retrieve the appropriate data for a specific exercise according to the template id and the row id that is provided by the user. The provided endpoints are shown in Figure 5.

exercise-api

GET	/api/v1/language1Connection
GET	/api/v1/language1Connection/{Language1ConnectionId}
GET	/api/v1/exercises/{templateId}
GET	/api/v1/exercises/{templateId}/{rowId}

Figure 5. Exercises API

Patient Exercises functionality tracks the scores achieved by the patient to specific exercises. Indicative scores are category, food, and movement scores. The provided endpoints are show in Figure 6.

exercise-patient-api

GET	/api/v1/exerciseevaluation/{exerciseevaluationId}
PUT	/api/v1/exerciseevaluation/{exerciseevaluationId}
DELETE	/api/v1/exerciseevaluation/{exerciseevaluationId}
GET	/api/v1/exerciseevaluation
POST	/api/v1/exerciseevaluation

Figure 6. Patients Exercise API

Multimedia functionality, used mainly from the Web interface to load multimedia content like video and images. The provided endpoints are shown in Figure 7.

Figure 7. Multimedia API

C. Interfaces

Graphic interfaces are one of the key components of the system. Users (patients and speech therapists) will be able to interact with the platform through self-intuitive and user-friendly graphical interfaces. The graphical interfaces visualize the speech and language intervention plans and also provide certain interactions with the virtual assistant during the execution of the assigned plans. To meet the modern needs of web-based applications, especially those with special requirements such as the graphical interfaces of Plan-V, all the current trends in the design and development of interfaces have been employed.

As in the Plan-V cloud computing infrastructure discussed earlier, graphical interfaces are a reactive web application [5] served by a web server running in the protected environment of a Docker container. Unlike web applications that are closely connected to the server, reactive web applications react immediately to any user reaction. In addition, the computing load of the server is significantly reduced, thus improving the overall response of the interfaces to much higher user loads. These requirements are critical for the Plan-V interfaces as they are comprised of multiple modules with their own state and high demands in rendering.

A key component in the development of Plan-V reactive interfaces is the ReactJS library [6]. ReactJS (React) is a JavaScript library for creating graphical web interfaces. Developed and maintained by Facebook and a community of developers and companies. To serve the needs of the Plan-V platform, apart from the core React libraries, the use of additional modules for state management, routing, and API interaction were employed.

D. User Management

Users involved with the platform are a) Patients, b) Speech therapists, c) Speech Therapy Centers. By registering the platform, the patient will have the opportunity to perform the exercises of the therapy plan set by his speech therapist. A therapist can set the therapy plans, assign them to his patients and evaluate the patients' answers.

Every system that is available online and manages sensitive data such as patients' personal & medical information is required to ensure the security of these data and the privacy of its users. In this context, Plan-V adopts the OAuth2 [7] platform authorization protocol. OAuth2 is an authorization framework that allows applications to gain limited access to user accounts via an online service and its general description is shown in Fig. 8.

OAuth has four roles:

• Resource Owner: The resource holder is the user who authorizes an application to access their account. The access of the application to the user's account is limited to the scope of the authorization granted (e.g. read or write access)

• Client: The client is the application that wants to access the user's account. To be able to do so, it must be authorized by the user

• Resource server: The resource server hosts protected user accounts

• Authorization Server: The authorization server verifies the user's identity and then issues application access tokens

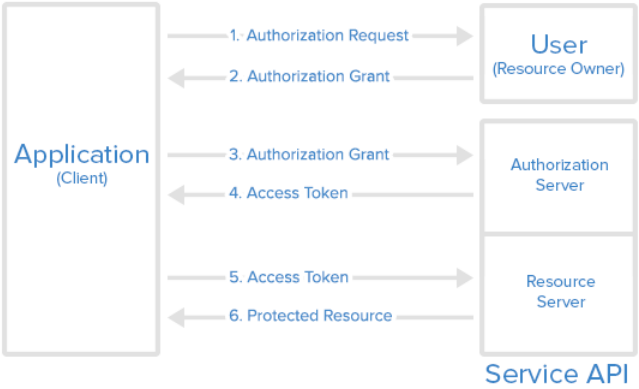

Figure 8: General description of OAuth2 flow

An overview of the OAuth2 authorization process is shown in Fig. 8, with a more detailed explanation of these steps:

1. The application requests authorization for access to service resources by the user
2. If the user authorized the request, the application receives an authorization grant
3. The application requests an access token from the authorization server (API) by presenting its authentication and authorization
4. If the application is authenticated and the authorization is valid, the authorization server (API) issues an access token to the application. The authorization has been completed.
5. The application requests the resource from the Resource Server (API) and presents the access token for authentication
6. If the access token is valid, the Resource Server (API) serves the resource in the application

The adoption of the OAuth2 standard for the authorization of access to the resources of the Plan-V platform and the specialization of its implementation for the purposes of the project, achieves the central and unified control of the access by the individual subsystems and the users of the platform.

E. File Management Subsystem

The architectural selection of microservices through the distributed operation of multiple docker containers within a Kubernetes cluster gives many facilities as has been made clear so far. At the same time, however, it introduces many

challenges such as the management of files that the platform must share. These files are images and videos that compose the exercises of the intervention plans. Plan-V met this challenge by adopting Minio [8] as the technology for storing and disposing of the files needed to execute intervention plans.
More specifically, Minio is an open source distributed storage server written in Go, designed for cloud computing infrastructure. Minio is suitable for storing unstructured data such as photos, videos, logs, backups and containers.
The basic interface for the Plan-V Minio service is given in Figure 9.

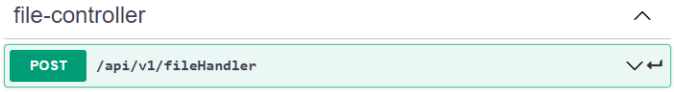

Figure 9: Minio Interface

F. Database

Exercise and user data as well as the results of the exercises will be stored in a relational database to be implemented in PostgreSQL. Figure 5 shows a part of the database schema, where the basic tables and their correlations can be seen.
The tables that will store the data of the basic entities, i.e. the users and the exercises, are shown in yellow. There will be 3 groups of users (patients, speech therapists and treatment centers) and for each entity a table will be created as shown in Figure 10.
The speech therapist will assign a plan for each patient, i.e. a set of exercises to be completed. The exercises that will be implemented belong to categories of exercises and each one has different data that need to be stored. Exercises with the same type of data will be stored in the same table and their retrieval will be based on the category they belong. The results of the exercises will be stored in other tables, which are shown in the figure in green, where in addition to the information we need depending on the exercise (e.g., response time, selection) will be stored the date of the exercise and the paths of resulting files (e.g. speech exercises will always result in an audio file).

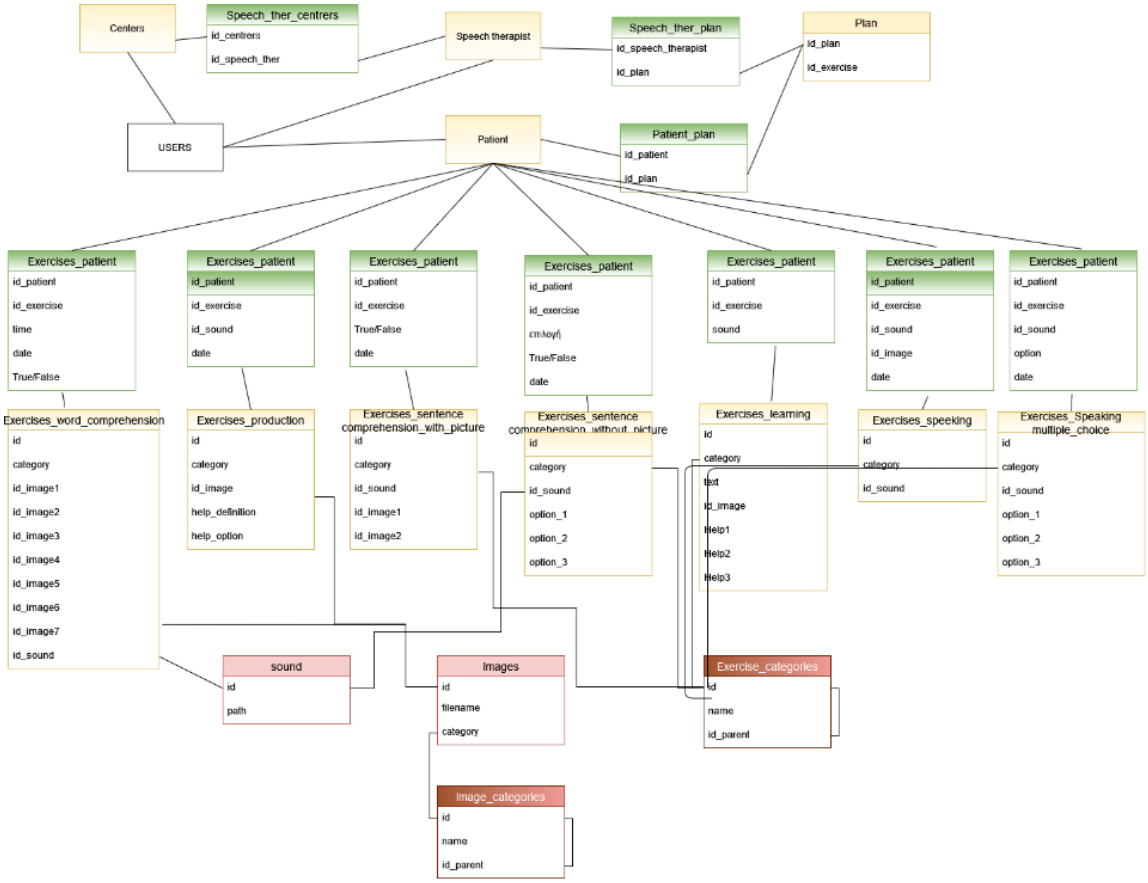

Figure 10:Database Schema

IV. Conclusions

In this work the architecture of a speech language therapy platform was presented among with the details of all components developed. The proposed platform incorporates technologies in a scalable architecture that manages to efficiently handle heterogeneous data in terms of data type. In the future we plan to explore gamification aspects of the therapeutic interventions to increase user's motivation in favor of speech and language therapies.

Acknowledgment

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH CREATE INNOVATE (project code: T2EΔK-02159 PLan-V).

References

[1] Zookeeper. https://zookeeper.apache.org/

[2] Momjian B. PostgreSQL (2001). Introduction and concepts. New York: Addison-Wesley.

[3] Richardson, L., & Ruby, S. (2008). RESTful web services. " O'Reilly Media, Inc."

[4] Fette, Ian, and Alexey Melnikov (2011). The websocket protocol.

[5] M. Bernhardt, Reactive Web Applications: Covers Play, Akka, and Reactive Streams., Simon and Schuster, 2016

[6] Website: https://reactjs.org/

[7] Website: https://oauth.net/2/.

[8] Website: https://min.io/.